aws

Guide de l'utilisateur

Amazon CloudWatch

Amazon CloudWatch: Guide de l'utilisateur

Table of Contents

Qu'est-ce qu'Amazon CloudWatch ?

Amazon CloudWatch surveille vos ressources Amazon Web Services (AWS) et les applications que vous utilisez AWS en temps réel. Vous pouvez les utiliser CloudWatch pour collecter et suivre les métriques, qui sont des variables que vous pouvez mesurer pour vos ressources et vos applications.

La page d' CloudWatch accueil affiche automatiquement les statistiques relatives à chaque AWS service que vous utilisez. Vous pouvez également créer des tableaux de bord personnalisés pour afficher les métriques relatives à vos applications personnalisées, ainsi que des collections personnalisées de métriques de votre choix.

Vous pouvez créer des alertes pour surveiller les métriques et envoyer des notifications ou apporter automatiquement des modifications aux ressources surveillées, lorsqu'un seuil est dépassé. Par exemple, vous pouvez surveiller l'utilisation de l'UC ainsi que les lectures et écritures sur disque de vos instances Amazon EC2, puis utiliser ces données pour déterminer si vous devez lancer des instances supplémentaires afin de faire face à l'augmentation de la charge. Vous pouvez également utiliser ces données pour arrêter les instances sous-utilisées et réaliser des économies.

Vous bénéficiez CloudWatch ainsi d'une visibilité à l'échelle du système sur l'utilisation des ressources, les performances des applications et la santé opérationnelle.

Accès CloudWatch

Vous pouvez y accéder CloudWatch en utilisant l'une des méthodes suivantes :

- CloudWatch Console Amazon — https://console.aws.amazon.com/cloudwatch/
- AWS CLI — Pour plus d'informations, voir Getting Set Up with the AWS Command Line Interface dans le guide de AWS Command Line Interface l'utilisateur.
- CloudWatch API — Pour plus d'informations, consultez le Amazon CloudWatch API Reference.
- AWS SDK — Pour plus d'informations, consultez Outils pour Amazon Web Services.

AWS Services connexes

Les services suivants sont utilisés conjointement avec Amazon CloudWatch :

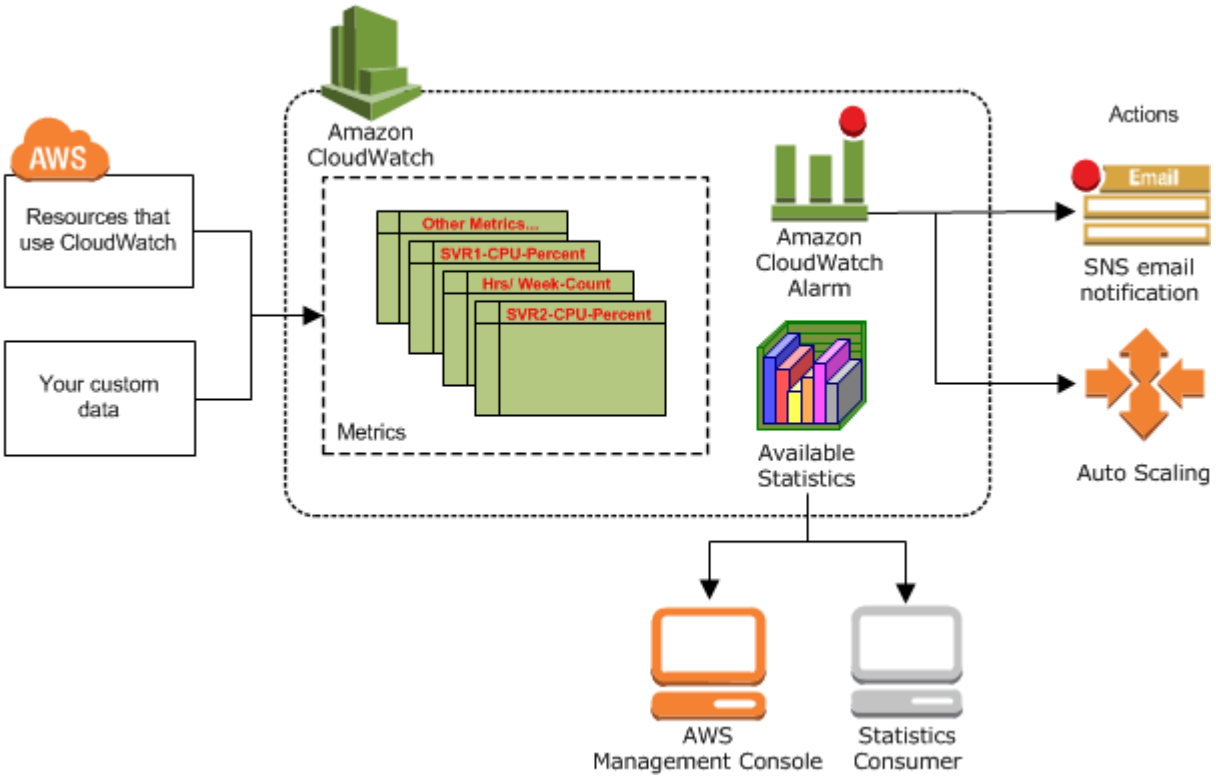
- Amazon Simple Notification Service (Amazon SNS) coordonne et gère la mise à disposition ou l'envoi de messages à des clients ou à des points de terminaison abonnés. Vous utilisez Amazon

SNS CloudWatch pour envoyer des messages lorsqu'un seuil d'alarme est atteint. Pour plus d'informations, consultez Configuration des notifications Amazon SNS.

- Amazon EC2 Auto Scaling vous permet de lancer ou de mettre hors service automatiquement des instances Amazon EC2 en fonction des stratégies définies par l'utilisateur, des vérifications de l'état et des calendriers. Vous pouvez utiliser une CloudWatch alarme avec Amazon EC2 Auto Scaling pour dimensionner vos instances EC2 en fonction de la demande. Pour de plus amples informations, veuillez consulter Dimensionnement dynamique dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.
- AWS CloudTrailvous permet de surveiller les appels passés à l' CloudWatch API Amazon pour votre compte, y compris les appels passés par le AWS Management Console AWS CLI, et d'autres services. Lorsque la CloudTrail journalisation est activée, CloudWatch écrit les fichiers journaux dans le compartiment Amazon S3 que vous avez spécifié lors de la configuration CloudTrail. Pour plus d'informations, consultez Journalisation des appels CloudWatch d'API Amazon avec AWS CloudTrail.
- AWS Identity and Access Management (IAM) est un service Web qui vous permet de contrôler en toute sécurité l'accès aux AWS ressources pour vos utilisateurs. Utilisez IAM pour contrôler qui peut utiliser vos ressources AWS (authentification) et quelles ressources pourront être utilisées de quelle manière (autorisation). Pour plus d'informations, consultez Gestion des identités et des accès pour Amazon CloudWatch.

Comment CloudWatch fonctionne Amazon

Amazon CloudWatch est essentiellement un référentiel de métriques. Un AWS service, tel qu'Amazon EC2, place des métriques dans le référentiel, et vous récupérez des statistiques en fonction de ces métriques. Si vous mettez vos propres métriques personnalisées dans le référentiel, vous pouvez également extraire des statistiques sur ces métriques.



Vous pouvez utiliser des métriques pour calculer des statistiques, puis présenter les données graphiquement dans la CloudWatch console. Pour plus d'informations sur les autres AWS ressources qui génèrent et envoient des métriques CloudWatch, consultezAWS services qui publient CloudWatch des statistiques.

Vous pouvez configurer des actions d'alerte pour arrêter ou démarrer une instance Amazon EC2, ou y mettre fin, quand certaines conditions sont satisfaites. De plus, vous pouvez créer des alertes qui initient des actions Amazon EC2 Auto Scaling et Amazon Simple Notification Service (Amazon SNS) en votre nom. Pour plus d'informations sur la création d' CloudWatchalarmes, consultezalertes.

AWS Les ressources de cloud computing sont hébergées dans des centres de données à haute disponibilité. Pour offrir une fiabilité et une capacité de mise à l'échelle supplémentaires, chaque installation de centre de données se trouve dans une zone géographique spécifique, appelée région. Chaque région est conçue pour être totalement isolée des autres régions, l'objectif étant d'isoler autant que possible les défaillances et d'optimiser la stabilité. Les métriques sont stockées séparément dans les régions, mais vous pouvez utiliser CloudWatch la fonctionnalité inter-régions pour agréger les statistiques de différentes régions. Pour de plus amples informations, veuillez consulter Console multicompte et multirégion CloudWatch et Régions et points de terminaison dans le Référence générale d'Amazon Web Services.

CloudWatch Concepts d'Amazon

La terminologie et les concepts suivants sont essentiels à votre compréhension et à votre utilisation d'Amazon CloudWatch :

- Espaces de noms
- Métriques
- Dimensions
- Résolution
- Statistiques
- Centiles
- alertes

Pour plus d'informations sur les quotas de service pour les CloudWatch métriques, les alarmes, les demandes d'API et les notifications par e-mail d'alarme, consultez la section Quotas CloudWatch de service.

Espaces de noms

Un espace de noms est un conteneur pour les CloudWatch métriques. Les métriques d'applications différentes sont placées dans des espaces de noms différents et isolées ainsi les unes des autres afin de ne pas être regroupées par erreur dans les mêmes statistiques.

Il n'existe aucun espace de noms par défaut. Vous devez spécifier un espace de noms pour chaque point de données sur lequel vous publiez CloudWatch. Vous pouvez spécifier un nom d'espace de noms au moment de créer une métrique. Ces noms doivent contenir des caractères ASCII valides et comporter 255 caractères ou moins. Les caractères possibles sont les suivants : caractères alphanumériques (0-9a-Za-Z), point (.), tiret (-), trait de soulignement (_), barre oblique (/), hachage (#), deux-points (:) et le caractère espace. Un espace de noms doit contenir au moins un caractère autre qu'un espace.

Les AWS espaces de noms utilisent généralement la convention de dénomination suivante :`AWS/service`. Par exemple, Amazon EC2 utilise l'espace de noms `AWS/EC2`. Pour la liste des AWS espaces de noms, consultezAWS services qui publient CloudWatch des statistiques.

Métriques

Les métriques sont le concept fondamental de CloudWatch. Une métrique représente un ensemble chronologique de points de données publiés sur CloudWatch. Envisagez une métrique comme une variable à surveiller et les points de données comme les valeurs de cette variable au fil du temps. Par exemple, l'utilisation de l'UC d'une instance EC2 déterminée est une métrique fournie par Amazon EC2. Les points de données proprement dits peuvent être issus d'une application ou d'une activité métier dont vous collectez les données.

Par défaut, de nombreux AWS services fournissent des métriques gratuites pour les ressources (telles que les instances Amazon EC2, les volumes Amazon EBS et les instances de base de données Amazon RDS). Pour des frais, vous pouvez activer la surveillance détaillée de certaines ressources, telles que vos instances Amazon EC2, ou ajouter vos propres métriques d'applications. Pour des métriques personnalisées, vous pouvez ajouter les points de données dans n'importe quel ordre et à n'importe quelle fréquence. Vous pouvez extraire des statistiques sur ces points de données en tant qu'un ensemble ordonné de données chronologiques.

Les métriques existent uniquement dans la région où elles ont été créées. Les métriques ne peuvent pas être supprimées, mais elles arriveront automatiquement à expiration après 15 mois sans publication de nouvelles données. Les points de données de plus de 15 mois expirent sur une base continue ; quand de nouveaux points de données arrivent, les données de plus de 15 mois sont abandonnées.

Les métriques sont uniquement définies par un nom, un espace de noms et aucune ou plusieurs dimensions. Chaque point de données d'une métrique comporte un horodatage et (le cas échéant) une unité de mesure. Vous pouvez récupérer les statistiques CloudWatch de n'importe quelle métrique.

Pour plus d’informations, consultez Affichage des métriques disponibles et Publier des métriques personnalisées .

Horodatages

Chaque point de données de métrique doit être associé à un horodatage. L'horodatage peut remonter jusqu'à deux semaines ou être anticipé de deux heures. Si vous ne fournissez pas d'horodatage, CloudWatch crée un horodatage pour vous en fonction de l'heure à laquelle le point de données a été reçu.

Les horodatages sont des objets `dateTime` constitués de la date complète, à laquelle s'ajoutent les heures, les minutes et les secondes (par exemple, 2016-10-31T23:59:59Z). Pour plus d'informations,

consultez dateTime. Bien que vous n'y soyez pas contraint, nous vous recommandons d'utiliser le temps universel (UTC). Lorsque vous récupérez des statistiques à partir de CloudWatch, toutes les heures sont exprimées en UTC.

CloudWatch les alarmes vérifient les métriques en fonction de l'heure actuelle en UTC. Les métriques personnalisées envoyées CloudWatch avec un horodatage autre que l'heure UTC actuelle peuvent provoquer l'affichage de l'état « Données insuffisantes » ou retarder les alarmes.

Conservation des métriques

CloudWatch conserve les données métriques comme suit :

- Les points de données dont la période est inférieure à 60 secondes sont disponibles pendant 3 heures. Ces points de données sont des métriques personnalisées haute résolution.
- Les points de données d'une durée de 60 secondes (1 minute) sont disponibles pendant 15 jours
- Les points de données d'une durée de 300 secondes (5 minutes) sont disponibles pendant 63 jours
- Les points de données d'une durée de 3 600 secondes (1 heure) sont disponibles pendant 455 jours (15 mois)

Les points de données qui sont initialement publiés pour une plus courte période sont regroupés pour un stockage à long terme. Par exemple, si vous collectez des données sur une période d'1 minute, les données restent disponibles pendant 15 jours avec une résolution d'1 minute. Après 15 jours, ces données restent disponibles mais elles sont regroupées et récupérables uniquement avec une résolution de 5 minutes. Après 63 jours, ces données sont de nouveau regroupées et disponibles avec une résolution d'1 heure.

> **Note**
>
> Les métriques qui n'ont pas eu de nouveaux points de données au cours des deux dernières semaines n'apparaissent pas dans la console. Ils n'apparaissent pas non plus lorsque vous tapez leur nom de métrique ou leur nom de dimension dans la zone de recherche de l'onglet All metrics (Toutes les métriques) de la console, et ils ne sont pas renvoyés dans les résultats d'une commande list-metrics . La meilleure façon de récupérer ces métriques est d'utiliser les get-metric-statisticscommandes get-metric-dataor du AWS CLI.

Dimensions

Une dimension est une paire nom-valeur qui fait partie de l'identité d'une métrique. Vous pouvez associer jusqu'à 30 dimensions à une métrique.

Chaque métrique est décrite par des caractéristiques spécifiques, et vous pouvez imaginer les dimensions comme des catégories de ces caractéristiques. Les dimensions vous aident à concevoir une structure pour votre plan de statistiques. Comme les dimensions font partie de l'identifiant unique d'une métrique, chaque fois que vous ajoutez une paire nom/valeur unique à l'une de vos métriques, vous créez une nouvelle variation de cette métrique.

AWS services qui envoient des données pour CloudWatch associer des dimensions à chaque métrique. Vous pouvez utiliser des dimensions pour filtrer les résultats CloudWatch renvoyés. Par exemple, vous pouvez obtenir les statistiques d'une instance EC2 déterminée en spécifiant la dimension `InstanceId` lorsqu'il s'agit de rechercher des métriques.

Car les métriques produites par certains AWS services, tels qu'Amazon EC2, CloudWatch peuvent agréger les données de différentes dimensions. Par exemple, si vous recherchez des métriques dans l'espace de `AWS/EC2` noms mais que vous ne spécifiez aucune dimension, CloudWatch agrège toutes les données pour la métrique spécifiée afin de créer la statistique que vous avez demandée. CloudWatch n'agrège pas toutes les dimensions pour vos statistiques personnalisées.

Combinaisons de dimensions

CloudWatch traite chaque combinaison unique de dimensions comme une métrique distincte, même si les métriques portent le même nom de métrique. Vous pouvez uniquement récupérer des statistiques en utilisant des combinaisons de dimensions que vous n'avez pas spécifiquement publiées. Lorsque vous récupérez des statistiques, indiquez les mêmes valeurs pour l'espace de noms, le nom de métrique et les paramètres de dimension qui ont été utilisés lors de la création des métriques. Vous pouvez également spécifier les heures de début et de fin CloudWatch à utiliser pour l'agrégation.

Supposons, par exemple, que vous publiez quatre métriques distinctes nommées ServerStats dans l'espace de DataCenterMetric noms avec les propriétés suivantes :

```
Dimensions: Server=Prod, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:30:00Z, Value: 105
Dimensions: Server=Beta, Domain=Frankfurt, Unit: Count, Timestamp:
 2016-10-31T12:31:00Z, Value: 115
```

```
Dimensions: Server=Prod, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:32:00Z, Value: 95
Dimensions: Server=Beta, Domain=Rio,       Unit: Count, Timestamp:
 2016-10-31T12:33:00Z, Value: 97
```

Si vous publiez uniquement ces quatre métriques, vous pouvez récupérer les statistiques pour ces combinaisons de dimensions :

- `Server=Prod,Domain=Frankfurt`
- `Server=Prod,Domain=Rio`
- `Server=Beta,Domain=Frankfurt`
- `Server=Beta,Domain=Rio`

Vous ne pouvez pas récupérer de statistiques pour les dimensions suivantes ou si vous ne spécifiez aucune dimension. (L'exception est l'utilisation de la fonction de mathématiques de métriques SEARCH qui peut récupérer des statistiques pour plusieurs métriques. Pour plus d'informations, consultez Utiliser des expressions de recherche dans les graphiques.)

- `Server=Prod`
- `Server=Beta`
- `Domain=Frankfurt`
- `Domain=Rio`

Résolution

Chaque métrique appartient à l'une des catégories suivantes :

- Résolution standard, avec des données dont la granularité est d'une minute
- Haute résolution, avec des données dont la granularité est d'une seconde

Les métriques produites par les AWS services ont une résolution standard par défaut. Lorsque vous publiez une métrique personnalisée, vous pouvez la définir en tant que résolution standard ou haute résolution. Lorsque vous publiez une métrique haute résolution, que vous la CloudWatch stockez avec une résolution de 1 seconde, et vous pouvez la lire et la récupérer sur une période de 1 seconde, 5 secondes, 10 secondes, 30 secondes ou un multiple de 60 secondes.

Les métriques haute résolution peuvent vous donner des informations immédiates sur l'activité de votre application sur une période inférieure à une minute. Gardez à l'esprit que chaque appel `PutMetricData` pour des métriques personnalisées est facturé, donc des appels `PutMetricData` plus fréquents sur une métrique haute résolution peut entraîner des frais plus élevés. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Si vous définissez une alerte sur une métrique haute résolution, vous pouvez spécifier une alerte haute résolution avec une période de 10 secondes ou de 30 secondes ou vous pouvez définir une alerte régulière avec une période correspondant à n'importe quel multiple de 60 secondes. Les frais sont plus élevés pour les alertes haute résolution dont la période est de 10 ou 30 secondes.

Statistiques

Les statistiques sont des agrégations de données métriques sur des périodes spécifiques. CloudWatch fournit des statistiques basées sur les points de données métriques fournis par vos données personnalisées ou fournis par d'autres AWS services à CloudWatch. Les regroupements sont effectués en utilisant l'espace de noms, le nom métrique, les dimensions et l'unité de mesure des points de données, pendant la période spécifiée.

Pour des définitions détaillées des statistiques prises en charge par CloudWatch, voirCloudWatch définitions des statistiques.

Unités

Chaque statistique est associée à une unité de mesure. Il peut s'agir, par exemple, des unités `Bytes`, `Seconds`, `Count` ou `Percent`. Pour obtenir la liste complète des unités prises CloudWatch en charge, consultez le type de MetricDatumdonnées dans le Amazon CloudWatch API Reference.

Vous pouvez spécifier une unité au moment de créer une métrique personnalisée. Si vous ne spécifiez pas d'unité, CloudWatch `None` utilisez-la comme unité. Les unités permettent de donner une signification conceptuelle à vos données. Bien qu'elle n' CloudWatch attache aucune importance à une unité en interne, d'autres applications peuvent en déduire des informations sémantiques basées sur l'unité.

Les points de données de métriques qui spécifient une unité de mesure sont regroupés séparément. Lorsque vous obtenez des statistiques sans spécifier d'unité, CloudWatch agrège tous les points de données d'une même unité. Si vous avez deux métriques identiques avec des unités différentes, deux flux de données distincts sont renvoyés, un pour chaque unité.

Périodes

Une période est la durée associée à une CloudWatch statistique Amazon spécifique. Chaque statistique représente un regroupement des données des métriques recueillies pendant une durée spécifiée. Les périodes sont définies en nombre de secondes, et les valeurs valides pour la période sont 1, 5, 10, 30 ou un multiple de 60. Ainsi, pour spécifier une période de six minutes, vous utilisez la valeur 360. Vous pouvez ajuster le regroupement des données en variant la durée de la période. La valeur par défaut d'une période est de 60 secondes. Une période peut être aussi courte qu'une seconde et doit être un multiple de 60 si elle est supérieure à la valeur par défaut de 60 secondes.

Seules les métriques personnalisées que vous définissez avec une résolution de stockage d'1 seconde prennent en charge les périodes inférieures à une minute. Même si la possibilité de définir une période inférieure à 60 est toujours disponible dans la console, vous devez sélectionner une période qui s'adapte à la façon dont la métrique est stockée. Pour plus d'informations sur les métriques qui prennent en charge des périodes inférieures à une minute, consultez Métriques haute résolution.

Lorsqu'il s'agit de récupérer des statistiques, vous pouvez spécifier une période, une heure de début et une heure de fin. Ces paramètres déterminent la durée totale associée aux statistiques. Les valeurs par défaut d'heure de début et d'heure de fin vous donnent la dernière heure de statistiques. Les valeurs que vous spécifiez pour l'heure de début et l'heure de fin déterminent le nombre de périodes CloudWatch renvoyées. Par exemple, la récupération de statistiques à partir des valeurs par défaut de période, d'heure de début et d'heure de fin retourne un ensemble regroupé de statistiques pour chaque minute de l'heure précédente. Si vous préférez obtenir des statistiques regroupées en blocs de 10 minutes, spécifiez une période égale à 600. Pour les statistiques regroupées sur l'heure entière, spécifiez une période de 3 600.

Lorsque les statistiques sont regroupées sur une période, elles comportent l'heure correspondant au début de la période. Par exemple, les données regroupées de 19 h à 20 h comportent l'indication 19 h. De plus, les données agrégées entre 19 h 00 et 20 h 00 commencent à être visibles à 19 h 00, puis les valeurs de ces données agrégées peuvent changer à mesure que de nouveaux échantillons sont CloudWatch collectés au cours de la période.

Les périodes sont également importantes pour les CloudWatch alarmes. Lorsque vous créez une alarme pour surveiller une métrique spécifique, vous demandez CloudWatch de comparer cette métrique à la valeur de seuil que vous avez spécifiée. Vous avez un contrôle étendu sur la manière dont CloudWatch cette comparaison est effectuée. Non seulement vous pouvez spécifier la période sur laquelle la comparaison doit porter, mais vous pouvez aussi définir le nombre de périodes

d'évaluation utilisées pour parvenir à une conclusion. Par exemple, si vous spécifiez trois périodes d'évaluation, CloudWatch compare une fenêtre de trois points de données. CloudWatch vous avertit uniquement si le point de données le plus ancien est violé et si les autres sont violés ou manquants.

Agrégation

Amazon CloudWatch agrège les statistiques en fonction de la durée de la période que vous spécifiez lors de la récupération des statistiques. Vous pouvez publier autant de points de données que vous le souhaitez avec des horodatages identiques ou similaires. CloudWatch les agrège en fonction de la durée de période spécifiée. CloudWatch n'agrège pas automatiquement les données entre les régions, mais vous pouvez utiliser les mathématiques métriques pour agréger les métriques de différentes régions.

Vous pouvez publier des points de données pour une métrique qui partagent non seulement le même horodatage, mais également le même espace de noms et les mêmes dimensions. CloudWatch renvoie des statistiques agrégées pour ces points de données. Vous pouvez aussi publier plusieurs points de données pour une métrique identique ou des métriques différentes, avec n'importe quel horodatage.

Pour les ensembles de données volumineux, vous pouvez insérer un ensemble de données regroupées au préalable, appelé ensemble de statistiques. Avec les ensembles de statistiques, vous indiquez CloudWatch le minimum, le maximum, la somme et SampleCount pour un certain nombre de points de données. Ils sont couramment utilisés lorsqu'il est nécessaire de recueillir des données plusieurs fois à la minute. Par exemple, supposons que vous disposez d'une métrique qui mesure la latence de demande d'une page Web. Il ne serait pas judicieux de publier des données après chaque accès à la page Web. Nous vous suggérons de collecter la latence de tous les accès à cette page Web, de les agréger une fois par minute et d'envoyer cette statistique définie sur. CloudWatch

Amazon CloudWatch ne différencie pas la source d'une métrique. Si vous publiez une métrique avec le même espace de noms et les mêmes dimensions à partir de sources différentes, CloudWatch traitez-la comme une métrique unique. Cela peut être utile pour les métriques de service dans un système dimensionné distribué. Par exemple, tous les hôtes d'une application de serveur Web peuvent publier des métriques identiques représentant la latence des demandes qu'ils traitent. CloudWatchles traite comme une métrique unique, ce qui vous permet d'obtenir les statistiques relatives au minimum, au maximum, à la moyenne et à la somme de toutes les demandes de votre application.

Centiles

Un centile indique la position relative d'une valeur dans un ensemble de données. Par exemple, le 95e centile signifie que 95 % des données sont inférieures à cette valeur et que 5 % des données lui sont supérieures. Les centiles vous permettent de mieux comprendre la distribution des données de vos métriques.

Les centiles sont souvent utilisés pour isoler les anomalies. Dans une distribution normale, 95 % des données se situent à deux écarts types de la moyenne et 99,7 % des données se situent à trois écarts types de la moyenne. Les données qui se situent au-delà de trois écarts types sont souvent considérées comme des anomalies, car elles sont très éloignées de la valeur moyenne. Par exemple, supposons que vous surveillez l'utilisation de l'UC de vos instances EC2 pour vous assurer que vos clients ont une expérience satisfaisante. Si vous surveillez la moyenne, cela peut occulter des anomalies. Si vous surveillez la valeur maximale, la moindre anomalie peut fausser les résultats. En utilisant des centiles, vous pouvez surveiller le 95e centile de l'utilisation de l'UC pour identifier les instances dont la charge est anormalement élevée.

Certains CloudWatch indicateurs prennent en charge les percentiles en tant que statistiques. Pour ces indicateurs, vous pouvez surveiller votre système et vos applications à l'aide de percentiles comme vous le feriez pour les autres CloudWatch statistiques (moyenne, minimale, maximale et somme). Par exemple, lorsque vous créez une alerte, vous pouvez utiliser les centiles comme fonction statistique. Vous pouvez spécifier le centile avec dix décimales maximum (par exemple, p95.0123456789).

Les statistiques sur les centiles sont disponibles pour les métriques personnalisées dans la mesure où vous publiez les points de données bruts non résumés pour votre métrique personnalisée. Les statistiques sur les centiles ne sont pas disponibles pour les métriques lorsque l'une des valeurs des métriques est un nombre négatif.

CloudWatch a besoin de points de données bruts pour calculer les percentiles. Si, au lieu de cela, vous publiez des données avec un ensemble de statistiques, vous ne pouvez récupérer de statistiques relatives aux centiles pour ces données que si l'une des conditions suivantes est vraie :

- La SampleCount valeur de l'ensemble de statistiques est 1 et Min, Max et Sum sont tous égaux.
- Les valeurs Min et Max sont égales, et Sum est égale à Min multiplié par SampleCount.

Les AWS services suivants incluent des mesures qui prennent en charge les statistiques par centiles.

- API Gateway

- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing
- Kinesis
- Amazon RDS

CloudWatch prend également en charge les moyennes ajustées et d'autres statistiques de performance, qui peuvent être utilisées de la même manière que les percentiles. Pour plus d'informations, consultez CloudWatch définitions des statistiques.

alertes

Vous pouvez utiliser une alerte pour déclencher automatiquement des actions de votre part. Une alerte surveille une métrique unique sur une période de temps définie et exécute une ou plusieurs actions spécifiées en fonction de la valeur de la métrique par rapport à un seuil sur la période. L'action est une notification envoyée à une rubrique Amazon SNS ou à une stratégie Auto Scaling. Vous pouvez également ajouter des alertes aux tableaux de bord.

Les alarmes déclenchent des actions uniquement pour les changements d'état prolongés. CloudWatch les alarmes n'appellent pas d'actions simplement parce qu'elles sont dans un état particulier. L'état doit avoir changé et avoir été maintenu pendant un nombre de périodes spécifié.

Lors de la création d'une alerte, sélectionnez une période de surveillance d'alerte supérieure ou égale à la résolution de la métrique. Par exemple, une surveillance basique pour Amazon EC2 fournit des métriques pour vos instances toutes les 5 minutes. Lorsque vous définissez une alerte sur une métrique de surveillance basique, sélectionnez une période d'au moins 300 secondes (5 minutes). La surveillance détaillée pour Amazon EC2 fournit des métriques pour vos instances avec une résolution toutes les minutes. Lorsque vous définissez une alerte sur une métrique de surveillance détaillée, sélectionnez une période d'au moins 60 secondes (1 minute).

Si vous définissez une alerte sur une métrique haute résolution, vous pouvez spécifier une alerte haute résolution avec une période de 10 secondes ou de 30 secondes ou vous pouvez définir une alerte régulière avec une période correspondant à n'importe quel multiple de 60 secondes. Les frais engendrés par des alertes haute résolution sont plus élevés. Pour plus d'informations sur les métriques haute résolution, consultez Publier des métriques personnalisées .

Pour plus d'informations, consultez Utilisation des CloudWatch alarmes Amazon et Créer une alerte à partir d'une métrique sur un graphique.

Facturation et coûts

Pour obtenir des informations complètes sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Pour obtenir des informations qui peuvent vous aider à analyser votre facture et éventuellement à optimiser et réduire les coûts, consultez CloudWatch facturation et coût.

CloudWatch Ressources Amazon

Les ressources connexes suivantes peuvent s'avérer utiles lors de l'utilisation de ce service.

Ressource	Description
CloudWatchFAQ Amazon	La FAQ couvre les principales questions posées par les développeurs concernant ce produit.
AWS Centre pour développeurs	Un point de départ central pour trouver de la documenta tion, des exemples de code, des notes de version et d'autres informations qui vous aideront à créer des applications innovantes AWS.
AWS Management Console	La console vous permet d'exécuter la plupart des fonctions d'Amazon CloudWatch et de diverses autres AWS offres sans programmation.
Forums CloudWatch de discussion Amazon	Forum communautaire permettant aux développeurs de discuter de questions techniques liées à Amazon CloudWatch.
AWS Support	La plateforme de création et de gestion de vos AWS Support dossiers. Comprend également des liens vers d'autres ressources utiles, telles que des forums, des FAQ techniques, l'état de santé du service et AWS Trusted Advisor.
Informations sur les CloudWatch produits Amazon	La page Web principale contenant des informations sur Amazon CloudWatch.

Ressource	Description
Contactez-nous	Point de contact central pour les demandes concernant la AWS facturation, le compte, les événements, les abus, etc.

Configuration

Pour utiliser Amazon, CloudWatch vous avez besoin d'un AWS compte. Votre AWS compte vous permet d'utiliser des services (par exemple, Amazon EC2) pour générer des métriques que vous pouvez consulter dans la CloudWatch console, une point-and-click interface Web. En outre, vous pouvez installer et configurer l'interface de ligne de AWS commande (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 https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

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

   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. Pour des raisons de sécurité, attribuez un accès administratif à un utilisateur et utilisez uniquement l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

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

Création d'un utilisateur doté d'un accès administratif

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

Sécurisez votre Utilisateur racine d'un compte AWS

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

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

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

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

Connectez-vous en tant qu'utilisateur disposant 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.

Attribuer l'accès à des utilisateurs supplémentaires

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme aux meilleures pratiques en matière d'application des autorisations du moindre privilège.

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

2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

   Pour obtenir des instructions, voir Ajouter des groupes dans le guide de AWS IAM Identity Center l'utilisateur.

Connectez-vous à la CloudWatch console Amazon

Pour vous connecter à la CloudWatch console Amazon

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Si nécessaire, utilisez la barre de navigation pour remplacer la région par la région dans laquelle vous disposez de vos AWS ressources.
3. Même si c'est la première fois que vous utilisez la CloudWatch console, Your Metrics peut déjà fournir des statistiques, car vous avez utilisé un AWS produit qui envoie automatiquement des métriques à Amazon CloudWatch gratuitement. D'autres services nécessitent d'activer des métriques.

   Si vous n'avez pas d'alarme, la section Your Alarms (Vos alarmes) contiendra un bouton Create Alarm (Créer une alarme).

Configurez le AWS CLI

Vous pouvez utiliser la AWS CLI ou l'Amazon CloudWatch CLI pour exécuter des CloudWatch commandes. Notez que le AWS CLI remplace la CloudWatch CLI ; nous incluons de nouvelles CloudWatch fonctionnalités uniquement dans le AWS CLI.

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.

Pour plus d'informations sur l'installation et la configuration de l' CloudWatch interface de ligne de commande Amazon, consultez la section Configurer l'interface de ligne de commande dans le manuel Amazon CloudWatch CLI Reference.

Commencer à utiliser Amazon CloudWatch

Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

La page CloudWatch d'accueil de l'aperçu apparaît.

La présentation affiche les éléments suivants, actualisés automatiquement.

- Les alarmes par AWS service affichent la liste des AWS services que vous utilisez dans votre compte, ainsi que l'état des alarmes associées à ces services. À côté de cela, deux ou quatre alarmes s'affichent dans votre compte. Le nombre dépend du nombre de AWS services que vous utilisez. Les alarmes affichés sont celles se trouvant à l'état ALARME ou celles ayant changé d'état le plus récemment.

  Ces zones supérieures vous permettent d'évaluer rapidement l'état de vos AWS services, en visualisant l'état des alarmes de chaque service et les alarmes ayant récemment changé d'état. Elles vous permettent de surveiller et de diagnostiquer rapidement les problèmes.
- Sous ces zones se trouve le tableau de bord par défaut, le cas échéant. Le tableau de bord par défaut est un tableau de bord personnalisé que vous avez créé et nommé CloudWatch-Devault. Il s'agit d'un moyen pratique d'ajouter des mesures relatives à vos propres services ou applications personnalisés à la page de présentation, ou de proposer des indicateurs clés supplémentaires provenant AWS des services que vous souhaitez le plus surveiller.

> **Note**
>
> Les tableaux de bord automatiques de la page d' CloudWatch accueil affichent uniquement les informations du compte courant, même s'il s'agit d'un compte de surveillance configuré pour l'observabilité CloudWatch entre comptes. Pour de plus amples informations sur la création de tableaux de bord entre comptes personnalisés, veuillez consulter CloudWatch tableau de bord d'observabilité entre comptes.

À partir de cette vue d'ensemble, vous pouvez consulter un tableau de bord AWS multiservices contenant les indicateurs de plusieurs services, ou vous concentrer sur un groupe de ressources ou un AWS service spécifique. Cela vous permet de réduire votre affichage à un sous-ensemble de ressources qui vous intéresse. Pour plus d'informations, consultez les sections suivantes.

Consultez le tableau de bord automatique prédéfini pour un service unique

Pour voir le tableau de bord automatique prédéfini pour un seul service

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

   La page d'accueil s'affiche.
2. Dans le volet de navigation de gauche, sélectionnez Tableaux de bord.
3. Choisissez l'onglet Tableaux de bord automatiques, puis choisissez le service que vous souhaitez voir.
4. Pour passer à l'affichage des alarmes pour ce service, cochez la case En alarme, Données insuffisantes ou OK en haut de l'écran où le nom du service est actuellement affiché.
5. Lors de la consultation des métriques, vous pouvez vous concentrer sur une métrique spécifique de plusieurs façons :

   a. Pour plus d'informations sur les métriques dans n'importe quel graphique, passez la souris sur le graphique et choisissez l'icône d'actions View in metrics (Afficher dans les métriques).

      Le graphique s'affiche dans un nouvel onglet, avec les métriques correspondantes répertoriées sous le graphique. Vous pouvez personnaliser votre affichage de ce graphique, en modifiant les métriques et les ressources affichées, les statistiques, la période, et d'autres facteurs, afin de mieux comprendre la situation actuelle.

b. Vous pouvez afficher des événements de journaux pour la plage de temps affichée dans le graphique. Cela peut vous aider à détecter les événements survenus dans votre infrastructure qui entraînent une modification inattendue de vos mesures.

   Pour afficher les événements de journaux, passez la souris sur le graphique, puis choisissez l'icône d'actions View in logs (Afficher dans les journaux).

   La vue CloudWatch Journaux apparaît dans un nouvel onglet et affiche la liste de vos groupes de journaux. Pour afficher les événements de journaux dans l'un de ces groupes de journaux s'étant produits au cours de la plage de temps illustrée dans le graphique d'origine, choisissez ce groupe de journaux.

6. Lors de la consultation des alarmes, vous pouvez vous concentrer sur une alarme spécifique de plusieurs façons :

   • Pour plus d'informations sur une alarme, passez votre souris sur l'alarme, choisissez l'icône d'actions View in alarms (Afficher dans les alarmes).

     La vue des alarmes s'affiche dans un nouvel onglet, indiquant une liste de vos alarmes, ainsi que les détails relatifs à l'alarme choisie. Pour afficher l'historique de cette alarme, choisissez l'onglet Historique.

7. Les alarmes sont toujours actualisées chaque minute. Pour actualiser la vue, choisissez l'icône d'actualisation (deux flèches courbées) en haut à droite de l'écran. Pour modifier le taux de rafraîchissement automatique d'éléments à l'écran autres que les alarmes, choisissez la flèche vers le bas en regard de l'icône d'actualisation et choisissez un taux de rafraîchissement. Vous pouvez également choisir de désactiver l'actualisation automatique.
8. Pour modifier la plage de temps affichée dans tous les graphiques et alarmes actuellement affichés, en regard de Plage de temps en haut de l'écran, choisissez la plage de temps. Pour effectuer votre sélection parmi davantage d'options de plages de temps que celles affichées par défaut, choisissez personnalisé.
9. Pour revenir au tableau de bord inter-services, choisissez Présentation dans la liste située en haut de l'écran qui affiche actuellement le service sur lequel vous vous concentrez.

   À partir de n'importe quel affichage, vous pouvez également choisir CloudWatchen haut de l'écran d'effacer tous les filtres et de revenir à la page d'aperçu.

Voir le tableau de bord multiservice prédéfini

Vous pouvez passer à l'écran du tableau de bord multiservice et interagir avec les tableaux de bord de tous les AWS services que vous utilisez. La CloudWatch console affiche vos tableaux de bord par ordre alphabétique et affiche une ou deux mesures clés sur chaque tableau de bord.

> **Note**
>
> Si vous utilisez cinq AWS services ou plus, la CloudWatch console n'affichera pas le tableau de bord interservices sur l'écran de présentation.

Pour ouvrir le tableau de bord inter-services

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

   Vous êtes redirigé vers la page de présentation.
2. Sur la page de présentation, sélectionnez la liste déroulante Overview (Présentation), puis sélectionnez Cross service dashboard (Tableau de bord inter-services).

   Vous êtes redirigé vers l'écran du tableau de bord inter-services.
3. (Facultatif) Si vous utilisez l'interface d'origine, faites défiler jusqu'à la section Cross-service dashboard (Tableau de bord inter-services), puis sélectionnez View Cross-service dashboard (Afficher le tableau de bord inter-services).

   Vous êtes redirigé vers l'écran du tableau de bord inter-services.
4. Vous pouvez vous concentrer sur un service spécifique de deux manières :

   a. Pour afficher davantage de métriques clés pour un service, choisissez son nom dans la liste en haut de l'écran, où Cross service dashboard (Tableau de bord inter-services) est actuellement affiché. Sinon, vous pouvez choisir View Service dashboard (Afficher le tableau de bord du service) en regard du nom du service.

      Un tableau de bord automatique pour ce service s'affiche, indiquant plusieurs métriques pour ce service. En outre, pour certains services, le bas du tableau de bord du service affiche les ressources associées à ce service. Vous pouvez choisir l'une de ces ressources dans cette console de service et vous concentrer davantage sur cette ressource.

b. Pour afficher toutes les alarmes associées à un service, choisissez le bouton à droite de l'écran en regard de ce nom de service. Le texte sur ce bouton indique le nombre d'alarmes créées dans ce service, et si l'une d'elle se trouve à l'état ALARME.

   Lorsque les alarmes sont affichées, plusieurs alarmes avec des paramètres similaires (telles que les dimensions, le seuil ou la période) peuvent être affichées en un seul graphique.

   Vous pouvez ensuite consulter les détails relatifs à une alarme et consulter l'historique de l'alarme. Pour cela, passez la souris sur le graphique de l'alarme, puis choisissez l'icône d'actions View in alarms (Afficher dans les alarmes).

   La vue des alarmes s'affiche dans un nouvel onglet du navigateur, indiquant une liste de vos alarmes, ainsi que les détails relatifs à l'alarme choisie. Pour afficher l'historique de cette alarme, choisissez l'onglet Historique.

5. Vous pouvez vous concentrer sur les ressources d'un groupe de ressources spécifique. Pour cela, choisissez le groupe de ressources dans la liste située en haut de la page où Toutes les ressources s'affiche.

   Pour plus d’informations, consultez Voir un tableau de bord prédéfini pour un groupe de ressources.

6. Pour modifier la plage de temps affichée dans tous les graphiques et alarmes actuellement affichés, sélectionnez la plage de votre choix en regard de Plage de temps en haut de l'écran. Choisissez personnalisé pour effectuer votre sélection parmi davantage d'options de plages de temps que celles affichées par défaut.
7. Les alarmes sont toujours actualisées chaque minute. Pour actualiser la vue, choisissez l'icône d'actualisation (deux flèches courbées) en haut à droite de l'écran. Pour modifier le taux de rafraîchissement automatique d'éléments à l'écran autres que les alarmes, choisissez la flèche vers le bas en regard de l'icône d'actualisation et choisissez le taux de rafraîchissement de votre choix. Vous pouvez également choisir de désactiver l'actualisation automatique.

Supprimer un service du tableau de bord multiservice

Vous pouvez empêcher les métriques d'un service de s'afficher dans le tableau de bord inter-services. Cela vous permet d'axer votre tableau de bord inter-services sur les services que vous souhaitez surveiller le plus.

Si vous supprimez un service du tableau de bord inter-services, les alarmes de ce service continueront de s'afficher dans les vues de vos alarmes.

Pour supprimer les métriques d'un service du tableau de bord inter-services

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

   La page d'accueil s'affiche.
2. En haut de la page, sous Présentation, choisissez le service à supprimer.

   La vue change pour afficher uniquement les métriques de ce service.
3. Choisissez Actions, puis désactivez la case à cocher Show on cross service dashboard (Afficher sur le tableau de bord inter-services).

Voir un tableau de bord prédéfini pour un groupe de ressources

Vous pouvez concentrer votre affichage pour indiquer les mesures et alarmes d'un seul groupe de ressources. L'utilisation de groupes de ressources vous permet d'employer des balises pour organiser les projets, de vous concentrer sur un sous-ensemble de votre architecture, ou de faire la distinction entre vos environnements de production et de développement. Ils vous permettent également de vous concentrer sur chacun de ces groupes de ressources dans la CloudWatch vue d'ensemble. Pour plus d'informations, consultez Qu'est-ce qu' AWS Resource Groups ?

Lorsque vous vous concentrez sur un groupe de ressources, l'affichage change afin d'indiquer uniquement les services dans lesquels vous avez des ressources balisées dans le cadre de ce groupe de ressources. La zone des alarmes récentes affiche uniquement les alarmes associées aux ressources faisant partie du groupe de ressources. De plus, si vous avez créé un tableau de bord nommé CloudWatch-Devault- ResourceGroupName, il est affiché dans la zone Tableau de bord par défaut.

Vous pouvez approfondir votre analyse en vous concentrant à la fois sur un seul AWS service et sur un groupe de ressources. La procédure suivante explique uniquement comment se concentrer sur un groupe de ressources.

Pour se concentrer sur un seul groupe de ressources

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. En haut de la page, où Toutes les ressources s'affiche, choisissez un groupe de ressources.
3. Pour afficher davantage de métriques associées à ce groupe de ressources, près du bas de l'écran, choisissez View cross service dashboard (Afficher le tableau de bord inter-services).

Le tableau de bord inter-services s'affiche, indiquant uniquement les services associés à ce groupe de ressources. Pour chaque service, une ou deux métriques clés s'affichent.

4. Pour modifier la plage de temps affichée dans tous les graphiques et alarmes actuellement affichés, pour Plage de temps en haut de l'écran, sélectionnez une plage. Pour effectuer votre sélection parmi davantage d'options de plages de temps que celles affichées par défaut, choisissez personnalisé.
5. Les alarmes sont toujours actualisées chaque minute. Pour actualiser la vue, choisissez l'icône d'actualisation (deux flèches courbées) en haut à droite de l'écran. Pour modifier le taux de rafraîchissement automatique d'éléments à l'écran autres que les alarmes, choisissez la flèche vers le bas en regard de l'icône d'actualisation et choisissez un taux de rafraîchissement. Vous pouvez également choisir de désactiver l'actualisation automatique.
6. Pour revenir à l'affichage des informations sur toutes les ressources de votre compte, près du haut de l'écran où le nom du groupe de ressources est actuellement affiché, choisissez Toutes les ressources.

Voir le tableau de bord multiservice prédéfini

Vous pouvez passer à l'écran du tableau de bord multiservice et interagir avec les tableaux de bord de tous les AWS services que vous utilisez. La CloudWatch console affiche vos tableaux de bord par ordre alphabétique et affiche une ou deux mesures clés pour chaque service.

> **Note**
>
> Si vous utilisez cinq AWS services ou plus, la CloudWatch console n'affichera pas le tableau de bord interservices sur l'écran de présentation.

Pour ouvrir le tableau de bord inter-services

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

   Vous êtes redirigé vers la page de présentation.
2. Sur la page de présentation, sélectionnez la liste déroulante Overview (Présentation), puis sélectionnez Cross service dashboard (Tableau de bord inter-services).

   Vous êtes redirigé vers l'écran du tableau de bord inter-services.

3. (Facultatif) Si vous utilisez l'interface d'origine, faites défiler jusqu'à la section Cross-service dashboard (Tableau de bord inter-services), puis sélectionnez View Cross-service dashboard (Afficher le tableau de bord inter-services).

   Vous êtes redirigé vers l'écran du tableau de bord inter-services.

4. Vous pouvez vous concentrer sur un service spécifique de deux manières :

   a. Pour afficher davantage de métriques clés pour un service, choisissez son nom dans la liste en haut de l'écran, où Cross service dashboard (Tableau de bord inter-services) est actuellement affiché. Sinon, vous pouvez choisir View Service dashboard (Afficher le tableau de bord du service) en regard du nom du service.

      Un tableau de bord automatique pour ce service s'affiche, indiquant plusieurs métriques pour ce service. En outre, pour certains services, le bas du tableau de bord du service affiche les ressources associées à ce service. Vous pouvez choisir l'une de ces ressources dans cette console de service et vous concentrer davantage sur cette ressource.

   b. Pour afficher toutes les alarmes associées à un service, choisissez le bouton à droite de l'écran en regard de ce nom de service. Le texte sur ce bouton indique le nombre d'alarmes créées dans ce service, et si l'une d'elle se trouve à l'état ALARME.

      Lorsque les alarmes sont affichées, plusieurs alarmes avec des paramètres similaires (telles que les dimensions, le seuil ou la période) peuvent être affichées en un seul graphique.

      Vous pouvez ensuite consulter les détails relatifs à une alarme et consulter l'historique de l'alarme. Pour cela, passez la souris sur le graphique de l'alarme, puis choisissez l'icône d'actions View in alarms (Afficher dans les alarmes).

      La vue des alarmes s'affiche dans un nouvel onglet du navigateur, indiquant une liste de vos alarmes, ainsi que les détails relatifs à l'alarme choisie. Pour afficher l'historique de cette alarme, choisissez l'onglet Historique.

5. Vous pouvez vous concentrer sur les ressources d'un groupe de ressources spécifique. Pour cela, choisissez le groupe de ressources dans la liste située en haut de la page où Toutes les ressources s'affiche.

   Pour plus d’informations, consultez Voir un tableau de bord prédéfini pour un groupe de ressources.

6. Pour modifier la plage de temps affichée dans tous les graphiques et alarmes actuellement affichés, sélectionnez la plage de votre choix en regard de Plage de temps en haut de l'écran.

Choisissez personnalisé pour effectuer votre sélection parmi davantage d'options de plages de temps que celles affichées par défaut.

7. Les alarmes sont toujours actualisées chaque minute. Pour actualiser la vue, choisissez l'icône d'actualisation (deux flèches courbées) en haut à droite de l'écran. Pour modifier le taux de rafraîchissement automatique d'éléments à l'écran autres que les alarmes, choisissez la flèche vers le bas en regard de l'icône d'actualisation et choisissez le taux de rafraîchissement de votre choix. Vous pouvez également choisir de désactiver l'actualisation automatique.

Empêcher un service de s'afficher dans le tableau de bord inter-services

Vous pouvez empêcher les métriques d'un service de s'afficher dans le tableau de bord inter-services. Cela vous permet d'axer votre tableau de bord inter-services sur les services que vous souhaitez surveiller le plus.

Si vous supprimez un service du tableau de bord inter-services, les alarmes de ce service continueront de s'afficher dans les vues de vos alarmes.

Pour supprimer les métriques d'un service du tableau de bord inter-services

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

   La page d'accueil s'affiche.

2. En haut de la page, sous Présentation, choisissez le service à supprimer.

   La vue change pour afficher uniquement les métriques de ce service.

3. Choisissez Actions, puis désactivez la case à cocher Show on cross service dashboard (Afficher sur le tableau de bord inter-services).

Consultez un tableau de bord prédéfini pour un service unique AWS

Sur la page d' CloudWatch accueil, vous pouvez concentrer l'affichage sur un seul AWS service. Vous pouvez approfondir votre analyse en vous concentrant à la fois sur un seul AWS service et sur un groupe de ressources. La procédure suivante montre uniquement comment se concentrer sur un AWS service.

Pour se concentrer sur un seul service

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

   La page d'accueil s'affiche.
2. Pour Vue d'ensemble, où la vue d'ensemble est actuellement affichée dans le menu déroulant, choisissez Tableaux de bord des services.
3. Choisissez le service sur lequel vous souhaitez vous concentrer.

   La vue change pour afficher des graphiques de métriques clés du service sélectionné.
4. Pour passer à l'affichage des alarmes pour ce service, cochez la case En alarme, Données insuffisantes ou OK en haut de l'écran où le nom du service est actuellement affiché.
5. Lors de la consultation des métriques, vous pouvez vous concentrer sur une métrique spécifique de plusieurs façons :

   a. Pour plus d'informations sur les métriques dans n'importe quel graphique, passez la souris sur le graphique et choisissez l'icône d'actions View in metrics (Afficher dans les métriques).

      Le graphique s'affiche dans un nouvel onglet, avec les métriques correspondantes répertoriées sous le graphique. Vous pouvez personnaliser votre affichage de ce graphique, en modifiant les métriques et les ressources affichées, les statistiques, la période, et d'autres facteurs, afin de mieux comprendre la situation actuelle.

   b. Vous pouvez afficher des événements de journaux pour la plage de temps affichée dans le graphique. Cela peut vous aider à détecter les événements survenus dans votre infrastructure qui entraînent une modification inattendue de vos mesures.

      Pour afficher les événements de journaux, passez la souris sur le graphique, puis choisissez l'icône d'actions View in logs (Afficher dans les journaux).

      La vue CloudWatch Journaux apparaît dans un nouvel onglet et affiche la liste de vos groupes de journaux. Pour afficher les événements de journaux dans l'un de ces groupes de journaux s'étant produits au cours de la plage de temps illustrée dans le graphique d'origine, choisissez ce groupe de journaux.
6. Lors de la consultation des alarmes, vous pouvez vous concentrer sur une alarme spécifique de plusieurs façons :

   - Pour plus d'informations sur une alarme, passez votre souris sur l'alarme, choisissez l'icône d'actions View in alarms (Afficher dans les alarmes).

La vue des alarmes s'affiche dans un nouvel onglet, indiquant une liste de vos alarmes, ainsi que les détails relatifs à l'alarme choisie. Pour afficher l'historique de cette alarme, choisissez l'onglet Historique.

7. Les alarmes sont toujours actualisées chaque minute. Pour actualiser la vue, choisissez l'icône d'actualisation (deux flèches courbées) en haut à droite de l'écran. Pour modifier le taux de rafraîchissement automatique d'éléments à l'écran autres que les alarmes, choisissez la flèche vers le bas en regard de l'icône d'actualisation et choisissez un taux de rafraîchissement. Vous pouvez également choisir de désactiver l'actualisation automatique.
8. Pour modifier la plage de temps affichée dans tous les graphiques et alarmes actuellement affichés, en regard de Plage de temps en haut de l'écran, choisissez la plage de temps. Pour effectuer votre sélection parmi davantage d'options de plages de temps que celles affichées par défaut, choisissez personnalisé.
9. Pour revenir au tableau de bord inter-services, choisissez Présentation dans la liste située en haut de l'écran qui affiche actuellement le service sur lequel vous vous concentrez.

   À partir de n'importe quel affichage, vous pouvez également choisir CloudWatchen haut de l'écran d'effacer tous les filtres et de revenir à la page d'aperçu.

Voir un tableau de bord prédéfini pour un groupe de ressources

Vous pouvez concentrer votre affichage pour indiquer les mesures et alarmes d'un seul groupe de ressources. L'utilisation de groupes de ressources vous permet d'employer des balises pour organiser les projets, de vous concentrer sur un sous-ensemble de votre architecture, ou de faire la distinction entre vos environnements de production et de développement. Ils vous permettent également de vous concentrer sur chacun de ces groupes de ressources dans la CloudWatch vue d'ensemble. Pour plus d'informations, consultez Qu'est-ce qu' AWS Resource Groups ?

Lorsque vous vous concentrez sur un groupe de ressources, l'affichage change afin d'indiquer uniquement les services dans lesquels vous avez des ressources balisées dans le cadre de ce groupe de ressources. La zone des alarmes récentes affiche uniquement les alarmes associées aux ressources faisant partie du groupe de ressources. De plus, si vous avez créé un tableau de bord nommé CloudWatch-Devault- ResourceGroupName, il est affiché dans la zone Tableau de bord par défaut.

Vous pouvez approfondir votre analyse en vous concentrant à la fois sur un seul AWS service et sur un groupe de ressources. La procédure suivante décrit simplement comment se concentrer sur un groupe de ressources.

Pour se concentrer sur un seul groupe de ressources

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. En haut de la page, où Toutes les ressources s'affiche, choisissez un groupe de ressources.
3. Pour afficher davantage de métriques associées à ce groupe de ressources, près du bas de l'écran, choisissez View cross service dashboard (Afficher le tableau de bord inter-services).

   Le tableau de bord inter-services s'affiche, indiquant uniquement les services associés à ce groupe de ressources. Pour chaque service, une ou deux métriques clés s'affichent.
4. Pour modifier la plage de temps affichée dans tous les graphiques et alarmes actuellement affichés, pour Plage de temps en haut de l'écran, sélectionnez une plage. Pour effectuer votre sélection parmi davantage d'options de plages de temps que celles affichées par défaut, choisissez personnalisé.
5. Les alarmes sont toujours actualisées chaque minute. Pour actualiser la vue, choisissez l'icône d'actualisation (deux flèches courbées) en haut à droite de l'écran. Pour modifier le taux de rafraîchissement automatique d'éléments à l'écran autres que les alarmes, choisissez la flèche vers le bas en regard de l'icône d'actualisation et choisissez un taux de rafraîchissement. Vous pouvez également choisir de désactiver l'actualisation automatique.
6. Pour revenir à l'affichage des informations sur toutes les ressources de votre compte, près du haut de l'écran où le nom du groupe de ressources est actuellement affiché, choisissez Toutes les ressources.

CloudWatch facturation et coût

Cette section décrit comment les CloudWatch fonctionnalités d'Amazon génèrent des coûts. Il fournit également des méthodes qui peuvent vous aider à analyser, optimiser et réduire les CloudWatch coûts. Dans cette section, nous faisons parfois référence à la tarification lorsque nous décrivons les CloudWatch fonctionnalités. Pour plus d'informations sur les tarifs, consultez CloudWatch les tarifs Amazon.

Rubriques

- Analysez les données de CloudWatch coûts et d'utilisation avec Cost Explorer
- Analysez les données de CloudWatch coûts et d'utilisation avec AWS Cost and Usage Report s et Athena
- Bonnes pratiques pour l'optimisation de vos coûts

Analysez les données de CloudWatch coûts et d'utilisation avec Cost Explorer

Avec AWS Cost Explorer, vous pouvez visualiser et analyser les données de coûts et d'utilisation Services AWS au fil du temps, notamment CloudWatch. Pour plus d'informations, consultez la section Mise en route avec AWS Cost Explorer.

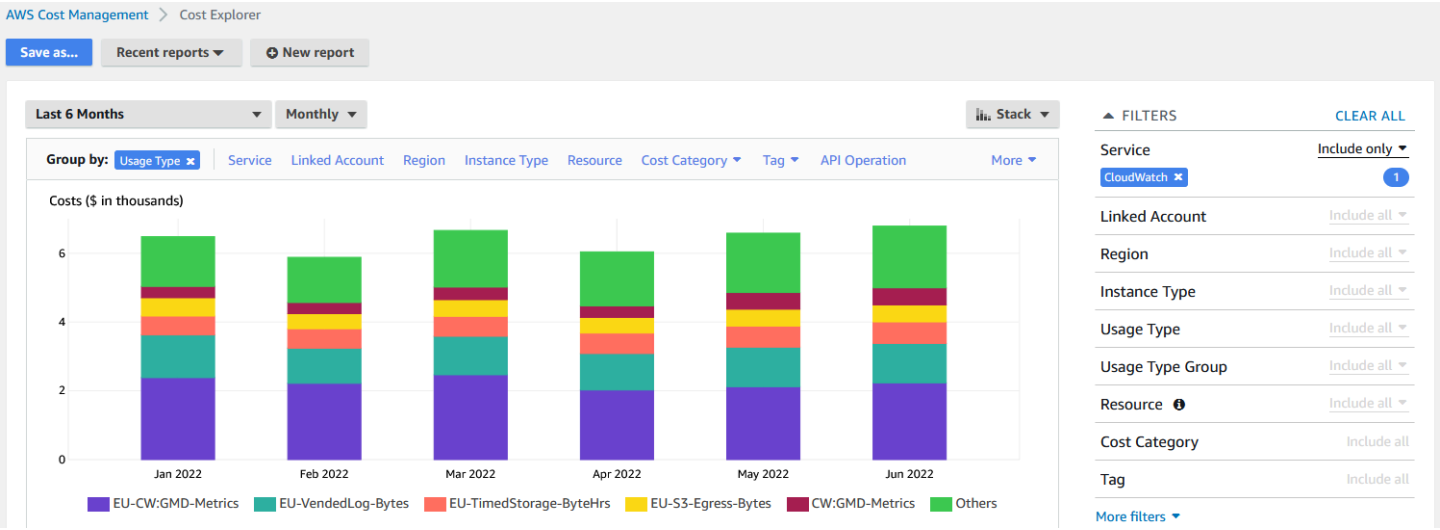
La procédure suivante décrit comment utiliser Cost Explorer pour visualiser et analyser les données de CloudWatch coûts et d'utilisation.

Pour visualiser et analyser les données relatives aux CloudWatch coûts et à l'utilisation

1. Connectez-vous à la console Cost Explorer à l'adresse https://console.aws.amazon.com/cost-management/home#/custom.
2. Sous FILTRES, pour Service, sélectionnez CloudWatch.
3. Pour Group by (Regrouper par), choisissez Usage Type (Type d'utilisation). Vous pouvez également regrouper vos résultats selon d'autres catégories, telles que les suivantes :
   - API Operation : découvrez quelles opérations d'API ont généré le plus de coûts.

- Region : découvrez quelles régions ont généré le plus de coûts.

L'image suivante montre un exemple des coûts générés par les CloudWatch fonctionnalités sur une période de six mois.



Pour savoir quelles CloudWatch fonctionnalités ont généré le plus de coûts, examinez les valeurs de`UsageType`. Par exemple, `EU-CW:GMD-Metrics` représente les coûts générés par les demandes d'API CloudWatch en masse.

> **Note**
>
> Les chaînes pour `UsageType` correspondent à des fonctionnalités et à des régions spécifiques. Par exemple, la première partie de `EU-CW:GMD-Metrics` (EU) correspond à la région Europe (Irlande), et la seconde partie de `EU-CW:GMD-Metrics` (`GMD-Metrics`) correspond aux demandes d'API CloudWatch en masse.
>
> La chaîne entière pour `UsageType` peut être formatée comme suit : `<Region>-CW:<Feature>` ou `<Region>-<Feature>`.
>
> Pour améliorer la lisibilité, les chaînes pour `UsageType` dans les tableaux de ce document ont été raccourcies à leurs suffixes de chaîne. Par exemple, `EU-CW:GMD-Metrics` est abrégée en `GMD-Metrics`.

Le tableau suivant inclut les noms de chaque CloudWatch fonctionnalité, répertorie les noms de chaque sous-fonctionnalité et répertorie les chaînes pour`UsageType`.

CloudWatch fonctionnalité	CloudWatch sous-fonctionnalité	**UsageType**
CloudWatch métriques	Métriques personnalisées	`MetricMonitorUsage`
	Surveillance détaillée	`MetricMonitorUsage`
	Métriques intégrées	`MetricMonitorUsage`
CloudWatch Demandes d'API	Demandes d'API	`Requests`
	En vrac (Obtenir)	`GMD-Metrics`
	Contributor Insights	`GIRR-Metrics`
	Instantané d'image bitmap	`GMWI-Metrics`
CloudWatch flux métriques	Flux de métriques	`MetricStreamUsage`
CloudWatch tableaux de bord	Tableau de bord avec 50 métriques ou moins	`DashboardsUsageHour-Basic`
	Tableau de bord avec plus de 50 métriques	`DashboardsUsageHour`
CloudWatch alarmes	Standard (alarme de métrique)	`AlarmMonitorUsage`
	Haute résolution (alarme métrique)	`HighResAlarmMonitorUsage`
	Alarme de requête Metrics Insights	

CloudWatch fonctionnalité	CloudWatch sous-fonctionnalité	**UsageType**
		`MetricInsightAlarmUsage`
	Composite (alarme agrégée)	`CompositeAlarmMonitorUsage`
CloudWatch Signaux d'application	Signaux d'application	`Application-Signals`
CloudWatch journaux personnalisés	Collecter (ingérer)	`DataProcessing-Bytes`
	Store (archive)	`TimedStorage-ByteHrs`
	Analyser (requête)	`DataScanned-Bytes`
CloudWatch Journaux d'accès peu fréquents	Collecter (ingérer)	`DataProcessingIA-Bytes`
CloudWatch journaux vendus	Livraison (Amazon CloudWatch Logs)	`VendedLog-Bytes`
	Livraison (CloudWatch journaux, journaux d'accès peu fréquents)	`VendedLogIA-Bytes`
	Mise à disposition (Amazon Simple Storage Service)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`
	Livraison (Amazon Data Firehose)	`FH-Egress-Bytes`

CloudWatch fonctionnalité	CloudWatch sous-fonctionnalité	**`UsageType`**
Contributor Insights	CloudWatch Logs (règles)	`ContributorInsightRules`
	CloudWatch Journaux (événements)	`ContributorInsightEvents`
	Amazon DynamoDB (Règles)	`ContributorRulesManaged`
	DynamoDB (Événements)	`ContributorEventsManaged`
Canaries (Synthetics)	Exécuter	`Canary-runs`
Evidently	Evénements	`Evidently-event`
	Unités d'analyse	`Evidently-eau`
RUM	Evénements	`RUM-event`

Analysez les données de CloudWatch coûts et d'utilisation avec AWS Cost and Usage Report s et Athena

Une autre méthode d'analyse des données relatives aux CloudWatch coûts et à l'utilisation consiste à utiliser AWS Cost and Usage Report s avec Amazon Athena. AWS Cost and Usage Report s contiennent un ensemble complet de données sur les coûts et l'utilisation. Vous pouvez créer des rapports qui suivent vos coûts et votre utilisation, et vous pouvez les publier dans un compartiment S3 de votre choix. Vous pouvez également télécharger et supprimer vos rapports de

votre compartiment S3. Pour plus d'informations, voir Que sont les AWS Cost and Usage Report s ? dans le guide AWS Cost and Usage Report de l'utilisateur s.

> **Note**
>
> Il n'y a aucun frais pour l'utilisation AWS Cost and Usage Report de s. Vous ne payez pour le stockage que lorsque vous publiez vos rapports sur Amazon Simple Storage Service (Amazon S3). Pour de plus amples informations, veuillez consulter Quotas et restrictions dans le Guide de l'utilisateur AWS Cost and Usage Report.

Athena est un service de requêtes que vous pouvez utiliser avec AWS Cost and Usage Report s pour analyser les données de coûts et d'utilisation. Vous pouvez interroger vos rapports dans votre compartiment S3 sans avoir à les télécharger au préalable. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon Athena ? dans le Guide de l'utilisateur d'Amazon Athena. Pour de plus amples informations, veuillez consulter Qu'est-ce que Amazon Athena ? dans le Guide de l'utilisateur d'Amazon Athena. Pour en savoir plus sur la tarification, consultez la Tarification d'Amazon Athena.

La procédure suivante décrit le processus d'activation de AWS Cost and Usage Report s et d'intégration du service à Athena. La procédure contient deux exemples de requêtes que vous pouvez utiliser pour analyser les données de CloudWatch coût et d'utilisation.

> **Note**
>
> Vous pouvez utiliser l'un des exemples de requêtes de ce document. Tous les exemples de requêtes de ce document correspondent à une base de données nommée costandusagereport, et affiche les résultats pour le mois d'avril et l'année 2022. Vous pouvez modifier ces informations. Toutefois, avant d'exécuter une requête, assurez-vous que le nom de votre base de données correspond au nom de la base de données dans la requête.

Pour analyser les données de coûts et d'utilisation avec AWS Cost and Usage Report s et Athena

1. Activez AWS Cost and Usage Report s. Pour en savoir plus, consultez Créer des rapports de coûts et d'utilisation dans le Guide de l'utilisateur des AWS Cost and Usage Report s.

> **Tip**
>
> Lorsque vous créez vos rapports, veillez à sélectionner Inclure ID de ressource. Sinon, vos rapports n'incluront pas la colonne `line_item_resource_id`. Cette ligne vous aide à mieux identifier les coûts lors de l'analyse des données de coûts et d'utilisation.

2. AWS Cost and Usage Report Intégrez-nous à Athena. Pour plus d'informations, consultez la section Configuration d'Athena à l'aide de AWS CloudFormation modèles dans le guide de l'utilisateur AWS Cost and Usage Report.
3. Interrogez vos rapports de coûts et d'utilisation.

Exemple : requête Athena

Vous pouvez utiliser la requête suivante pour indiquer quelles CloudWatch fonctionnalités ont généré le plus de coûts pour un mois donné.

```
SELECT
CASE
-- Metrics
WHEN line_item_usage_type LIKE '%%MetricMonitorUsage%%' THEN 'Metrics (Custom, Detailed
 monitoring management portal EMF)'
WHEN line_item_usage_type LIKE '%%Requests%%' THEN 'Metrics (API Requests)'
WHEN line_item_usage_type LIKE '%%GMD-Metrics%%' THEN 'Metrics (Bulk API Requests)'
WHEN line_item_usage_type LIKE '%%MetricStreamUsage%%' THEN 'Metric Streams'
-- Dashboard
WHEN line_item_usage_type LIKE '%%DashboardsUsageHour%%' THEN 'Dashboards'
-- Alarms
WHEN line_item_usage_type LIKE '%%AlarmMonitorUsage%%' THEN 'Alarms (Standard)'
WHEN line_item_usage_type LIKE '%%HighResAlarmMonitorUsage%%' THEN 'Alarms (High
 Resolution)'
WHEN line_item_usage_type LIKE '%%MetricInsightAlarmUsage%%' THEN 'Alarms (Metrics
 Insights)'
WHEN line_item_usage_type LIKE '%%CompositeAlarmMonitorUsage%%' THEN 'Alarms
 (Composite)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessing-Bytes%%' THEN 'Logs (Collect - Data
 Ingestion)'
-- Logs
WHEN line_item_usage_type LIKE '%%DataProcessingIA-Bytes%%' THEN 'Infrequent Access
 Logs (Collect - Data Ingestion)'
```

```
WHEN line_item_usage_type LIKE '%%TimedStorage-ByteHrs%%' THEN 'Logs (Storage -
 Archival)'
WHEN line_item_usage_type LIKE '%%DataScanned-Bytes%%' THEN 'Logs (Analyze - Logs
 Insights queries)'
-- Vended Logs
WHEN line_item_usage_type LIKE '%%VendedLog-Bytes%%' THEN 'Vended Logs (Delivered to
 CW)'
WHEN line_item_usage_type LIKE '%%VendedLogIA-Bytes%%' THEN 'Vended Infrequent Access
 Logs (Delivered to CW)'
WHEN line_item_usage_type LIKE '%%FH-Egress-Bytes%%' THEN 'Vended Logs (Delivered to
 Kinesis FH)'
WHEN (line_item_usage_type LIKE '%%S3-Egress-Bytes%%') OR (line_item_usage_type LIKE '%
%S3-Egress-
ComprBytes%%') THEN 'Vended Logs (Delivered to S3)'
-- Other
WHEN line_item_usage_type LIKE '%%Application-Signals%%' THEN 'Application Signals'
WHEN line_item_usage_type LIKE '%%Canary-runs%%' THEN 'Synthetics'
WHEN line_item_usage_type LIKE '%%Evidently%%' THEN 'Evidently'
WHEN line_item_usage_type LIKE '%%RUM-event%%' THEN 'RUM'
ELSE 'Others'
END AS UsageType,
-- REGEXP_EXTRACT(line_item_resource_id,'^(?:.+?:){5}(.+)$',1) as ResourceID,
-- SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
GROUP BY
1
ORDER BY
TotalSpend DESC,
UsageType;
```

Exemple : requête Athena

Vous pouvez utiliser la requête suivante pour afficher les résultats pour `UsageType` et `Operation`. Cela vous montre comment les CloudWatch fonctionnalités ont généré des coûts. Les résultats indiquent également les valeurs pour `UsageQuantity` et `TotalSpend`, afin que vous puissiez voir vos coûts d'utilisation totaux.

> **Tip**
>
> Pour plus d'informations sur `UsageType`, ajoutez la ligne suivante à cette requête : `line_item_line_item_description`
> Cette ligne crée une colonne appelée Description.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
```

Bonnes pratiques pour l'optimisation de vos coûts

CloudWatch métriques

Nombre d'entre eux Services AWS, tels qu'Amazon Elastic Compute Cloud (Amazon EC2), Amazon S3 et Amazon Data Firehose, envoient automatiquement des métriques gratuitement à. CloudWatch Toutefois, les mesures regroupées dans les catégories suivantes peuvent entraîner des coûts supplémentaires :

- Mesures personnalisées, surveillance détaillée et métriques intégrées
- Demandes d'API
- Flux de métriques

Pour plus d'informations, consultez la section Utilisation CloudWatch des métriques Amazon.

Mesures personnalisées, surveillance détaillée et métriques intégrées

Métriques personnalisées

Vous pouvez créer des métriques personnalisées pour organiser des points de données dans n'importe quel ordre et à n'importe quelle fréquence.

Toutes les mesures personnalisées sont calculées au prorata de l'heure. Ils ne sont mesurés que lorsqu'ils sont envoyés à CloudWatch. Pour plus d'informations sur la CloudWatch tarification des métriques, consultez Amazon Pricing.

Le tableau suivant répertorie les noms des sous-fonctionnalités pertinentes pour les CloudWatch métriques. Le tableau comprend les chaînes pour `UsageType` et `Operation`, ce qui peut vous aider à analyser et à identifier les coûts liés aux métriques.

> **Note**
>
> Pour obtenir plus de détails sur les mesures répertoriées dans le tableau suivant lorsque vous interrogez les données de coût et d'utilisation auprès d'Athena, faites correspondre les chaînes de `Operation` avec les résultats affichés pour `line_item_operation`.

CloudWatchsous-fonctionnalité	**`UsageType`**	**`Operation`**	Objectif
Métriques personnalisées	`MetricMonitorUsage`	`MetricStorage`	Métriques personnalisées
Surveillance détaillée	`MetricMonitorUsage`	`MetricStorage:AWS/`*`{Service}`*	Surveillance détaillée
Métriques intégrées	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Consigne les métriques intégrées
Filtres de journaux	`MetricMonitorUsage`	`MetricStorage:AWS/CloudWatchLogs`	Filtres de métriques de groupes de journaux

Surveillance détaillée

CloudWatch propose deux types de surveillance :

- Surveillance de base

  La surveillance de base est gratuite et activée automatiquement pour tous les Services AWS qui prennent en charge la fonctionnalité.
- Surveillance détaillée

  Une surveillance détaillée entraîne des coûts et apporte différentes améliorations en fonction du Service AWS. Pour chaque Service AWS qui prend en charge la surveillance détaillée, vous pouvez choisir de l'activer pour ce service. Pour de plus amples informations, veuillez consulter Surveillance basique et détaillée.

> **Note**
>
> D'autres Services AWS prennent en charge la surveillance détaillée et peuvent faire référence à cette fonctionnalité sous un autre nom. Par exemple, la surveillance détaillée d'Amazon S3 est appelée métriques de demande.

À l'instar des mesures personnalisées, la surveillance détaillée est calculée au prorata de l'heure et mesurée uniquement lorsque les données sont envoyées à. CloudWatch Une surveillance détaillée génère des coûts en fonction du nombre de métriques envoyées à CloudWatch. Pour réduire les coûts, activez une surveillance détaillée uniquement lorsque cela est nécessaire. Pour plus d'informations sur le prix de la surveillance détaillée, consultez Amazon CloudWatch Pricing.

Exemple : requête Athena

Vous pouvez utiliser la requête suivante pour afficher les instances EC2 dont la supervision détaillée est activée.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_usage_type AS UsageType,
line_item_operation AS Operation,
line_item_resource_id AS ResourceID,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation='MetricStorage:AWS/EC2'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation,
line_item_line_item_description
```

```
ORDER BY line_item_operation
```

Métriques intégrées

Grâce au format métrique CloudWatch intégré, vous pouvez ingérer les données de l'application sous forme de données de journal, afin de générer des mesures exploitables. Pour plus d'informations, voir Ingestion de journaux à forte cardinalité et génération de métriques avec le format de métrique CloudWatch intégré.

Les métriques intégrées génèrent des coûts en fonction du nombre de journaux ingérés, du nombre de journaux archivés et du nombre de métriques personnalisées générées.

Le tableau suivant répertorie les noms des sous-fonctionnalités pertinentes pour le format métrique CloudWatch intégré. Le tableau comprend les chaînes pour `UsageType` et `Operation`, ce qui peut vous aider à analyser et à identifier les coûts.

CloudWatch sous-fonctionnalité	**`UsageType`**	**`Operation`**	Objectif
Métriques personnalisées	`MetricMonitorUsage`	`MetricStorage:AWS/Logs-EMF`	Consigne les métriques intégrées
Ingestion journaux	`DataProcessing-Bytes`	`PutLogEvents`	Charge un lot d'événements de journaux dans le groupe de journaux ou le flux de journaux spécifié
Archivage des journaux	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Stocke les logs par heure et les logs par octet dans CloudWatch Logs

Pour analyser les coûts, utilisez AWS Cost and Usage Report s avec Athena afin d'identifier les indicateurs qui génèrent des coûts et de déterminer comment les coûts sont générés.

Pour tirer le meilleur parti des coûts générés par le format de métrique CloudWatch intégré, évitez de créer des métriques basées sur des dimensions à cardinalité élevée. De cette façon, vous CloudWatch ne créez pas de métrique personnalisée pour chaque combinaison de dimensions unique. Pour de plus amples informations, veuillez consulter Dimensions.

Si vous utilisez CloudWatch Container Insights pour tirer parti du format métrique intégré, vous pouvez utiliser AWS Distro for Open Telemetry comme alternative pour tirer le meilleur parti des coûts liés aux métriques. Container Insight vous permet de collecter, regrouper et récapituler les métriques et les journaux de vos applications et microservices conteneurisés. Lorsque vous activez Container Insights, l' CloudWatch agent envoie vos journaux à CloudWatch, afin qu'il puisse les utiliser pour générer des métriques intégrées. Cependant, l' CloudWatch agent envoie uniquement un nombre fixe de métriques à CloudWatch, et vous êtes facturé pour toutes les métriques disponibles, y compris celles que vous n'utilisez pas. Avec AWS Distro for Open Telemetry, vous pouvez configurer et personnaliser les métriques et les dimensions à envoyer. CloudWatch Cela vous permet de réduire le volume de données et les coûts générés par Container Insights. Pour plus d'informations, consultez les ressources suivantes :

- Utilisation de Container Insights
- AWS Distro pour Open Telemetry

Demandes d'API

CloudWatch possède les types de demandes d'API suivants :

- Demandes d'API
- En vrac (Obtenir)
- Contributor Insights
- Instantané d'image bitmap

Les demandes d'API génèrent des coûts en fonction du type de demande et du nombre de métriques demandées.

Le tableau suivant répertorie les types de requêtes d'API et comprend les chaînes pour `UsageType` et `Operation`, ce qui peut vous aider à analyser et à identifier les coûts liés aux API.

Type de demande d'API	**UsageType**	**Operation**	Objectif
Demandes d'API	Requests	GetMetric Statistics	Obtient des statistiq ues pour la métrique spécifiée.
	Requests	ListMetrics	Répertorie les mesures spécifiées
	Requests	PutMetricData	Publie des points de données métriques sur CloudWatch
	Requests	GetDashboard	Affiche les détails des tableaux de bord spécifiés
	Requests	ListDashboards	Répertorie les tableaux de bord de votre compte
	Requests	PutDashboard	Crée ou met à jour un tableau de bord
	Requests	DeleteDas hboards	Supprime tous les tableaux de bord spécifiés
En vrac (Obtenir)	GMD-Metrics	GetMetricData	Récupère les valeurs CloudWatch métriques
Contributor Insights	GIRR-Metrics	GetInsigh tRuleReport	Renvoie des données de séries temporell es collectées par une règle Contributor Insights

Type de demande d'API	**`UsageType`**	**`Operation`**	Objectif
Instantané d'image bitmap	`GMWI-Metrics`	`GetMetric WidgetImage`	Récupère un instantané d'une ou plusieurs CloudWatch métriques sous forme d'image bitmap

Pour analyser les coûts, utilisez Cost Explorer et regroupez vos résultats par Opération API.

Les coûts des demandes d'API varient, et vous encourez des frais lorsque vous dépassez le nombre d'appels d'API qui vous sont fournis dans le cadre de la limite du niveau AWS gratuit.

> **Note**
>
> `GetMetricData`et `GetMetricWidgetImage` ne sont pas inclus dans la limite du niveau AWS gratuit. Pour plus d'informations, consultez la section Utilisation du niveau AWS gratuit dans le guide de AWS Billing l'utilisateur.

Les demandes d'API qui génèrent généralement des coûts sont les suivantes : requêtes `Put` et `Get`.

`PutMetricData`

`PutMetricData` génère des coûts chaque fois qu'il est appelé et peut entraîner des coûts importants en fonction du cas d'utilisation. Pour plus d'informations, consultez PutMetricDatale Amazon CloudWatch API Reference.

Pour tirer le meilleur parti des coûts générés par `PutMetricData`, regroupez davantage de données dans vos appels d'API. Selon votre cas d'utilisation, pensez à utiliser CloudWatch des journaux ou le format de métrique CloudWatch intégré pour injecter des données métriques. Pour plus d'informations, consultez les ressources suivantes :

- Qu'est-ce qu'Amazon CloudWatch Logs ? dans le guide de l'utilisateur d'Amazon CloudWatch Logs
- Ingestion de journaux à haute cardinalité et génération de métriques avec CloudWatch un format de métrique intégré

- Réduire les coûts et se concentrer sur nos clients grâce aux métriques personnalisées CloudWatch intégrées à Amazon

GetMetricData

`GetMetricData` peut également générer des coûts importants. Les cas d'utilisation courants qui augmentent les coûts impliquent des outils de surveillance tiers qui extraient des données pour générer des informations. Pour plus d'informations, consultez GetMetricDatale Amazon CloudWatch API Reference.

Pour réduire les coûts générés par `GetMetricData`, envisagez de n'extraire que les données surveillées et utilisées, ou envisagez d'extraire des données moins souvent. Selon votre cas d'utilisation, vous pourriez utiliser des flux de métriques au lieu de`GetMetricData`, afin que vous puissiez transmettre des données en temps quasi réel à des tiers à moindre coût. Pour plus d'informations, consultez les ressources suivantes :

- Utilisation des flux de métriques
- CloudWatch Streams métriques - Envoyez AWS des métriques à vos partenaires et à vos applications en temps réel

GetMetricStatistics

Selon votre cas d'utilisation, vous pourriez utiliser `GetMetricStatistics` au lieu de `GetMetricData`. Avec `GetMetricData`, vous pouvez récupérer des données rapidement et à grande échelle. Toutefois, elle `GetMetricStatistics` est incluse dans la limite du niveau AWS gratuit pour un million de demandes d'API, ce qui peut vous aider à réduire les coûts si vous n'avez pas besoin de récupérer autant de métriques et de points de données par appel. Pour plus d'informations, consultez les ressources suivantes :

- GetMetricStatisticsdans le Amazon CloudWatch API Reference
- Dois-je utiliser GetMetricData ou GetMetricStatistics ?

> **Note**
>
> Les appelants externes effectuent des appels d'API. À l'heure actuelle, le seul moyen d'identifier ces appelants est d'adresser une demande d'assistance technique à l' CloudWatch équipe et de demander des informations les concernant. Pour plus d'informations sur la

création d'une demande de support technique, voir Comment puis-je obtenir une assistance technique AWS ? .

CloudWatch flux métriques

Avec les CloudWatch flux de mesures, vous pouvez envoyer des métriques en continu vers des AWS destinations et vers des destinations de fournisseurs de services tiers.

Les flux de métriques génèrent des coûts par le nombre de mises à jour de métriques. Les mises à jour des métriques incluent toujours des valeurs pour les statistiques suivantes :

- `Minimum`
- `Maximum`
- `Sample Count`
- `Sum`

Pour de plus amples informations, veuillez consulter Statistiques pouvant être diffusées.

Pour analyser les coûts générés par les flux CloudWatch métriques, utilisez AWS Cost and Usage Report s avec Athena. De cette façon, vous pouvez identifier les flux de mesures qui génèrent des coûts et déterminer comment les coûts sont générés.

Exemple : requête Athena

Vous pouvez utiliser la requête suivante pour suivre les flux de métriques qui génèrent des coûts par Amazon Resource Name (ARN).

```
SELECT
SPLIT_PART(line_item_resource_id,'/',2) AS "Stream Name",
line_item_resource_id as ARN,
SUM(CAST(line_item_unblended_cost AS decimal(16,2))) AS TotalSpend
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
```

```
-- AND line_item_usage_account_id = '123456789012' - If you want to filter on a
 specific account, you can
remove this comment at the beginning of the line and specify an AWS account.
AND line_item_usage_type LIKE '%%MetricStreamUsage%%'
GROUP BY line_item_resource_id
ORDER BY TotalSpend DESC
```

Pour réduire les coûts générés par les CloudWatch flux de mesures, diffusez uniquement les indicateurs qui apportent de la valeur à votre entreprise. Vous pouvez également arrêter ou suspendre tout flux de métriques que vous n'utilisez pas.

CloudWatch alarmes

Avec les CloudWatch alarmes, vous pouvez créer des alarmes basées sur une seule métrique, des alarmes basées sur une requête Metrics Insights et des alarmes composites qui surveillent les autres alarmes.

> **Note**
>
> Les coûts des alarmes de métriques et alarmes composites sont calculés au prorata par heure. Vous n'engagez des coûts pour vos alarmes que tant que vos alarmes existent. Pour optimiser les coûts, veillez à ne pas laisser derrière vous des alarmes mal configurées ou de faible valeur. Pour y parvenir, vous pouvez automatiser le nettoyage des CloudWatch alarmes dont vous n'avez plus besoin. Pour plus d'informations, consultez Automatiser Amazon CloudWatch Alarm Cleanup à grande échelle

Alarmes de métrique

Les paramètres de résolution des alertes métriques sont les suivants :

- Standard (évalué toutes les 60 secondes)
- Haute résolution (évalué toutes les 10 secondes)

Lorsque vous créez une alarme de métrique, vos coûts sont basés sur le paramètre de résolution de votre alarme et le nombre de métriques auxquelles votre alarme fait référence. Par exemple, une alarme de métrique qui fait référence à une seule métrique entraîne un coût horaire pour une alarme de métrique. Pour plus d'informations, consultez la section Utilisation des CloudWatch alarmes Amazon.

Si vous créez une alarme de métrique qui contient une expression mathématique de métrique, faisant référence à plusieurs métrique, vous engagez un coût pour chaque métrique d'alarme référencée dans l'expression mathématique de la métrique. Pour plus d'informations sur la création d'une alarme métrique contenant une expression mathématique métrique, voir Création CloudWatch d'une alarme basée sur une expression mathématique métrique.

Si vous créez une alarme de détection d'anomalie, dans laquelle votre alarme analyse les données de métriques passées afin de créer un modèle de valeurs attendues, vous engagez un coût pour chaque alarme de métrique référencée dans votre alarme, plus deux métriques supplémentaires, pour les métriques de bande supérieure et inférieure créées par le modèle de détection d'anomalies. Pour plus d'informations sur la création d'une alarme de détection d'anomalie, voir Création d'une CloudWatch alarme basée sur la détection d'anomalies.

Alarmes de requête Metrics Insights

Les alarmes de requête Metrics Insights sont un type spécifique d'alarme métrique, uniquement disponible avec une résolution standard (évaluée toutes les 60 secondes).

Lorsque vous créez une alarme de requête Metrics Insights, vos coûts sont basés sur le nombre de métriques analysées par la requête que votre alarme référence. Par exemple, une alarme de requête Metrics Insights qui fait référence à une requête dont le filtre correspond à dix métriques entraîne un coût horaire de dix métriques analysées. Pour plus d'informations, consultez l'exemple de tarification sur Amazon CloudWatch Pricing.

Si vous créez une alarme qui contient à la fois une requête Metrics Insights et une expression mathématique de métrique, elle est signalée comme une alarme de requête Metrics Insights. Si votre alarme contient une expression mathématique de métrique qui fait référence à d'autres métriques en plus de celles analysées par la requête Metrics Insights, vous encourrez un coût supplémentaire pour chaque métrique d'alarme référencée dans l'expression mathématique de métrique. Pour plus d'informations sur la création d'une alarme métrique contenant une expression mathématique métrique, voir Création CloudWatch d'une alarme basée sur une expression mathématique métrique.

Alarmes composites

Les alarmes composites contiennent des expressions de règles qui spécifient comment elles doivent évaluer les états des autres alarmes pour déterminer leurs propres états. Les alarmes composites ont un coût horaire standard, quel que soit le nombre d'autres alarmes qu'elles évaluent. Les alarmes auxquelles les alarmes composites font référence dans les expressions de règles entraînent des coûts distincts. Pour plus d'informations, consultez Création d'une alerte composite.

Types d'utilisation d'alarme

Le tableau suivant répertorie les noms des sous-fonctionnalités pertinentes pour les CloudWatch alarmes. Le tableau comprend les chaînes pour `UsageType`, ce qui peut vous aider à analyser et à identifier les coûts liés aux alertes.

CloudWatchsous-fonctionnalité	**`UsageType`**
Alarme de métrique standard	`AlarmMonitorUsage`
Alarme métrique haute résolution	`HighResAlarmMonitorUsage`
Alarme de requête Metrics Insights	`MetricInsightAlarmUsage`
alerte composite	`CompositeAlarmMonitorUsage`

Réduction des coûts d'alarme

Pour optimiser les coûts générés par les alarmes mathématiques qui regroupent quatre mesures ou plus, vous pouvez agréger les données avant qu'elles ne soient envoyées à CloudWatch. De cette façon, vous pouvez créer une alerte pour une seule métrique au lieu d'une alerte qui agrège les données pour plusieurs métriques. Pour plus d'informations, consultez Publication de métriques personnalisées.

Pour optimiser les coûts générés par les alarmes des requêtes Metrics Insights, vous pouvez vous assurer que le filtre utilisé pour la requête correspond uniquement aux métriques que vous voulez surveiller.

Le meilleur moyen de réduire les coûts est de supprimer toutes les alertes inutiles ou inutilisées. Par exemple, vous pouvez supprimer les alarmes qui évaluent les métriques émises par AWS des ressources qui n'existent plus.

Exemple : vérifiez la présence d'alarmes dans l'état ***`INSUFFICIENT_DATA`*** avec ***`DescribeAlarms`***

Si vous supprimez une ressource, mais pas les alarmes de métrique que la ressource émet, les alarmes existent toujours et passent généralement à l'état `INSUFFICIENT_DATA`. Pour vérifier la présence d'alarmes dans `INSUFFICIENT_DATA` cet état, utilisez la commande suivante AWS Command Line Interface (AWS CLI).

```
$ aws cloudwatch describe-alarms –state-value INSUFFICIENT_DATA
```

Les autres moyens de réduire les coûts sont les suivants :

- Assurez-vous de créer des alertes pour les bonnes mesures.
- Assurez-vous qu'aucune alerte n'est activée dans les régions où vous ne travaillez pas.
- N'oubliez pas que, bien que les alarmes composites réduisent le bruit, elles génèrent également des coûts supplémentaires.
- Lorsque vous décidez de créer une alerte standard ou une alerte haute résolution, tenez compte de votre cas d'utilisation et de la valeur apportée par chaque type d'alerte.

CloudWatch Journaux

Amazon CloudWatch Logs possède les types de journaux suivants :

- Journaux personnalisés (journaux que vous créez pour vos applications)
- Journaux automatiques (journaux que d'autres Services AWS, tels qu'Amazon Virtual Private Cloud (Amazon VPC) et Amazon Route 53, créent en votre nom)

Pour plus d'informations sur les journaux vendus, consultez la section Activation de la journalisation à partir de certains AWS services dans le guide de l'utilisateur Amazon CloudWatch Logs.

Les journaux personnalisés et vended génèrent des coûts en fonction du nombre de journaux qui sont collectés, conservés, et analysés. Séparément, les journaux vendus génèrent des coûts de livraison vers Amazon S3 et Firehose.

Le tableau suivant répertorie les noms des fonctionnalités CloudWatch Logs et les noms des sous-fonctionnalités pertinentes. La table comprend les chaînes pour `UsageType` et `Operation`, ce qui peut vous aider à analyser et à identifier les coûts liés aux journaux.

CloudWatch Fonctionnalité de journalisation	CloudWatch Sous-fonctionnalité Logs	**`UsageType`**	**`Operation`**	Objectif
Journaux personnalisés	Collecter (ingérer)	`DataProcessing-Bytes`	`PutLogEvents`	Télécharge un lot de journaux dans un flux

CloudWatch Fonctionnalité de journalisation	CloudWatch Sous-fonctionnalité Logs	**UsageType**	**Operation**	Objectif
				de journaux spécifique
	Store (archive))	`TimedStorage-ByteHrs`	`HourlyStorageMetering`	Stocke les logs par heure et les logs par octet dans CloudWatch Logs
	Analyser (requêtes Logs Insights)	`DataScanned-Bytes`	`StartQuery`	Données de journalisation analysées par les requêtes CloudWatch Logs Insights
Journaux vended	Livraison (CloudWatch journaux)	`VendedLog-Bytes`	`PutLogEvents`	Télécharge un lot de journaux dans un flux de journaux spécifique
	Mise à disposition (Amazon S3)	`S3-Egress-ComprBytes` `S3-Egress-Bytes`	`LogDelivery`	Envoie des journaux vendus (CloudWatchAmazon S3 ou Firehose)
	Livraison (Firehose)	`FH-Egress-Bytes`	`LogDelivery`	Envoie des journaux vendus (CloudWatchAmazon S3 ou Firehose)

Pour analyser les coûts, utilisez AWS Cost and Usage Report s avec Athena, afin d'identifier les journaux qui génèrent des coûts et de déterminer comment les coûts sont générés.

Exemple : requête Athena

Vous pouvez utiliser la requête suivante pour suivre quels journaux génèrent des coûts par ID de ressource.

```
SELECT
bill_payer_account_id as Payer,
line_item_usage_account_id as LinkedAccount,
line_item_resource_id AS ResourceID,
line_item_usage_type AS UsageType,
SUM(CAST(line_item_unblended_cost AS decimal(16,8))) AS TotalSpend,
SUM(CAST(line_item_usage_amount AS double)) AS UsageQuantity
FROM
costandusagereport
WHERE
product_product_name = 'AmazonCloudWatch'
AND year='2022'
AND month='4'
AND line_item_operation IN
 ('PutLogEvents','HourlyStorageMetering','StartQuery','LogDelivery')
AND line_item_line_item_type NOT IN
 ('Tax','Credit','Refund','EdpDiscount','Fee','RIFee')
GROUP BY
bill_payer_account_id,
line_item_usage_account_id,
line_item_usage_type,
line_item_resource_id,
line_item_operation
ORDER BY
TotalSpend DESC
```

Pour tirer le meilleur parti des coûts générés par les CloudWatch journaux, tenez compte des points suivants :

- Consignez uniquement les événements qui apportent de la valeur à votre entreprise. Cela vous permet de réduire les coûts d'ingestion.
- Modifiez vos paramètres de conservation des journaux afin de réduire les coûts de stockage. Pour plus d'informations, consultez la section Conservation des données du journal des modifications dans CloudWatch Logs du guide de l'utilisateur Amazon CloudWatch Logs.

- Exécutez des requêtes que CloudWatch Logs Insights enregistre automatiquement dans votre historique. Ainsi, vous réduisez les coûts d'analyse. Pour plus d'informations, consultez Afficher les requêtes en cours ou l'historique des requêtes dans le guide de l'utilisateur Amazon CloudWatch Logs.
- Utilisez l' CloudWatch agent pour collecter les journaux du système et des applications et les envoyer à CloudWatch. Ainsi, vous ne pouvez collecter que les événements de journaux qui répondent à vos critères. Pour plus d'informations, consultez Amazon CloudWatch Agent ajoute le Support pour les expressions de filtre de journal.

Pour réduire les coûts liés aux journaux vendus, considérez votre cas d'utilisation, puis déterminez si vos journaux doivent être envoyés à Amazon S3 CloudWatch ou à Amazon S3. Pour plus d'informations, consultez la section Logs envoyés à Amazon S3 dans le guide de l'utilisateur Amazon CloudWatch Logs.

> **Tip**
>
> Si vous souhaitez utiliser des filtres métriques, des filtres d'abonnement, CloudWatch Logs Insights et Contributor Insights, envoyez les journaux vendus à CloudWatch.
> Sinon, si vous travaillez avec des journaux de flux VPC et que vous les utilisez à des fins d'audit et de conformité, envoyez des journaux vended à Amazon S3.
> Pour plus d'informations sur le suivi des frais générés par la publication des journaux de flux VPC dans des compartiments S3, consultez la section Utilisation de balises AWS Cost and Usage Report s et de répartition des coûts pour comprendre l'ingestion des données des journaux de flux VPC dans Amazon S3.

Pour plus d'informations sur la manière de tirer le meilleur parti des coûts générés par CloudWatch les journaux, voir Quel groupe de journaux entraîne une augmentation soudaine de ma facture de CloudWatch journaux ? .

Utilisation des tableaux de CloudWatch bord Amazon

Les CloudWatch tableaux de bord Amazon sont des pages d'accueil personnalisables dans la CloudWatch console que vous pouvez utiliser pour surveiller vos ressources dans une seule vue, même celles qui sont réparties dans différentes régions. Vous pouvez utiliser CloudWatch des tableaux de bord pour créer des vues personnalisées des mesures et des alarmes relatives à vos AWS ressources.

Avec les tableaux de bord, vous pouvez créer les éléments suivants :

- Une vue unique pour les métriques et alertes sélectionnées afin de vous aider à évaluer l'état de vos ressources et applications d'une ou de plusieurs régions. Vous pouvez sélectionner la couleur utilisée pour chaque métrique sur chaque graphique afin de suivre facilement la même métrique sur plusieurs graphiques.
- Un manuel opérationnel qui fournit des conseils aux membres d'une équipe pendant des événements opérationnels sur la façon de répondre à des incidents spécifiques.
- Une vue globale des mesures critiques des ressources et applications qui peut être partagée par les membres de l'équipe pour un flux de communication plus rapide au cours des événements opérationnels.

Si vous avez plusieurs AWS comptes, vous pouvez configurer l'observabilité CloudWatch entre comptes, puis créer de riches tableaux de bord inter-comptes dans vos comptes de surveillance. Ces tableaux de bord peuvent inclure des graphiques de statistiques provenant de comptes sources et des widgets CloudWatch Logs Insights contenant des requêtes de groupes de journaux provenant de comptes sources. En outre, les alarmes que vous créez dans le compte de surveillance peuvent surveiller les métriques des comptes sources. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

Vous pouvez créer des tableaux de bord à partir de la console ou à l'aide de l'opération AWS CLI ou de `PutDashboard` l'API. Vous pouvez ajouter des tableaux de bord à une liste de favoris, dans laquelle vous pouvez accéder non seulement à vos tableaux de bord favoris, mais également aux tableaux de bord que vous avez récemment consultés. Pour plus d'informations, consultez Add a dashboard to your favorites list (Ajout d'un tableau de bord à votre liste de favoris).

Pour accéder aux CloudWatch tableaux de bord, vous avez besoin de l'un des éléments suivants :

- La politique `AdministratorAccess`

- La politique `CloudWatchFullAccess`
- Une politique personnalisée qui inclut une ou plusieurs de ces autorisations spécifiques :
  - `cloudwatch:GetDashboard` et `cloudwatch:ListDashboards` pour pouvoir afficher des tableaux de bord
  - `cloudwatch:PutDashboard` pour pouvoir créer ou modifier des tableaux de bord
  - `cloudwatch:DeleteDashboards` pour pouvoir supprimer des tableaux de bord

Table des matières

Création d'un CloudWatch tableau de bord

Pour commencer, créez un CloudWatch tableau de bord. Vous pouvez créer plusieurs tableaux de bord et ajouter des tableaux de bord à une liste de favoris. Vous n'êtes pas limité au nombre de tableaux de bord que peut contenir votre Compte AWS. Tous les tableaux de bord sont globaux. Ils ne sont pas spécifiques à une région.

La procédure suivante explique comment créer un tableau de bord à partir de la CloudWatch console. Vous pouvez utiliser l'opération d'API `PutDashboard` pour créer un tableau de bord depuis l'interface de ligne de commande. L'opération d'API contient une chaîne JSON qui définit le contenu de votre tableau de bord. Pour plus d'informations sur la création d'un tableau de bord avec

le fonctionnement de l'`PutDashboard`API, consultez PutDashboardle Amazon CloudWatch API Reference.

> **Tip**
>
> Si vous créez un tableau de bord avec l'opération d'API `PutDashboard`, vous pouvez utiliser la chaîne JSON d'un tableau de bord existant.

Pour créer un tableau de bord depuis la console

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), puis Create dashboard (Créer un tableau de bord).
3. Dans la boîte de dialogue Create new dashboard (Créer un tableau de bord), saisissez un nom pour le tableau de bord, puis cliquez sur Create dashboard (Créer un tableau de bord).

   Si vous utilisez le nom CloudWatch-Default ou CloudWatch -Default - ***`ResourceGroupName`***, le tableau de bord apparaît dans l'aperçu de la page d' CloudWatch accueil sous Tableau de bord par défaut. Pour plus d'informations, consultez Commencer à utiliser Amazon CloudWatch.
4. Dans la boîte de dialogue Add to this dashboard (Ajouter à ce tableau de bord), exécutez l'une des actions suivantes :
   - Pour ajouter un graphique au tableau de bord, choisissez Line (Ligne) ou Stacked area (Aires empilées), puis choisissez Configure (Configurer). Dans la boîte de dialogue Add metric graph (Ajouter un graphique métrique), sélectionnez la ou les métriques à représenter graphiquement, puis choisissez Create widget (Créer un widget). Si une métrique ne s'affiche pas dans la boîte de dialogue parce qu'elle n'a pas publié de données en plus de 14 jours, vous pouvez l'ajouter manuellement. Pour plus d'informations, consultez Représentez les métriques manuellement sur un CloudWatch tableau de bord.
   - Pour ajouter un nombre affichant une métrique au tableau de bord, choisissez Number (Nombre), puis choisissez Configure (Configurer). Dans la boîte de dialogue Add metric graph (Ajouter un graphique métrique), sélectionnez la ou les métriques à représenter graphiquement, puis choisissez Create widget (Créer un widget).
   - Pour ajouter un bloc de texte au tableau de bord, choisissez Text (Texte), puis choisissez Configure (Configurer). Dans la boîte de dialogue New text widget (Nouveau widget de texte), pour Markdown, formatez votre texte avec Markdown, puis choisissez Create widget (Créer un widget).

5. (Facultatif) Choisissez Add widget (Ajouter un widget), puis répétez l'étape 4 pour ajouter un autre widget au tableau de bord. Vous pouvez répéter cette étape plusieurs fois.

   Pour chaque graphique du tableau de bord, une icône d'information se trouve en haut à droite. Cliquez sur cette icône pour afficher les descriptions des métriques dans le graphique.

6. Choisissez Save dashboard (Enregistrer le tableau de bord).

CloudWatch tableau de bord d'observabilité entre comptes

Si vous avez plusieurs AWS comptes, vous pouvez configurer l'observabilité CloudWatch entre comptes, puis créer de riches tableaux de bord inter-comptes dans vos comptes de surveillance. Vous pouvez rechercher, visualiser et analyser de manière transparente vos métriques, journaux et traces sans limites de compte.

Pour plus d'informations sur la configuration de l'observabilité CloudWatch entre comptes, consultez. CloudWatch observabilité entre comptes

Grâce à CloudWatch l'observabilité entre comptes, vous pouvez effectuer les opérations suivantes dans le tableau de bord d'un compte de surveillance :

- Rechercher, afficher et créer des graphiques de métriques qui résident dans des comptes sources. Un seul graphique peut inclure des métriques provenant de plusieurs comptes.
- Créer des alarmes dans le compte de surveillance qui surveillent les métriques dans les comptes sources.
- Affichez les événements du journal des groupes de journaux situés dans les comptes source et exécutez CloudWatch des requêtes Logs Insights sur les groupes de journaux des comptes source. Une seule requête CloudWatch Logs Insights dans un compte de surveillance peut interroger simultanément plusieurs groupes de journaux dans plusieurs comptes sources.
- Affichez les nœuds des comptes sources dans une carte de suivi dans X-Ray. Vous pouvez ensuite filtrer le mappage sur des comptes sources spécifiques.

Lorsque vous êtes connecté à un compte de surveillance, un badge bleu de compte de surveillance apparaît en haut à droite de chaque page qui prend en charge la fonctionnalité d'observabilité CloudWatch entre comptes.

Tableaux de bord entre régions et comptes

Vous pouvez créer des tableaux de bord inter-comptes interrégionaux, qui résument les CloudWatch données de plusieurs AWS comptes et de plusieurs régions dans un seul tableau de bord. Ce tableau de bord de haut niveau vous fournit une vue de l'ensemble de votre application, ainsi que des tableaux de bord plus spécifiques sans devoir vous connecter et vous déconnecter des comptes ou changer de région.

Vous pouvez créer des tableaux de bord inter-comptes interrégionaux dans et par programmation. AWS Management Console

Prérequis

Avant de pouvoir créer un tableau de bord entre régions et comptes, vous devez activer au moins un compte de partage et au moins un compte de surveillance. En outre, pour pouvoir utiliser la CloudWatch console pour créer un tableau de bord multi-comptes, vous devez activer la console pour la fonctionnalité multi-comptes. Pour plus d'informations, consultez Console multicompte et multirégion CloudWatch .

Création et utilisation d'un tableau de bord entre régions et comptes à l'aide de la AWS Management Console

Vous pouvez utiliser le AWS Management Console pour créer un tableau de bord inter-comptes interrégional.

Pour créer un tableau de bord entre régions et comptes

1. Connectez-vous au compte de surveillance.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation, choisissez Dashboards (Tableaux de bord).
4. Choisissez un tableau de bord ou créez-en un nouveau.
5. En haut de l'écran, vous pouvez basculer entre les comptes et les régions. Lorsque vous créez votre tableau de bord, vous pouvez inclure des widgets provenant de plusieurs comptes et régions. Les widgets incluent des graphiques, des alarmes et CloudWatch des widgets Logs Insights.

Création d'un graphique avec des métriques de différents comptes et régions

1. Connectez-vous au compte de surveillance.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques).
4. Sélectionnez le compte et la région à partir desquels vous voulez ajouter des métriques. Vous pouvez sélectionner votre compte et votre région dans les menus déroulants Account (Compte) et Region (Région) situés en haut à droite de l'écran.
5. Ajoutez les métriques souhaitées au graphique. Pour de plus amples informations, consultez Graphique des métriques.
6. Répétez les étapes 4 à 5 pour ajouter des métriques provenant d'autres comptes et régions.
7. (Facultatif) Choisissez l'onglet Graphed metrics (Métriques sous forme de graphique) et ajoutez une fonction mathématique de métrique qui utilise celles que vous avez choisies. Pour de plus amples informations, consultez Utilisation des mathématiques appliquées aux métriques.

   Vous pouvez également configurer un graphique unique pour inclure plusieurs fonctions `SEARCH`. Chaque recherche peut faire référence à une région ou à un compte différent.
8. Lorsque vous avez terminé avec le graphique, choisissez Actions, Add to dashboard (Ajouter au tableau de bord).

   Sélectionnez votre tableau de bord entre comptes, puis choisissez Add to dashboard (Ajouter au tableau de bord).

Ajout d'une alerte à partir d'un autre compte à votre tableau de bord entre comptes

1. Connectez-vous au compte de surveillance.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. En haut de la page, choisissez le compte où se trouve l'alerte.
4. Dans le panneau de navigation, cliquez sur Alarms (alertes).
5. Activez la case à cocher en regard de l'alerte que vous souhaitez ajouter, puis choisissez Add to dashboard (Ajouter au tableau de bord).
6. Sélectionnez le tableau de bord entre comptes auquel vous souhaitez l'ajouter, puis choisissez Add to dashboard (Ajouter au tableau de bord).

Création par programmation d'un tableau de bord entre régions et comptes

Vous pouvez utiliser les AWS API et les SDK pour créer des tableaux de bord par programmation. Pour plus d'informations, consultez PutDashboard.

Pour activer les tableaux de bord entre régions et comptes, nous avons ajouté de nouveaux paramètres à la structure du corps du tableau de bord, comme le montre le tableau et les exemples suivants. Pour plus d'informations sur la structure globale du corps du tableau de bord, consultez Structure et syntaxe du corps d'un tableau de bord.

Paramètre	Utiliser	Portée	Par défaut
`accountId`	Spécifie l'ID du compte où se trouve le widget ou la métrique.	Widget ou métrique	Compte actuellement connecté
`region`	Spécifie la région de la métrique.	Widget ou métrique	Région actuelle sélectionnée dans la console

Les exemples suivants illustrent la source JSON pour les widgets dans un tableau de bord entre régions et comptes.

Cet exemple montre comment définir le champ `accountId` pour l'ID du compte de partage au niveau du widget. Cela spécifie que toutes les métriques de ce widget proviendront de ce compte de partage et de cette région.

```
{
    "widgets": [
        {
            ...
            "properties": {
              "metrics": [
                     …..
                     ],
                    "accountId": "111122223333",
                    "region": "us-east-1"
            }
        }
  ]
```

```
}
```

Cet exemple montre comment définir le champ `accountId` différemment au niveau de chaque métrique. Dans cet exemple, les différentes métriques de cette expression mathématique de mesure proviennent de différents comptes de partage et de différentes régions.

```
{
"widgets": [
    {
        ...
        "properties": {
            "metrics": [
                [ { "expression": "SUM(METRICS())", "label": "[avg: ${AVG}]
 Expression1", "id": "e1", "stat": "Sum" } ],
                [ "AWS/EC2", "CPUUtilization", { "id": "m2", "accountId":
 "5555666677778888", "region": "us-east-1", "label": "[avg: ${AVG}] ApplicationALabel
 " } ],
                [ ".", ".", { "id": "m1", "accountId": "9999000011112222", "region":
 "eu-west-1", "label": "[avg: ${AVG}] ApplicationBLabel" } ]
            ],
            "view": "timeSeries",
            "region":"us-east-1", ---> home region of the metric. Not present in above
 example
            "stacked": false,
            "stat": "Sum",
            "period": 300,
            "title": "Cross account example"
        }
    }
]
}
```

Cet exemple montre un widget d'alerte.

```
{
    "type": "metric",
    "x": 6,
    "y": 0,
    "width": 6,
    "height": 6,
    "properties": {
        "accountID": "111122223333",
```

```
        "title": "over50",
        "annotations": {
            "alarms": [
                "arn:aws:cloudwatch:us-east-1:379642911888:alarm:over50"
            ]
        },
        "view": "timeSeries",
        "stacked": false
    }
}
```

Cet exemple concerne un widget CloudWatch Logs Insights.

```
{
    "type": "log",
    "x": 0,
    "y": 6,
    "width": 24,
    "height": 6,
    "properties": {
        "query": "SOURCE 'route53test' | fields @timestamp, @message\n| sort @timestamp
 desc\n| limit 20",
        "accountId": "111122223333",
        "region": "us-east-1",
        "stacked": false,
        "view": "table"
    }
}
```

Une autre méthode pour créer des tableaux de bord par programmation consiste à en créer un dans le AWS Management Console, puis à copier la source JSON de ce tableau de bord. Pour ce faire, chargez le tableau de bord et choisissez Actions, View-edit source (Afficher/Modifier la source). Vous pouvez ensuite copier ce tableau de bord JSON pour l'utiliser comme modèle pour créer des tableaux de bord similaires.

Créer des tableaux de bord flexibles avec des variables de tableau de bord

Utilisez des variables de tableau de bord pour créer des tableaux de bord flexibles capables d'afficher rapidement différents contenus dans plusieurs widgets en fonction de la valeur d'une zone de saisie

dans le tableau de bord. Par exemple, vous pouvez créer un tableau de bord capable de basculer rapidement entre différentes fonctions Lambda ou différents ID d'instance Amazon EC2, ou un tableau de bord qui peut passer d'une région à l'autre. AWS

Après avoir créé un tableau de bord qui utilise une variable, vous pouvez copier ce même modèle de variable dans d'autres tableaux de bord existants.

L'utilisation de variables de tableau de bord améliore le déroulement des opérations pour les utilisateurs de vos tableaux de bord. Elle peut également réduire vos coûts, car vous utilisez les variables de tableau de bord dans un seul tableau de bord au lieu de créer plusieurs tableaux de bord similaires.

> **Note**
>
> Si vous partagez un tableau de bord contenant des variables de tableau de bord, les personnes avec lesquelles vous le partagez ne pourront pas changer les valeurs des variables.

Types de variables de tableau de bord

Les variables de tableau de bord peuvent être des variables de propriété ou des variables de modèle.

- Les variables de propriété modifient toutes les instances d'une propriété dans tous les widgets du tableau de bord. Il peut s'agir de n'importe quelle propriété JSON de la source JSON d'un tableau de bord, comme `region`. Il peut également s'agir du nom d'une dimension pour une métrique, comme `InstanceID` ou `FunctionName`.

  Consultez Tutoriel : Créer un tableau de bord Lambda avec le nom de la fonction comme variable pour voir un didacticiel avec une variable de propriété.

  Pour de plus amples informations sur la source JSON des tableaux de bord, consultez Dashboard Body Structure and Syntax. Dans la CloudWatch console, vous pouvez voir la source JSON de n'importe quel tableau de bord personnalisé en choisissant Actions, Afficher/modifier la source.
- Les variables de modèle utilisent un modèle d'expression régulière pour modifier l'ensemble d'une propriété JSON ou seulement une partie de celle-ci.

  Consultez Didacticiel : Créer un tableau de bord qui utilise un modèle d'expression régulière pour passer d'une région à l'autre pour voir un didacticiel avec une variable de modèle.

Les variables de propriété s'appliquent à la plupart des cas d'utilisation et sont moins complexes à configurer.

Rubriques

- Tutoriel : Créer un tableau de bord Lambda avec le nom de la fonction comme variable
- Didacticiel : Créer un tableau de bord qui utilise un modèle d'expression régulière pour passer d'une région à l'autre
- Copier une variable dans un autre tableau de bord

Tutoriel : Créer un tableau de bord Lambda avec le nom de la fonction comme variable

Les étapes de cette procédure montrent comment créer un tableau de bord flexible qui affiche divers graphiques de métriques à l'aide d'une variable de propriété. Cela inclut un menu déroulant dans le tableau de bord que vous pouvez utiliser pour changer les métriques de tous les graphiques entre les différentes fonctions Lambda.

D'autres exemples d'utilisation de ce type de tableau de bord incluent l'utilisation de `InstanceId` en tant que variable pour créer un tableau de bord de métriques avec une liste déroulante pour les identifiants d'instance. Vous pouvez également créer un tableau de bord qui utilise `region` comme variable pour afficher le même ensemble de métriques provenant de différentes régions.

Pour utiliser une variable de propriété de tableau de bord afin de créer un tableau de bord Lambda flexible

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Tableaux de bord, Créer un tableau de bord.
3. Saisissez un nom pour le tableau de bord, puis choisissez Créer un tableau de bord.
4. Ajoutez des widgets au tableau de bord qui affichent les métriques d'une fonction Lambda. Lorsque vous créez ces widgets, spécifiez Lambda, Par nom de fonction pour les métriques de widget. Pour la fonction, spécifiez l'une des fonctions Lambda que vous souhaitez inclure dans ce tableau de bord.

   Pour en savoir plus sur l'ajout de widgets à un tableau de bord, consultez Création et utilisation de widgets sur les CloudWatch tableaux de bord.

5. Après avoir ajouté les widgets, lorsque vous affichez le tableau de bord, choisissez Actions, Variables, Créer une variable.
6. Choisissez Variable de propriété.
7. Pour Propriété modifiée par la variable, sélectionnez FunctionName.
8. Pour Type d'entrée, pour ce cas d'utilisation, nous vous recommandons de choisir Menu de sélection (liste déroulante). Cela crée un menu déroulant dans le tableau de bord dans lequel vous pouvez sélectionner le nom de la fonction Lambda pour laquelle afficher les métriques.

   S'il s'agissait d'un tableau de bord basculant entre deux ou trois valeurs uniquement pour une variable, Bouton radio serait un bon choix.

   Si vous préférez saisir ou coller des valeurs pour la variable, choisissez Entrée de texte. Cette option n'inclut pas de liste déroulante ni de boutons radio.
9. Lorsque vous choisissez Menu de sélection (liste déroulante), vous devez ensuite choisir de remplir le menu en saisissant des valeurs ou en utilisant une recherche de métriques. Pour ce cas d'utilisation, supposons que vous disposez d'un grand nombre de fonctions Lambda et que vous ne souhaitez pas toutes les saisir manuellement. Choisissez Utiliser les résultats d'une recherche de métriques, puis procédez comme suit :

   a. Choisissez Requêtes prédéfinies, Lambda, Erreurs.

      (Le choix des erreurs n'ajoute pas la métrique des erreurs au tableau de bord. Cependant, il remplit rapidement la zone de sélection des FunctionNamevariables.)

   b. Choisissez Par nom de fonction, puis sélectionnez Rechercher.

      Sous le bouton Rechercher, vous verrez ensuite la case FunctionNamesélectionnée. Un message indiquant le nombre de valeurs de FunctionNamedimension trouvées pour remplir la zone de saisie s'affiche également.
10. (Facultatif) Pour ajouter d'autres paramètres, choisissez Paramètres secondaires et effectuez une ou plusieurs des opérations suivantes :

    - Pour personnaliser le nom de votre variable, saisissez-le dans Nom de variable personnalisé.
    - Pour personnaliser l'étiquette de la zone de saisie de variable, saisissez-la dans Étiquette d'entrée.
    - Pour définir la valeur par défaut de cette variable lorsque le tableau de bord est ouvert pour la première fois, saisissez la valeur par défaut dans Valeur par défaut.
11. Choisissez Ajouter une variable.

Une boîte de sélection FunctionNamedéroulante apparaît en haut du tableau de bord. Vous pouvez sélectionner une fonction Lambda dans cette zone pour que tous les widgets utilisant la variable affichent des informations sur la fonction sélectionnée.

Plus tard, si vous ajoutez d'autres widgets au tableau de bord qui surveillent les métriques Lambda avec la FunctionNamedimension, ils utiliseront automatiquement la variable.

Didacticiel : Créer un tableau de bord qui utilise un modèle d'expression régulière pour passer d'une région à l'autre

Les étapes de cette procédure montrent comment créer un tableau de bord flexible pouvant passer d'une région à l'autre. Ce didacticiel utilise une variable de modèle d'expression régulière au lieu d'une variable de propriété. Consultez Tutoriel : Créer un tableau de bord Lambda avec le nom de la fonction comme variable pour voir un didacticiel avec une variable de propriété.

Dans de nombreux cas d'utilisation, vous pouvez créer un tableau de bord qui passe d'une région à l'autre à l'aide d'une variable de propriété. Mais si les widgets s'appuient sur des Amazon Resource Names (ARN) qui incluent des noms de région, vous devez utiliser une variable de modèle pour modifier les noms de région dans les ARN.

Pour utiliser une variable de modèle de tableau de bord afin de créer un tableau de bord multi-régions flexible

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Tableaux de bord, Créer un tableau de bord.
3. Saisissez un nom pour le tableau de bord, puis choisissez Créer un tableau de bord.
4. Ajoutez des widgets au tableau de bord. Lorsque vous ajoutez les widgets qui afficheront les données spécifiques à une région, évitez de spécifier des dimensions avec des valeurs qui n'apparaissent que dans une seule région. Par exemple, pour les métriques Amazon EC2, spécifiez des métriques agrégées plutôt que des métriques utilisant InstanceId comme dimension.

   Pour en savoir plus sur l'ajout de widgets à un tableau de bord, consultez Création et utilisation de widgets sur les CloudWatch tableaux de bord.
5. Après avoir ajouté les widgets, lorsque vous affichez le tableau de bord, choisissez Actions, Variables, Créer une variable.

6. Choisissez Variable de modèle.
7. Pour Propriété modifiée par la variable, saisissez le nom de la région actuelle du tableau de bord, par exemple **us-east-2**.

   Vous avez saisi la bonne région si l'étiquette située sous cette zone affiche les widgets qui seront affectés par la variable.
8. Pour Type d'entrée, pour ce cas d'utilisation, sélectionnez Bouton radio.
9. Pour Définir le mode de remplissage des entrées, choisissez Créer une liste des valeurs personnalisées.
10. Pour Créer vos valeurs personnalisées, saisissez les régions entre lesquelles vous souhaitez basculer, avec une région par ligne. Après chaque région, saisissez une virgule puis l'étiquette à afficher pour le bouton radio en question. Par exemple :

    us-east-1, N. Virginia

    us-east-2, Ohio

    eu-west-3, Paris

    Au fur et à mesure que vous saisissez les valeurs personnalisées, le volet Aperçu s'actualise pour montrer à quoi ressembleront les boutons radio.
11. (Facultatif) Pour ajouter d'autres paramètres, choisissez Paramètres secondaires et effectuez une ou plusieurs des opérations suivantes :

    - Pour personnaliser le nom de votre variable, saisissez-le dans Nom de variable personnalisé.
    - Pour personnaliser l'étiquette de la zone de saisie de variable, saisissez-la dans Étiquette d'entrée. Dans le cadre de ce didacticiel, entrez **Region:**.

      Si vous saisissez une valeur ici, le volet Aperçu s'actualise pour montrer à quoi ressembleront les boutons radio.
    - Pour définir la valeur par défaut de cette variable lorsque le tableau de bord est ouvert pour la première fois, saisissez la valeur par défaut dans Valeur par défaut.
12. Choisissez Ajouter une variable.

    Le tableau de bord apparaît, avec une étiquette Région : à côté des boutons radio pour les régions près du haut. Lorsque vous passez d'une région à l'autre, tous les widgets qui utilisent la variable affichent des informations sur la région sélectionnée.

Copier une variable dans un autre tableau de bord

Après avoir créé un tableau de bord avec des variables utiles, vous pouvez copier ces variables dans d'autres tableaux de bord existants. Pour plus d'informations sur les variables de tableau de bord, consultez Créer des tableaux de bord flexibles avec des variables de tableau de bord.

Copier une variable de tableau de bord dans un autre tableau de bord

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Tableaux de bord, puis choisissez le nom du tableau de bord qui contient la variable que vous souhaitez copier. Saisissez une chaîne pour rechercher les tableaux de bord dont les noms correspondent, si nécessaire.
3. Choisissez Actions, Variables, Gérer les variables.
4. Cliquez sur le bouton radio à côté de la variable que vous souhaitez copier, puis choisissez Copier vers un autre tableau de bord.
5. Choisissez la zone de sélection et commencez à saisir le nom du tableau de bord dans lequel vous souhaitez copier la variable.
6. Sélectionnez le nom du tableau de bord et choisissez Copier une variable.

Création et utilisation de widgets sur les CloudWatch tableaux de bord

Utilisez les rubriques de cette section pour créer et utiliser des graphiques, des alertes et des widgets de texte au sein de vos tableaux de bord.

Table des matières

- Ajouter ou supprimer un widget de texte dans un CloudWatch tableau de bord
- Ajouter ou supprimer un widget d'alarme dans un CloudWatch tableau de bord
- Ajouter ou supprimer un widget de tableau de données dans un CloudWatch tableau de bord
- Lier et dissocier des graphiques sur un tableau de bord CloudWatch

Ajouter ou supprimer un graphique dans un CloudWatch tableau de bord

Vous pouvez ajouter des graphiques contenant une ou plusieurs mesures à votre CloudWatch tableau de bord. Les types de graphiques que vous pouvez ajouter à votre tableau de bord incluent : Line (Ligne), Stacked area (Aires empilées), Number (Numéro), Gauge (Jauge), Bar (À barres) et Pie (À secteurs). Vous pouvez supprimer des graphiques de votre tableau de bord lorsque vous n'en avez plus besoin. Les procédures de cette section expliquent comment ajouter et supprimer des graphiques sur votre tableau de bord. Pour plus d'informations sur la façon de modifier un graphique sur votre tableau de bord, voir Modifier un graphique sur un CloudWatch tableau de bord.

Pour ajouter un graphique à 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Cliquez sur le symbole +, puis sélectionnez le type de graphique que vous souhaitez ajouter à votre tableau de bord, puis choisissez Suivant.

   - Si vous sélectionnez Line (Ligne), Stacked area (Aires empilées), Bar (À barres), ou Pie (À secteurs), choisissez Metrics (Métriques).
4. Dans l'onglet Parcourir, recherchez ou parcourez les métriques à représenter graphiquement, puis sélectionnez celles que vous souhaitez.
5. (Facultatif) Pour modifier l'intervalle de temps de votre graphique, sélectionnez l'un des intervalles de temps prédéfinis dans la partie supérieure de l'écran. Les plages de temps durent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j, ou 1 sem).

   Pour définir votre propre plage de temps, choisissez Custom (Personnalisée).

   - (Facultatif) Pour que ce widget continue d'utiliser l'intervalle de temps que vous sélectionnez, même si l'intervalle de temps pour le reste du tableau de bord est modifié ultérieurement, choisissez Conserver l'intervalle de temps.

6. (Facultatif) Pour modifier le type de widget de votre graphique, utilisez le menu déroulant situé à côté des intervalles de temps prédéfinis.
7. (Facultatif) Dans Graphed metrics (Métriques représentées graphiquement), vous pouvez ajouter une étiquette dynamique à votre métrique et modifier l'étiquette, la couleur d'étiquette, les statistiques et la période de votre métrique. Vous pouvez également définir la position des étiquettes sur l'axe Y de gauche à droite.

   a. Pour ajouter une étiquette dynamique, choisissez Graphed metrics (Métriques représentées graphiquement), puis choisissez Add dynamic labels (Ajouter des étiquettes dynamiques). Les étiquettes dynamiques affichent une statistique concernant votre métrique dans la légende du graphique. Les étiquettes dynamiques se mettent à jour automatiquement lorsque votre tableau de bord ou votre graphique s'actualise. Par défaut, les valeurs dynamiques que vous ajoutez aux étiquettes s'affichent au début de vos étiquettes. Pour plus d'informations, consultez Utiliser des étiquettes dynamiques.

   b. Pour modifier la couleur d'une métrique, choisissez le carré de couleur situé à côté de la métrique.

   c. Pour modifier la statistique, sélectionnez la liste déroulante sous Statistique (Statistiques), puis choisissez une nouvelle valeur. Pour plus d'informations, consultez Statistics (Statistiques).

   d. Pour modifier la période, sélectionnez la liste déroulante dans la colonne Period (Période), puis choisissez une nouvelle valeur.
8. Si vous créez un widget de jauge, vous devez choisir l'onglet Options et spécifier les valeurs Min et Max à utiliser pour les deux extrémités de la jauge.
9. (Facultatif) Pour personnaliser l'axe Y, choisissez Options. Vous pouvez ajouter une étiquette personnalisée sous Left Y-axis (Axe Y de gauche) dans le champ d'étiquette. Si votre graphique affiche des valeurs sur le côté droit de l'axe Y, vous pouvez personnaliser cette étiquette également. Vous pouvez également définir des limites minimum et maximum pour les valeurs de l'axe Y, de façon à ce que votre graphique affiche uniquement la plage de valeurs que vous spécifiez.
10. (Facultatif) Pour ajouter ou modifier des annotations horizontales sur des graphiques linéaires ou à aires empilées, ou pour ajouter des seuils aux widgets de jauge, choisissez Options :

    a. Pour ajouter une annotation horizontale ou un seuil, choisissez Ajouter des annotations horizontales ou Ajouter un seuil.

    b. Pour Étiquette, saisissez une étiquette pour l'annotation, puis cliquez sur l'icône de coche.

c. Pour Value (Valeur), choisissez l'icône en forme de crayon et papier qui est à côté de la valeur actuelle, puis saisissez votre nouvelle valeur. Après avoir saisi votre valeur, choisissez l'icône de coche.

d. Pour Fill (Remplir), sélectionnez la liste déroulante, puis spécifiez comment votre annotation utilisera l'ombrage. Vous pouvez choisir None (Aucun), Above (Au-dessus), Between (Entre) ou Below (En-dessous). Pour modifier la couleur de remplissage, choisissez le carré de couleur en regard de l'annotation.

e. Pour Axis (Axe), indiquez si votre annotation apparaît sur le côté droit ou gauche de l'axe des Y.

f. Pour masquer une annotation, décochez la case en regard de l'annotation que vous souhaitez masquer.

g. Pour supprimer une annotation, choisissez X sous Actions.

> **Note**
>
> Vous pouvez répéter ces étapes pour ajouter plusieurs annotations horizontales ou seuils au même graphique ou à la même jauge.

11. (Facultatif) Pour ajouter ou modifier des annotations verticales, choisissez Options :

    a. Pour ajouter une annotation verticale, choisissez Add vertical annotation (Ajouter une annotation verticale).

    b. Pour Label (Étiquette), choisissez l'icône en forme de crayon et papier qui est à côté de l'annotation actuelle, puis saisissez votre nouvelle annotation. Si vous souhaitez afficher seulement la date et l'heure, ne renseignez pas le champ Label (Étiquette).

    c. Pour Date, choisissez la date et l'heure actuelles, puis saisissez la nouvelle date et la nouvelle heure.

    d. Pour Fill (Remplir), sélectionnez la liste déroulante, puis spécifiez comment votre annotation utilisera l'ombrage. Vous pouvez choisir None (Aucun), Above (Au-dessus), Between (Entre) ou Below (En-dessous). Pour modifier la couleur de remplissage, sélectionnez le carré de couleur en regard de l'annotation.

    e. Pour masquer une annotation, décochez la case en regard de l'annotation que vous souhaitez masquer.

    f. Pour supprimer une annotation, choisissez X sous Actions.

> **Note**
>
> Répétez ces étapes pour ajouter plusieurs annotations verticales au même graphique.

12. Choisissez Create widget (Créer un widget).
13. Choisissez Save dashboard (Enregistrer le tableau de bord).

Pour supprimer un graphique d'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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Dans le coin supérieur droit du graphique que vous souhaitez supprimer, choisissez Widget actions (Actions du widget), puis choisissez Delete (Supprimer).
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Représentez les métriques manuellement sur un CloudWatch tableau de bord

Si aucune donnée n'a été publiée au cours des 14 derniers jours, vous ne la trouvez pas lorsque vous recherchez des statistiques à ajouter à un graphique sur un CloudWatch tableau de bord. Utilisez les étapes suivantes pour ajouter manuellement une métrique à un graphique existant.

Pour ajouter à un graphique une métrique que vous ne pouvez pas rechercher

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), puis choisissez un tableau de bord.
3. Le tableau de bord doit déjà contenir un graphique où vous voulez ajouter la métrique. S'il n'existe pas déjà, créez le graphique et ajoutez-lui une métrique. Pour de plus amples informations, consultez Ajouter ou supprimer un graphique dans un CloudWatch tableau de bord .
4. Choisissez Actions, View/Edit Attributes (Afficher/Modifier la source).

Un bloc JSON s'affiche. Le bloc définit les widgets du tableau de bord, ainsi que leur contenu. Voici un exemple d'une partie du bloc, qui définit un graphique.

```
{
            "type": "metric",
            "x": 0,
            "y": 0,
            "width": 6,
            "height": 3,
            "properties": {
                "view": "singleValue",
                "metrics": [
                    [ "AWS/EBS", "VolumeReadOps", "VolumeId",
 "vol-1234567890abcdef0" ]
                ],
                "region": "us-west-1"
            }
        },
```

Dans cet exemple, la section suivante définit la métrique affichée sur le graphique.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ]
```

5. Ajoutez une virgule après le crochet de fin, s'il n'y en a pas déjà une, puis ajoutez une section similaire entre crochets après la virgule. Dans cette nouvelle section, spécifiez l'espace de noms, le nom de la métrique et toute dimension nécessaire de la métrique que vous ajoutez au graphique. Voici un exemple.

```
[ "AWS/EBS", "VolumeReadOps", "VolumeId", "vol-1234567890abcdef0" ],
[ "MyNamespace", "MyMetricName", "DimensionName", "DimensionValue" ]
```

Pour plus d'informations sur la mise en forme des métriques dans JSON, consultez la section relative aux propriétés d'un objet widget de type métrique.

6. Choisissez Mettre à jour.

Utilisation de graphiques existants

Suivez les procédures décrites dans ces sections pour éditer et modifier vos widgets graphiques de tableau de bord existants.

Rubriques

- Modifier un graphique sur un CloudWatch tableau de bord
- Déplacer ou redimensionner un graphique sur un tableau de bord CloudWatch
- Renommer un graphique sur un tableau de bord CloudWatch

Modifier un graphique sur un CloudWatch tableau de bord

Vous pouvez modifier les graphiques que vous ajoutez à votre CloudWatch tableau de bord. Vous pouvez modifier le titre, la statistique ou la période d'un graphique. Vous pouvez ajouter, mettre à jour et supprimer des métriques de vos graphiques. Si votre graphique contient plusieurs métriques, vous pouvez réduire l'encombrement en masquant les métriques que vous n'utilisez pas. Les procédures de cette section expliquent comment modifier un graphique sur votre tableau de bord. Pour plus d'informations sur la création d'un graphique, voir Ajouter ou supprimer un graphique d'un CloudWatch tableau de bord.

New interface

Pour modifier un graphique de tableau de bord

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), puis choisissez un tableau de bord.
3. Dans le coin supérieur droit du graphique que vous souhaitez modifier, choisissez Widget actions (Actions du widget), puis choisissez Edit (Modifier).
4. Pour modifier le titre du graphique, choisissez l'icône en forme de crayon et papier en regard du titre actuel. Saisissez le nouveau titre, puis choisissez Apply (Appliquer).
5. (Facultatif) Pour modifier l'intervalle de temps de votre graphique, sélectionnez l'un des intervalles de temps prédéfinis dans la partie supérieure du graphique. Les plages de temps durent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j, ou 1 sem).

   Pour définir votre propre plage de temps, choisissez Custom (Personnalisée).

- (Facultatif) Pour que ce widget continue d'utiliser l'intervalle de temps que vous sélectionnez, même si l'intervalle de temps pour le reste du tableau de bord est modifié ultérieurement, choisissez Conserver l'intervalle de temps.

6. Pour modifier le type de widget de votre graphique, utilisez le menu déroulant situé à côté des intervalles de temps prédéfinis.
7. Dans Graphed metrics (Métriques représentées graphiquement), vous pouvez ajouter une étiquette dynamique à votre métrique et modifier l'étiquette, la couleur d'étiquette, les statistiques et la période de votre métrique. Vous pouvez également définir la position des étiquettes sur l'axe Y de gauche à droite.

   a. Pour ajouter une étiquette dynamique à une métrique, choisissez Dynamic labels (Étiquettes dynamiques). Les étiquettes dynamiques affichent une statistique concernant la métrique dans la légende du graphique. Les étiquettes dynamiques se mettent à jour automatiquement lorsque votre tableau de bord ou votre graphique s'actualise. Par défaut, les valeurs dynamiques que vous ajoutez aux étiquettes s'affichent au début des étiquettes. Pour plus d'informations, consultez Utiliser des étiquettes dynamiques.

   b. Pour modifier la couleur d'une métrique, choisissez le carré de couleur situé à côté de la métrique.

   c. Pour modifier les statistiques, choisissez la valeur de statistique dans la colonne Statistic (Statistique), puis choisissez une nouvelle valeur. Pour plus d'informations, consultez Statistiques.

   d. Pour modifier la période, choisissez la valeur de période dans la colonne Period (Période), puis choisissez une nouvelle valeur.

8. Pour ajouter ou modifier des annotations horizontales, choisissez Options :

   a. Pour ajouter une annotation horizontale, choisissez Add horizontal annotation (Ajouter une annotation horizontale).

   b. Pour Label (Étiquette), choisissez l'icône en forme de crayon et papier à côté de l'annotation actuelle. Saisissez ensuite votre nouvelle annotation. Après avoir saisi votre annotation, choisissez l'icône de coche.

   c. Pour Value (Valeur), choisissez l'icône en forme de crayon et papier en regard de la valeur de métrique actuelle. Saisissez ensuite votre nouvelle valeur de métrique. Après avoir saisi votre valeur, sélectionnez l'icône de coche.

   d. Pour Fill (Remplir), choisissez la liste déroulante sous la colonne, puis spécifiez comment votre annotation utilisera l'ombrage. Vous pouvez choisir None (Aucun), Above (Au-

dessus), Between (Entre) ou Below (En-dessous). Si vous choisissez Between (Entre), un autre nouveau champ d'étiquette et de valeur apparaît.

> **Tip**
>
> Vous pouvez modifier la couleur de remplissage en choisissant le carré de couleur en regard de l'annotation.

e. Pour Axis (Axe), indiquez si votre annotation apparaît sur le côté droit ou gauche de l'axe des Y.

f. Pour masquer une annotation, désélectionnez la case à cocher en regard de l'annotation que vous souhaitez masquer sur le graphique.

g. Pour supprimer une annotation, choisissez X sous la colonne Actions.

> **Note**
>
> Répétez ces étapes pour ajouter plusieurs annotations horizontales au même graphique.

9. Pour ajouter ou modifier des annotations verticales, choisissez Options :

   a. Pour ajouter une annotation verticale, choisissez Add vertical annotation (Ajouter une annotation verticale).

   b. Pour Label (Étiquette), choisissez l'icône en forme de crayon et papier à côté de l'annotation actuelle. Saisissez ensuite votre nouvelle annotation. Après avoir saisi votre annotation, choisissez l'icône de coche.

   > **Tip**
   >
   > Pour afficher seulement la date et l'heure, ne renseignez pas le champ Label (Étiquette).

   c. Pour Date, choisissez la date et l'heure actuelles. Saisissez ensuite la nouvelle date et l'heure.

   d. Pour Fill (Remplir), choisissez la liste déroulante sous la colonne, puis spécifiez comment votre annotation utilisera l'ombrage. Vous pouvez choisir None (Aucun), Above (Au-

dessus), Between (Entre) ou Below (En-dessous). Si vous choisissez Between (Entre), un nouveau champ d'étiquette et de valeur apparaît.

> **Tip**
>
> Vous pouvez modifier la couleur de remplissage en choisissant le carré de couleur en regard de l'annotation.

> **Note**
>
> Répétez ces étapes pour ajouter plusieurs annotations verticales au même graphique.

e. Pour masquer une annotation, désélectionnez la case à cocher en regard de l'annotation que vous souhaitez masquer sur le graphique.

f. Pour supprimer une annotation, choisissez X sous la colonne Actions.

10. Pour personnaliser l'axe Y, choisissez Options. Sous Left Y-axis (Axe Y de gauche), vous pouvez saisir une étiquette personnalisée dans Label (Étiquette). Si le graphique affiche des valeurs sur l'axe Y de droite, vous pouvez personnaliser cette étiquette également. Vous pouvez également définir des minima et des maxima aux valeurs de l'axe Y pour que le graphique affiche uniquement la plage de valeurs que vous spécifiez.
11. Une fois les modifications terminées, choisissez Update widget (Mettre à jour le widget).

Pour masquer ou modifier la position d'une légende de graphe

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), puis choisissez un tableau de bord.
3. Dans le coin supérieur droit du graphique que vous souhaitez modifier, choisissez Widget actions (Actions du widget). Choisissez Legend (Légende), puis sélectionnez Hidden (Masqué), Bottom (Bas) ou Right (Droite).

Pour masquer temporairement des métriques pour un graphique de tableau de bord

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), puis choisissez un tableau de bord.
3. Sélectionnez le carré de couleur de la métrique que vous souhaitez masquer dans le pied de page du graphique. Un X s'affiche dans le carré de couleur lorsque vous passez la souris dessus, puis il devient gris lorsque vous le choisissez.
4. Pour restaurer la métrique masquée, désactivez la case X dans le carré gris.

Original interface

Pour modifier un graphique de tableau de bord

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), puis choisissez un tableau de bord.
3. Passez la souris sur le coin supérieur droit du graphique que vous souhaitez modifier. Choisissez Widget actions (Actions du widget), puis Edit (Modifier).
4. Pour modifier le titre du graphique, choisissez l'icône en forme de crayon et papier en regard du titre actuel, puis saisissez le nouveau titre.
5. Pour modifier l'intervalle de temps du graphique, choisissez l'un des intervalles de temps prédéfinis dans la partie supérieure du graphique. Celles-ci durent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j, ou1 sem).
   - Pour définir votre propre plage de temps, choisissez custom (personnalisée).
6. Pour modifier le type de widget de votre graphique, sélectionnez l'onglet Graph options (Options du graphique). Vous pouvez choisir Line (Lignes), Stacked area (Aires empilées), Number (Chiffres), Bar (Barres) ou Pie (Secteurs).

   > **Tip**
   >
   > Vous pouvez modifier le type de widget de votre graphique en choisissant la liste déroulante située à côté des plages de temps prédéfinies.

7. Dans Graphed metrics (Métriques représentées graphiquement), vous pouvez ajouter une étiquette dynamique à votre métrique et modifier l'étiquette, la couleur d'étiquette, les statistiques et la période de votre métrique. Vous pouvez également définir la position des étiquettes sur l'axe Y de gauche à droite.

a. Pour ajouter une étiquette dynamique à une métrique, choisissez Dynamic labels (Étiquettes dynamiques). Les étiquettes dynamiques affichent une statistique concernant la métrique dans la légende du graphique. Les étiquettes dynamiques se mettent à jour automatiquement lorsque votre tableau de bord ou votre graphique s'actualise. Par défaut, les valeurs dynamiques que vous ajoutez aux étiquettes s'affichent au début des étiquettes. Pour plus d'informations, consultez Utiliser des étiquettes dynamiques.

b. Pour modifier la couleur d'une métrique, choisissez le carré de couleur situé à côté de la métrique.

c. Pour modifier les statistiques, choisissez la valeur de statistique dans la colonne Statistic (Statistique), puis choisissez une nouvelle valeur. Pour plus d'informations, consultez Statistiques.

d. Pour modifier la période, choisissez la valeur de période dans la colonne Period (Période), puis choisissez une nouvelle valeur.

8. Pour ajouter ou modifier des annotations horizontales, choisissez Options de graphique :

a. Pour ajouter une annotation horizontale, choisissez Add horizontal annotation (Ajouter une annotation horizontale).

b. Pour Label (Étiquette), choisissez l'icône en forme de crayon à côté de l'annotation actuelle. Saisissez ensuite votre nouvelle annotation. Après avoir saisi votre annotation, choisissez l'icône de coche.

c. Pour Value (Valeur), choisissez l'icône en forme de crayon à côté de la valeur de métrique actuelle. Saisissez ensuite votre nouvelle valeur de métrique. Après avoir saisi votre valeur, sélectionnez l'icône de coche.

d. Pour Fill (Remplir), choisissez la liste déroulante sous la colonne, puis spécifiez comment votre annotation utilisera l'ombrage. Vous pouvez choisir None (Aucun), Above (Au-dessus), Between (Entre) ou Below (En-dessous). Si vous choisissez Between (Entre), un nouveau champ d'étiquette et de valeur apparaît.

> **Tip**
>
> Vous pouvez modifier la couleur de remplissage en choisissant le carré de couleur en regard de l'annotation.

e. Pour Axis (Axe), indiquez si votre annotation apparaît sur le côté droit ou gauche de l'axe des Y.

f. Pour masquer une annotation, désélectionnez la case à cocher en regard de l'annotation que vous souhaitez masquer sur le graphique.

g. Pour supprimer une annotation, choisissez X sous la colonne Actions.

> **Note**
>
> Répétez ces étapes pour ajouter plusieurs annotations horizontales au même graphique.

9. Pour ajouter ou modifier des annotations horizontales, choisissez Graph options (Options de graphique) :

a. Pour ajouter une annotation verticale, choisissez Add vertical annotation (Ajouter une annotation verticale).

b. Pour Label (Étiquette), choisissez l'icône en forme de crayon à côté de l'annotation actuelle. Saisissez ensuite votre nouvelle annotation. Après avoir saisi votre annotation, choisissez l'icône de coche.

> **Tip**
>
> Pour afficher seulement la date et l'heure, ne renseignez pas le champ Label (Étiquette).

c. Pour Date, choisissez l'icône en forme de crayon à côté de la date et de l'heure actuelles. Saisissez ensuite la nouvelle date et l'heure.

d. Pour Fill (Remplir), choisissez la liste déroulante sous la colonne, puis spécifiez comment votre annotation utilisera l'ombrage. Vous pouvez choisir None (Aucun), Above (Au-dessus), Between (Entre) ou Below (En-dessous). Si vous choisissez Between (Entre), un nouveau champ d'étiquette et de valeur apparaît.

> **Tip**
>
> Vous pouvez modifier la couleur de remplissage en choisissant le carré de couleur en regard de l'annotation.

> **Note**
>
> Répétez ces étapes pour ajouter plusieurs annotations verticales au même graphique.

e. Pour masquer une annotation, désélectionnez la case à cocher en regard de l'annotation que vous souhaitez masquer sur le graphique.

f. Pour supprimer une annotation, choisissez X sous la colonne Actions.

10. Pour personnaliser l'axe des Y, choisissez Options de graphique. Sous Left Y-axis (Axe Y de gauche), vous pouvez saisir une étiquette personnalisée dans Label (Étiquette). Si le graphique affiche des valeurs sur l'axe Y de droite, vous pouvez personnaliser cette étiquette également. Vous pouvez également définir des minima et des maxima aux valeurs de l'axe Y pour que le graphique affiche uniquement la plage de valeurs que vous spécifiez.
11. Une fois les modifications terminées, choisissez Update widget (Mettre à jour le widget).

Pour masquer ou modifier la position d'une légende de graphe

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), puis choisissez un tableau de bord.
3. Passez la souris sur le coin supérieur droit du graphique que vous souhaitez modifier, puis choisissez Widget actions (Actions du widget). Choisissez Legend (Légende), puis sélectionnez Hidden (Masqué), Bottom (Bas) ou Right (Droite).

Pour masquer temporairement des métriques pour un graphique de tableau de bord

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), puis choisissez un tableau de bord.
3. Sélectionnez le carré de couleur de la métrique que vous souhaitez masquer dans le pied de page du graphique. Un X s'affiche dans le carré de couleur lorsque vous passez la souris dessus, puis il devient gris lorsque vous le choisissez.
4. Pour restaurer la métrique masquée, désactivez la case X dans le carré gris.

Déplacer ou redimensionner un graphique sur un tableau de bord CloudWatch

Vous pouvez organiser et redimensionner les graphiques sur votre CloudWatch tableau de bord.

Pour déplacer un graphique vers 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Effectuez l'une des actions suivantes :
   - Passez le curseur sur le titre du graphique jusqu'à ce que l'icône de sélection apparaisse. Sélectionnez et faites glisser le graphique vers un nouvel emplacement sur le tableau de bord.
   - Pour déplacer le widget en haut à gauche ou en bas à gauche du tableau de bord, choisissez les points de suspension verticaux en haut à droite du widget pour ouvrir le menu Actions de widget. Choisissez ensuite Déplacer, puis choisissez où déplacer le widget.
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Redimensionner un graphique

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), puis choisissez un tableau de bord.
3. Pour augmenter ou diminuer la taille, passez le curseur sur le graphique et faites glisser le côté inférieur droit du graphique. Arrêtez de faire glisser le coin une fois la taille souhaitée obtenue.
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Agrandir un graphique temporairement

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), puis choisissez un tableau de bord.
3. Sélectionnez le graphique. Vous pouvez également passer le curseur sur le titre du graphique puis choisir Widget actions (Actions de widget), Enlarge (Élargir).

Renommer un graphique sur un tableau de bord CloudWatch

Vous pouvez modifier le nom par défaut attribué à CloudWatch un graphique sur votre tableau de bord.

Pour renommer un graphique 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Passez le curseur sur le titre du graphique et choisissez Actions de widget, puis Edit (Modifier).
4. Dans la fenêtre Edit graph (Modifier le graphique), près de la partie supérieure, choisissez le titre du graphique.
5. Dans Title (Titre), saisissez un nouveau nom et choisissez Ok (coche). Dans le coin inférieur droit de la fenêtre Edit graph (Modifier le graphique), choisissez Update widget (Mettre à jour le widget).

Ajouter un widget d'explorateur de métriques à un CloudWatch tableau de bord

Les widgets Explorateur de métriques incluent des graphiques de plusieurs ressources ayant la même balise ou partageant la même propriété de ressource, telle qu'un type d'instance. Ces widgets restent à jour, car les ressources correspondant sont créées ou supprimées. L'ajout de widgets Explorateur de métriques à votre tableau de bord vous aide à résoudre les problèmes de votre environnement plus efficacement.

Par exemple, vous pouvez surveiller votre parc d'instances EC2 en attribuant des balises qui représentent leurs environnements, tels que la production ou le test. Vous pouvez ensuite utiliser ces balises pour filtrer et regrouper les métriques opérationnelles, telles que `CPUUtilization`, pour comprendre l'état et les performances des instances EC2 associées à chaque balise.

Les étapes suivantes expliquent comment ajouter un widget Explorateur de métriques à un tableau de bord à l'aide de la console. Vous pouvez également l'ajouter par programmation ou en utilisant. AWS CloudFormation Pour plus d'informations, consultez les sections Définition de l'objet du widget Metrics Explorer et AWS::CloudWatch::Dashboard.

Pour ajouter un widget Explorateur de métriques à 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 Dashboards (Tableaux de bord).
3. Choisissez le nom du tableau de bord dans lequel vous souhaitez ajouter le widget Explorateur de métriques.
4. Cliquez sur le symbole +.
5. Sélectionnez Explorer (Explorateur), puis Next (Suivant).

> **Note**
>
> Vous devez avoir activé la nouvelle vue du tableau de bord pour pouvoir ajouter un widget Explorateur de métriques. Pour ce faire, sélectionnez Dashboards (Tableaux de bord) dans le panneau de navigation, puis choisissez try out the new interface (Essayer la nouvelle interface) dans la bannière en haut de la page.

6. Effectuez l'une des actions suivantes :
   - Pour utiliser un modèle, choisissez Pre-filled Explorer widget (Widget Explorateur prérempli), puis sélectionnez un modèle à utiliser.
   - Pour créer une visualisation personnalisée, choisissez Empty Explorer widget (Widget Explorateur vide).
7. Choisissez Créer.

   Si vous avez utilisé un modèle, le widget apparaît sur votre tableau de bord avec les métriques sélectionnées. Si le widget Explorateur et le tableau de bord vous conviennent, sélectionnez Save dashboard (Enregistrer le tableau de bord).

   Si vous n'avez pas utilisé de modèle, passez aux étapes suivantes.
8. Dans le nouveau widget, sous Explorer (Explorateur), dans la case Metrics (Métriques), choisissez une seule métrique ou toutes les métriques disponibles d'un service.

   Une fois que vous avez choisi une métrique, vous pouvez éventuellement répéter cette étape pour ajouter d'autres métriques.
9. Pour chaque métrique sélectionnée, CloudWatch affiche la statistique qu'elle utilisera immédiatement après le nom de la métrique. Pour la modifier, sélectionnez le nom de la statistique, puis sélectionnez la statistique de votre choix.

10. Sous From (De), choisissez une balise ou une propriété de ressource pour filtrer vos résultats.

    Si vous le souhaitez, vous pouvez ensuite répéter cette étape pour choisir d'autres balises ou propriétés de ressource.

    Si vous choisissez plusieurs valeurs de la même propriété, telles que deux types d'instances EC2, l'explorateur affiche toutes les ressources correspondant à l'une ou l'autre des propriétés choisies. L'opération est alors considérée comme une opération OR.

    Si vous choisissez des propriétés ou des balises différentes, telles que la balise **`Production`** et le type d'instance M5, seules les ressources correspondant à toutes ces sélections sont affichées. L'opération est alors considérée comme une opération AND.

11. (Facultatif) Pour Aggregate by (Regrouper par), choisissez une statistique à utiliser pour regrouper les métriques. Ensuite, à côté de for (Pour), choisissez le mode de regroupement de la métrique de la liste. Vous pouvez regrouper toutes les ressources actuellement affichées ou les regrouper par une seule balise ou propriété de ressource.

    Selon la méthode de regroupement que vous sélectionnez, le résultat peut être donner lieu à une seule série temporelle ou à plusieurs séries temporelles.

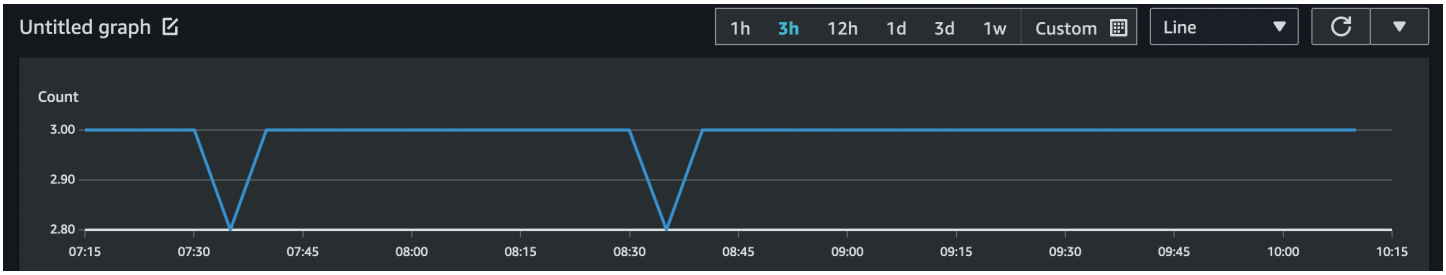
12. Sous Split by (Fractionner par), vous pouvez choisir de fractionner un seul graphique avec plusieurs séries temporelles en plusieurs graphiques. Le fractionnement peut être réalisé selon une variété de critères, que vous choisissez sous l'option Split by (Fractionner par).

13. Sous Graph options (Options de graphique), vous pouvez affiner le graphique en modifiant la période, le type de graphique, le placement de la légende et la mise en page.

14. Si le widget Explorateur et le tableau de bord vous conviennent, sélectionnez Save dashboard (Enregistrer le tableau de bord).

Ajouter ou supprimer un widget linéaire sur un CloudWatch tableau de bord

Avec le widget de ligne, vous pouvez comparer les métriques sur des périodes de temps. Vous pouvez également utiliser la fonction de zoom de la mini-carte du widget pour inspecter des sections de graphiques linéaires sans basculer entre les vues zoomée et dézoomée. Les procédures décrites dans cette section décrivent comment ajouter et supprimer un widget linéaire sur un CloudWatch tableau de bord. Pour plus d'informations sur l'utilisation de la fonction de zoom de la mini-carte du widget avec les graphiques linéaires, consultez Zooming in on a line or stacked area graph (Zoom avant sur un graphique linéaire ou à aires empilées).



Pour ajouter un widget de ligne 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Cliquez sur le symbole +, puis sélectionnezLine.
4. Choisissez Métriques.
5. Choisissez Browse (Naviguer), puis sélectionnez la métrique que vous souhaitez représenter graphiquement.
6. Choisissez Create widget (Créer un widget), puis choisissez Save dashboard (Enregistrer le tableau de bord).

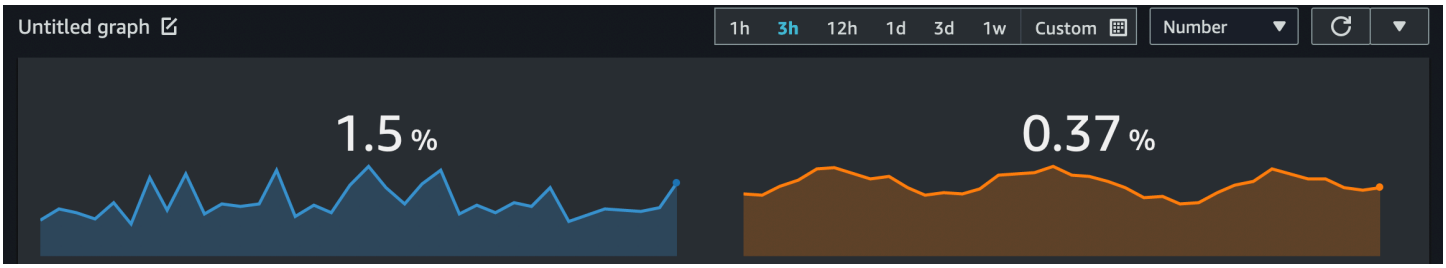
Pour supprimer un widget de ligne d'un tableau de bord

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Dans le coin supérieur droit du widget de ligne que vous souhaitez supprimer, choisissez Widget actions (Actions du widget), puis choisissez Delete (Supprimer).
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Ajouter ou supprimer un widget numérique dans un CloudWatch tableau de bord

Avec le widget de nombre, vous pouvez consulter les dernières valeurs et tendances des métriques dès qu'elles apparaissent. Comme le widget de nombre inclut la fonction Sparkline, vous pouvez visualiser les moitiés supérieure et inférieure des tendances des métriques dans un seul graphique.

Les procédures décrites dans cette section décrivent comment ajouter et supprimer un widget numérique dans un CloudWatch tableau de bord.



> **Note**
>
> Seule la nouvelle interface prend en charge la fonction Sparkline. Lorsque vous créez un widget de nombre, la fonction Sparkline est automatiquement incluse.

Pour ajouter un widget de nombre 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Cliquez sur le symbole +, puis sélectionnez Numéro.
4. Dans l'onglet Parcourir, recherchez ou parcourez les métriques que vous souhaitez afficher.
5. (Facultatif) Pour modifier la couleur de la fonction Sparkline, choisissez Graphes metrics (Métriques représentées graphiquement), puis sélectionnez la zone de couleur en regard de l'étiquette de métrique. Un menu s'affiche, dans lequel vous pouvez choisir une couleur différente ou saisir un code couleur hexadécimal à six chiffres pour préciser une couleur.
6. (Facultatif) Pour désactiver la fonction Sparkline, choisissez Options. Sous Sparkline, cochez la case.
7. (Facultatif) Pour modifier l'intervalle de temps de votre widget de nombre, sélectionnez l'un des intervalles de temps prédéfinis dans la partie supérieure du widget. Les plages de temps durent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j, ou 1 sem).

   Pour définir votre propre plage de temps, choisissez Custom (Personnalisée).

   - (Facultatif) Pour que ce widget continue d'utiliser l'intervalle de temps que vous sélectionnez, même si l'intervalle de temps pour le reste du tableau de bord est modifié ultérieurement, choisissez Conserver l'intervalle de temps.

8. (Facultatif) Pour que le widget numérique affiche un agrégat (1h, 3h, 12h, 1d, 3d ou 1w).

   Pour définir votre propre plage de temps, choisissez Custom (Personnalisée).

   - (Facultatif) Pour que ce widget affiche une moyenne de la valeur de la métrique sur toute la plage de temps, au lieu de la valeur la plus récente, choisissez Options, la valeur de la plage de temps indique la valeur de la plage de temps complète.
9. Choisissez Create widget (Créer un widget), puis choisissez Save dashboard (Enregistrer le tableau de bord).

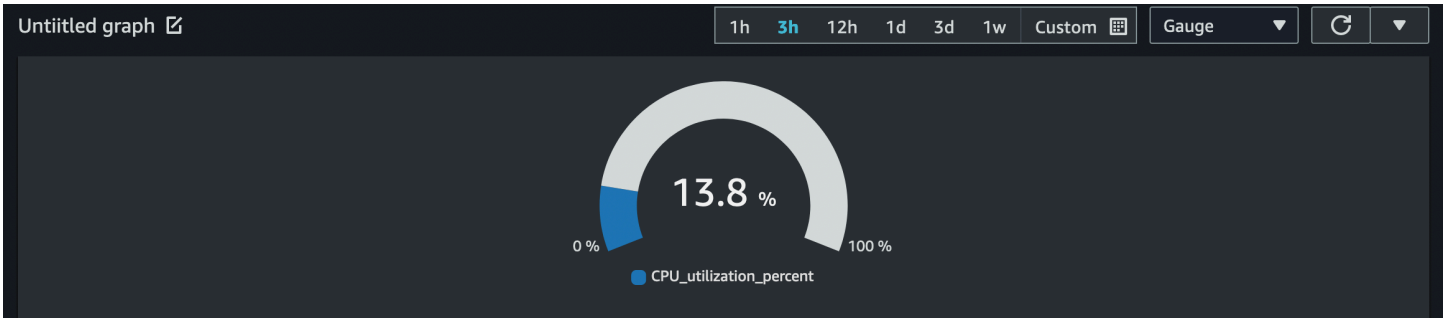
> **Tip**
>
> Vous pouvez désactiver la fonction Sparkline à partir du widget de nombre sur l'écran du tableau de bord. Dans le coin supérieur droit du widget de nombre que vous souhaitez modifier, choisissez Widget actions (Actions du widget). Sélectionnez Sparkline, puis choisissez Hide Sparkline (Masquer Sparkline).

Pour supprimer un widget de nombre d'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 Dashboards (Tableaux de bord), puis choisissez le tableau de bord qui contient le widget de nombre que vous souhaitez supprimer.
3. Dans le coin supérieur droit du widget de nombre que vous souhaitez supprimer, choisissez Widget actions (Actions du widget), puis choisissez Delete (Supprimer).
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Ajouter ou supprimer un widget de jauge dans un CloudWatch tableau de bord

Avec le widget de jauge, vous pouvez visualiser des valeurs de métriques qui figurent entre les plages. Par exemple, vous pouvez utiliser le widget de jauge pour représenter graphiquement des pourcentages et l'utilisation du CPU, de sorte que vous puissiez observer et diagnostiquer tous les problèmes de performance qui surviennent. Les procédures décrites dans cette section décrivent comment ajouter et supprimer un widget de jauge dans un CloudWatch tableau de bord.



> **Note**
>
> Seule la nouvelle interface de la CloudWatch console prend en charge la création du widget de jauge. Vous devez définir une plage de jauges lorsque vous créez ce widget.

Pour ajouter un widget de jauge 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Depuis l'écran du tableau de bord, choisissez + (Ajouter un widget), puis sélectionnez Gauge (Jauge).
4. Choisissez Browse (Naviguer), puis sélectionnez la métrique que vous souhaitez représenter graphiquement.
5. Choisissez Options. Sous Gauge range (Plage de jauges), définissez les valeurs de Min (Minimum) et Max (Maximum). Pour les pourcentages, comme l'utilisation du CPU, nous vous recommandons de définir les valeurs de `Min` sur `0`, et de `Max` sur `100`.
6. (Facultatif) Pour modifier la couleur du widget de jauge, choisissez Graphes metrics (Métriques représentées graphiquement), puis sélectionnez la zone de couleur en regard de l'étiquette de métrique. Un menu s'affiche, dans lequel vous pouvez choisir une couleur différente ou saisir un code couleur hexadécimal à six chiffres pour préciser une couleur.
7. Choisissez Create widget (Créer un widget), puis choisissez Save dashboard (Enregistrer le tableau de bord).

Pour supprimer un widget de jauge d'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 Dashboards (Tableaux de bord), puis choisissez le tableau de bord qui contient le widget de jauge que vous souhaitez supprimer.
3. Dans le coin supérieur droit du widget de jauge que vous souhaitez supprimer, choisissez Widget actions (Actions du widget), et choisissez Delete (Supprimer).
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Ajouter un widget personnalisé à un CloudWatch tableau de bord

Un widget personnalisé est un widget CloudWatch de tableau de bord qui peut appeler n'importe quelle AWS Lambda fonction avec des paramètres personnalisés. Il affiche ensuite le code HTML ou JSON renvoyé. Les widgets personnalisés permettent de créer simplement une vue de données personnalisée sur un tableau de bord. Si vous pouvez écrire du code Lambda et créer du code HTML, vous pouvez créer un widget personnalisé utile. En outre, Amazon fournit plusieurs widgets personnalisés prédéfinis que vous pouvez créer sans code.

Lorsque vous créez une fonction Lambda à utiliser comme widget personnalisé, nous vous recommandons fortement d'inclure le préfixe customWidget dans le nom de fonction. Cela vous aide à déterminer quelles fonctions Lambda sont sûres à utiliser lorsque vous ajoutez des widgets personnalisés à votre tableau de bord.

Les widgets personnalisés se comportent comme d'autres widgets sur votre tableau de bord. Ils peuvent être actualisés et actualisés automatiquement, redimensionnés et déplacés d'un endroit à un autre. Ils réagissent à la plage de temps du tableau de bord.

Si vous avez configuré CloudWatch la fonctionnalité multi-comptes de console, vous pouvez ajouter un widget personnalisé créé dans un compte aux tableaux de bord d'autres comptes. Pour plus d'informations, consultez Console multicompte et multirégion CloudWatch .

Vous pouvez également utiliser des widgets personnalisés sur votre propre site Web en utilisant la fonctionnalité de partage CloudWatch de tableau de bord. Pour plus d'informations, consultez Partage de CloudWatch tableaux de bord.

Rubriques

- Détails sur les widgets personnalisés
- Sécurité et JavaScript
- Interactivité dans le widget personnalisé

- Création d'un widget personnalisé
- Exemples de widgets personnalisés

Détails sur les widgets personnalisés

Les widgets personnalisés fonctionnent comme suit :

1. Le CloudWatch tableau de bord appelle la fonction Lambda contenant le code du widget. Il transmet tous les paramètres personnalisés définis dans le widget.
2. La fonction Lambda renvoie une chaîne de HTML, JSON ou Markdown. Markdown est renvoyé en tant que JSON au format suivant :

   ```
   {"markdown":"markdown content"}
   ```

3. Le tableau de bord affiche le code HTML ou JSON renvoyé.

Si la fonction renvoie du code HTML, la plupart des balises HTML sont prises en charge. Vous pouvez utiliser des feuilles de style en cascade (CSS, ou Cascading Style Sheets) et des graphiques vectoriels adaptables (SVG, ou Scalable Vector Graphics) pour créer des vues sophistiquées.

Le style par défaut des éléments HTML tels que les liens et les tableaux suit le style des CloudWatch tableaux de bord. Vous pouvez personnaliser ce style à l'aide de styles inclus, à l'aide de la balise `<style>`. Vous pouvez également désactiver les styles par défaut en incluant un seul élément HTML avec la classe de `cwdb-no-default-styles`. L'exemple suivant désactive les styles par défaut : `<div class="cwdb-no-default-styles"></div>`.

Chaque appel par un widget personnalisé à Lambda inclut un élément `widgetContext` avec le contenu suivant, afin de fournir au développeur de la fonction Lambda des informations contextuelles utiles.

```
{
    "widgetContext": {
        "dashboardName": "Name-of-current-dashboard",
        "widgetId": "widget-16",
        "accountId": "012345678901",
        "locale": "en",
        "timezone": {
            "label": "UTC",
```

```
            "offsetISO": "+00:00",
            "offsetInMinutes": 0
        },
        "period": 300,
        "isAutoPeriod": true,
        "timeRange": {
            "mode": "relative",
            "start": 1627236199729,
            "end": 1627322599729,
            "relativeStart": 86400012,
            "zoom": {
                "start": 1627276030434,
                "end": 1627282956521
             }
        },
        "theme": "light",
        "linkCharts": true,
        "title": "Tweets for Amazon website problem",
        "forms": {
            "all": {}
        },
        "params": {
            "original": "param-to-widget"
        },
        "width": 588,
        "height": 369
    }
}
```

Mise en forme CSS par défaut

Les widgets personnalisés fournissent les éléments de mise en forme CSS par défaut suivants :

- Vous pouvez utiliser la classe CSS btn pour ajouter un bouton. Celle-ci a pour effet de transformer un point d'ancrage (<a>) en bouton, comme dans l'exemple suivant :

  ```
  <a class="btn" href=https://amazon.com”>Open Amazon</a>
  ```

- Vous pouvez utiliser la classe CSS btn btn-primary pour ajouter un bouton principal.
- Les éléments suivants sont mise en forme par défaut :table (Tableau), select (Sélection), headers (h1, h2, and h3) (En-têtes [h1, h2 et h3]), preformatted text (pre) (Texte préformaté [pré]), input (Saisie) et text area (Zone de texte).

Utilisation du paramètre describe (Description)

Nous vous recommandons fortement de prendre en charge le paramètre describe (Description) dans vos fonctions, même si celui-ci renvoie simplement une chaîne vide. Si vous ne le prenez pas en charge et qu'il est appelé dans le widget personnalisé, le contenu du widget est affiché comme s'il s'agissait de documentation.

Si vous intégrez le paramètre describe (Description), la fonction Lambda renvoie la documentation au format Markdown et ne fait rien d'autre.

Lorsque vous créez un widget personnalisé dans la console, un bouton Get documentation (Obtenir de la documentation) apparaît après avoir sélectionné la fonction Lambda. Si vous cliquez sur ce bouton, la fonction est appelée avec le paramètre describe (Description) et la documentation de la fonction est renvoyée. Si la documentation est bien formatée en Markdown, CloudWatch analyse la première entrée de la documentation entourée de trois caractères simples (```) en YAML. Ensuite, il remplit automatiquement la documentation dans les paramètres. Voici un exemple de documentation correctement formatée.

```
``` yaml
echo: <h1>Hello world</h1>
```
```

Sécurité et JavaScript

Pour des raisons de sécurité, n' JavaScript est pas autorisé dans le code HTML renvoyé. La suppression permet d' JavaScript éviter les problèmes d'escalade des autorisations, dans lesquels l'auteur de la fonction Lambda injecte du code susceptible de s'exécuter avec des autorisations plus élevées que celles de l'utilisateur visualisant le widget sur le tableau de bord.

Si le code HTML renvoyé contient JavaScript du code ou d'autres failles de sécurité connues, il est effacé du code HTML avant d'être affiché sur le tableau de bord. Par exemple, les balises <iframe> et <use> ne sont pas autorisées et sont supprimées.

Les widgets personnalisés ne seront pas exécutés par défaut dans un tableau de bord. Au lieu de cela, vous devez autoriser explicitement l'exécution d'un widget personnalisé si vous faites confiance à la fonction Lambda à laquelle il fait appel. Vous pouvez choisir de l'autoriser une fois ou de l'autoriser toujours, tant pour les widgets individuels que pour l'ensemble du tableau de bord. Vous pouvez également refuser l'autorisation pour des widgets individuels et l'ensemble du tableau de bord.

Interactivité dans le widget personnalisé

Même si JavaScript ce n'est pas autorisé, il existe d'autres moyens d'autoriser l'interactivité avec le code HTML renvoyé.

- Tout élément du code HTML renvoyé peut être balisé avec une configuration spéciale dans une balise `<cwdb-action>`, qui peut afficher des informations dans des fenêtres contextuelles, demander une confirmation avant un clic et appeler n'importe quelle fonction Lambda lorsque cet élément est choisi. Par exemple, vous pouvez définir des boutons qui appellent n'importe quelle AWS API à l'aide d'une fonction Lambda. Le code HTML renvoyé peut être configuré pour remplacer le contenu du widget Lambda existant, ou pour s'afficher dans une boîte de dialogue modale.
- Le code HTML renvoyé peut inclure des liens qui ouvrent de nouvelles consoles, ouvrent d'autres pages client ou chargent d'autres tableaux de bord.
- Le code HTML peut inclure l'attribut `title` pour un élément, qui donne des informations supplémentaires si l'utilisateur passe la souris sur cet élément.
- L'élément peut inclure des sélecteurs CSS, tels que `:hover`, qui peut invoquer des animations ou d'autres effets CSS. Vous pouvez également afficher ou masquer des éléments dans la page.

Définition et utilisation de <cwdb-action>

L'élément `<cwdb-action>` définit un comportement sur l'élément précédent immédiat. Le contenu de `<cwdb-action>` équivaut à du code HTML pour l'affichage, ou à un bloc JSON de paramètres pour passer à une fonction Lambda.

Voici un exemple d'un élément `<cwdb-action>`.

```
<cwdb-action
     action="call|html"
     confirmation="message"
     display="popup|widget"
     endpoint="<lambda ARN>"
     event="click|dblclick|mouseenter">

     html | params in JSON
</cwdb-action>
```

- action — Les valeurs valides sont `call`, qui appelle une fonction Lambda, et `html`, qui affiche tout code HTML contenu dans `<cwdb-action>`. La valeur par défaut est `html`.

- confirmation — Affiche un message de confirmation devant être accepté avant que l'action ne soit prise, ce qui permet au client d'annuler l'opération.
- display (afficher) : les valeurs valides sont `popup` et `widget`, qui remplacent le contenu du widget lui-même. L'argument par défaut est `widget`.
- endpoint — Amazon Resource Name (ARN) de la fonction Lambda à appeler. Cet élément est obligatoire si `action` est `call`.
- event — Définit l'événement sur l'élément précédent qui appelle l'action. Les valeurs valides sont `click`, `dblclick` et `mouseenter`. Le `mouseenter` peut être utilisé avec l'action `html`. L'argument par défaut est `click`.

Exemples

Voici un exemple qui montre comment utiliser la balise `<cwdb-action>` pour créer un bouton qui redémarre une instance Amazon EC2 à l'aide d'un appel de fonction Lambda. Ainsi, la réussite ou l'échec de l'appel est affiché dans une fenêtre contextuelle.

```
<a class="btn">Reboot Instance</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:rebootInstance" display="popup">
       { "instanceId": "i-342389adbfef" }
</cwdb-action>
```

L'exemple suivant affiche davantage d'informations dans une fenêtre contextuelle.

```
<a>Click me for more info in popup</a>
<cwdb-action display="popup">
   <h1>Big title</h1>
   More info about <b>something important</b>.
</cwdb-action>
```

Cet exemple représente un bouton Next (Suivant) qui remplace le contenu d'un widget par un appel à une fonction Lambda.

```
<a class="btn btn-primary">Next</a>
<cwdb-action action="call" endpoint="arn:aws:lambda:us-
east-1:123456:function:nextPage">
   { "pageNum": 2 }
</cwdb-action>
```

Création d'un widget personnalisé

Pour créer un widget personnalisé, vous pouvez utiliser l'un des exemples fournis par AWS ou créer le vôtre. Les AWS exemples incluent des échantillons à la fois en Python JavaScript et sont créés par une AWS CloudFormation pile. Pour obtenir une liste d'exemples, consultez Exemples de widgets personnalisés.

Pour créer un widget personnalisé dans un CloudWatch tableau de bord

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), puis choisissez un tableau de bord.
3. Cliquez sur le symbole +.
4. Choisissez Custom widget (Widget personnalisé).
5. Utilisez l'une des méthodes suivantes :
   - Pour utiliser un exemple de widget personnalisé fourni par AWS, procédez comme suit :
     a. Sélectionnez l'exemple dans la liste déroulante.

        La AWS CloudFormation console s'ouvre dans un nouveau navigateur. Dans la AWS CloudFormation console, effectuez les opérations suivantes :
     b. (Facultatif) Personnalisez le nom de la AWS CloudFormation pile.
     c. Procédez à des sélections pour tous les paramètres utilisés par l'exemple.
     d. Sélectionnez Je reconnais que cela AWS CloudFormation pourrait créer des ressources IAM, puis choisissez Create stack.
   - Pour créer votre propre widget personnalisé fourni par AWS, procédez comme suit :
     a. Choisissez Next (Suivant).
     b. Décidez soit de sélectionner votre fonction Lambda depuis une liste, soit de saisir son Amazon Resource Name (ARN). Si vous la sélectionnez depuis une liste, spécifiez également la région où se trouve la fonction et la version à utiliser.
     c. Pour Parameters (Paramètres), procédez à des sélections pour tous les paramètres utilisés par la fonction.
     d. Saisissez un titre pour le widget.
     e. Pour Update on (Déclencheur de mise à jour), configurez le moment lors duquel le widget doit être mis à jour (lorsque la fonction Lambda doit être appelée à nouveau).

Il peut s'agir d'un ou de plusieurs des éléments suivants : Refresh (Actualiser) pour le mettre à jour lors de l'actualisation automatique du tableau de bord ; Resize (Redimensionner) pour le mettre à jour chaque fois que le widget est redimensionné ; ou Time Range (Plage de temps) pour le mettre à jour chaque fois que la plage de temps du tableau de bord est ajustée, y compris lorsque les graphiques sont zoomés.

f. Si la prévisualisation vous convient, choisissez Create widget (Créer un widget).

Exemples de widgets personnalisés

AWS fournit des exemples de widgets personnalisés à la fois en Python JavaScript et en Python. Vous pouvez créer ces exemples de widgets en utilisant le lien correspondant à chaque widget de cette liste. Vous pouvez également créer et personnaliser un widget à l'aide de la CloudWatch console. Les liens de cette liste ouvrent une AWS CloudFormation console et utilisent un lien de AWS CloudFormation création rapide pour créer le widget personnalisé.

Vous pouvez également accéder aux exemples de widgets personnalisés sur GitHub.

Après cette liste, des exemples complets du widget Echo sont affichés pour chaque langue.

JavaScript

Exemples de widgets personnalisés dans JavaScript

- Echo — Dispositif d'écho de base que vous pouvez utiliser pour tester la manière dont le code HTML apparaît dans un widget personnalisé, sans avoir à écrire un nouveau widget.
- Hello World — Widget de démarrage très basique.
- Débogueur de widgets personnalisés — Widget débogueur qui affiche des informations utiles sur l'environnement d'exécution Lambda.
- Query CloudWatch Logs Insights — Exécutez et modifiez CloudWatch les requêtes Logs Insights.
- Exécution de requêtes Amazon Athena — Exécutez et modifiez les requêtes Athena.
- AWS API d'appel — Appelez n'importe quelle AWS API en lecture seule et affichez les résultats au format JSON.
- Graphe CloudWatch bitmap rapide — Effectuez le rendu CloudWatch des graphiques à l'aide du serveur, pour un affichage rapide.
- Widget de texte depuis le CloudWatch tableau de bord — Affiche le premier widget de texte du tableau de CloudWatch bord spécifié.

- CloudWatch données métriques sous forme de tableau : affiche les données CloudWatch métriques brutes dans un tableau.
- Tableau Amazon EC2— Affiche les instances EC2 supérieures par utilisation du CPU. Ce widget comprend également un bouton de redémarrage, désactivé par défaut.
- AWS CodeDeploy déploiements : affiche les CodeDeploy déploiements.
- AWS Cost Explorer rapport — Affiche un rapport sur le coût de chaque AWS service pour une période sélectionnée.
- Affichage du contenu de l'URL externe — Affiche le contenu d'une URL accessible en externe.
- Affichage d'un objet Amazon S3 — Affiche un objet dans un compartiment Amazon S3 dans votre compte.
- Graphique circulaire SVG simple — Exemple d'un widget graphique basé sur SVG.

Python

Exemples de widgets personnalisés en Python

- Echo — Dispositif d'écho de base que vous pouvez utiliser pour tester la manière dont le code HTML apparaît dans un widget personnalisé, sans avoir à écrire un nouveau widget.
- Hello World — Widget de démarrage très basique.
- Débogueur de widgets personnalisés — Widget débogueur qui affiche des informations utiles sur l'environnement d'exécution Lambda.
- AWS API d'appel — Appelez n'importe quelle AWS API en lecture seule et affichez les résultats au format JSON.
- Graphe CloudWatch bitmap rapide — Effectuez le rendu CloudWatch des graphiques à l'aide du serveur, pour un affichage rapide.
- Envoi d'un instantané de tableau de bord par e-mail — Permet d'effectuer un instantané du tableau de bord actuel et de l'envoyer aux destinataires d'e-mail.
- Envoi d'un instantané de tableau de bord Amazon S3 — Créez un instantané du tableau de bord actuel et stockez-le dans Amazon S3.
- Widget de texte depuis le CloudWatch tableau de bord — Affiche le premier widget de texte du tableau de CloudWatch bord spécifié.
- Affichage du contenu de l'URL externe — Affiche le contenu d'une URL accessible en externe.
- Lecteur RSS — Affiche les flux RSS.

- Affichage d'un objet Amazon S3 — Affiche un objet dans un compartiment Amazon S3 dans votre compte.
- Graphique circulaire SVG simple — Exemple d'un widget graphique basé sur SVG.

Widget Echo dans JavaScript

Voici l'exemple de widget Echo dans JavaScript.

```
const DOCS = `
## Echo
A basic echo script. Anything passed in the \`\`\`echo\`\`\` parameter is returned as
 the content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
\`\`\` yaml
echo: <h1>Hello world</h1>
\`\`\`
`;

exports.handler = async (event) => {
    if (event.describe) {
        return DOCS;
    }

    let widgetContext = JSON.stringify(event.widgetContext, null, 4);
    widgetContext = widgetContext.replace(/</g, '&lt;');
    widgetContext = widgetContext.replace(/>/g, '&gt;');

    return `${event.echo || ''}<pre>${widgetContext}</pre>`;
};
```

Widget Echo en Python

Voici un exemple de widget Echo en Python.

```
import json

DOCS = """
```

```
## Echo
A basic echo script. Anything passed in the ```echo``` parameter is returned as the
 content of the custom widget.
### Widget parameters
Param | Description
---|---
**echo** | The content to echo back

### Example parameters
``` yaml
echo: <h1>Hello world</h1>
```"""

def lambda_handler(event, context):
    if 'describe' in event:
        return DOCS

    echo = event.get('echo', '')
    widgetContext = event.get('widgetContext')
    widgetContext = json.dumps(widgetContext, indent=4)
    widgetContext = widgetContext.replace('<', '<')
    widgetContext = widgetContext.replace('>', '>')

    return f'{echo}<pre>{widgetContext}</pre>'
```

Widget Echo en Java

Voici un exemple de widget Echo en Java.

```
package example;

import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;

public class Handler implements RequestHandler<Event, String>{

  static String DOCS = ""
    + "## Echo\n"
    + "A basic echo script. Anything passed in the ```echo``` parameter is returned as
 the content of the custom widget.\n"
    + "### Widget parameters\n"
```

```
    + "Param | Description\n"
    + "---|---\n"
    + "**echo** | The content to echo back\n\n"
    + "### Example parameters\n"
    + "```yaml\n"
    + "echo: <h1>Hello world</h1>\n"
    + "```\n";

  Gson gson = new GsonBuilder().setPrettyPrinting().create();

  @Override
  public String handleRequest(Event event, Context context) {

    if (event.describe) {
      return DOCS;
    }

    return (event.echo != null ? event.echo : "") + "<pre>" +
 gson.toJson(event.widgetContext) + "</pre>";
  }
}

class Event {

    public boolean describe;
    public String echo;
    public Object widgetContext;

    public Event() {}

    public Event(String echo, boolean describe, Object widgetContext) {
        this.describe = describe;
        this.echo = echo;
        this.widgetContext = widgetContext;
    }
}
```

Ajouter ou supprimer un widget de texte dans un CloudWatch tableau de bord

Un widget de texte contient un bloc de texte au format Markdown. Vous pouvez ajouter, modifier ou supprimer des widgets de texte de votre CloudWatch tableau de bord.

Pour ajouter un widget de texte 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Cliquez sur le symbole +.
4. Choisissez Texte.
5. Pour Markdown, ajoutez et mettez en forme votre texte à l'aide de Markdown, puis choisissez Create widget ()Créer un widget.
6. Pour rendre le widget de texte transparent, choisissez Arrière-plan transparent.
7. Choisissez Save dashboard (Enregistrer le tableau de bord).

Pour modifier un widget de texte 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Passez le curseur sur le coin supérieur droit du bloc de texte, puis choisissez Widget actions (Actions du widget). Ensuite, choisissez Edit (Modifier).
4. Mettez le texte à jour si besoin, puis choisissez Update widget (Mettre à jour le widget).
5. Choisissez Save dashboard (Enregistrer le tableau de bord).

Pour supprimer un widget de texte d'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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Passez le curseur sur le coin supérieur droit du bloc de texte, puis choisissez Widget actions (Actions du widget). Ensuite, choisissez Supprimer.
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Ajouter ou supprimer un widget d'alarme dans un CloudWatch tableau de bord

Pour ajouter un widget d'alerte à un tableau de bord, choisissez l'une des options suivantes :

- Ajouter une alerte unique dans un widget, qui affiche à la fois le graphique de la métrique de l'alerte et le statut de l'alerte.
- Ajouter un widget de statut d'alerte, qui affiche le statut de plusieurs alertes dans une grille. Seuls les noms d'alerte et le statut actuel sont indiqués ; les graphiques ne le sont pas. Vous pouvez ajouter un maximum de 100 alertes dans un widget de statut d'alerte.

Pour ajouter une alerte unique, y compris son graphique, à 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 alertes, sélectionnez l'alerte à ajouter, puis choisissez Ajouter au tableau de bord.
3. Sélectionnez un tableau de bord, choisissez un type de widget (Line, Stacked area ou Number), puis choisissez Add to dashboard (Ajouter au tableau de bord).
4. Pour voir votre alerte sur le tableau de bord, choisissez Dashboards (Tableaux de bord) dans le panneau de navigation, puis sélectionnez le tableau de bord.
5. (Facultatif) Pour agrandir temporairement un graphique d'alerte, sélectionnez-le.
6. (Facultatif) Pour modifier le type de widget, passez votre souris au-dessus du titre du graphique, choisissez Actions de widget, puis Type de widget.

Pour ajouter un widget de statut d'alerte à 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Cliquez sur le symbole +.
4. Sélectionnez Alarm status (Statut de l'alerte).
5. Cochez les cases à côté des alertes que vous souhaitez ajouter au widget, puis sélectionnez Create widget (Créer un widget).
6. Choisissez Add to dashboard (Ajouter au tableau de bord).

Pour supprimer un widget d'alerte d'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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Passez votre souris sur le widget, sélectionnez Actions de widget, puis choisissez Supprimer.
4. Choisissez Save dashboard (Enregistrer le tableau de bord). Si vous essayez de quitter le tableau de bord avant d'enregistrer vos modifications, vous êtes invité soit à enregistrer, soit à ignorer vos modifications.

Ajouter ou supprimer un widget de tableau de données dans un CloudWatch tableau de bord

Avec le widget de tableau de données, vous pouvez voir les points de données bruts de votre métrique et un bref résumé de ces données brutes. Comme le widget tableau de données n'est pas un graphique qui fait abstraction des données réelles, il est plus facile de comprendre les points de données présentés. Les procédures décrites dans cette section décrivent comment ajouter et supprimer un widget de table de données dans un CloudWatch tableau de bord.

Propriétés de tableau

Un tableau de données possède un ensemble de propriétés par défaut qui ne nécessitent aucune modification des options ou de la source. Ces propriétés incluent une colonne d'étiquette autocollante, toutes les colonnes de résumé activées, les points de données arrondis et leurs unités converties.

Chaque widget de tableau de données peut avoir les propriétés suivantes. Les informations relatives à chaque propriété incluent la manière de la configurer dans la source JSON du tableau de bord. Pour plus d'informations sur le tableau de bord JSON, veuillez consulter Dashboard Body Structure and Syntax.

Récapitulatif

Les colonnes de résumé sont une nouvelle propriété introduite avec le widget de tableau de données. Ces colonnes constituent un sous-ensemble spécifique des résumés de votre tableau actuel. Par exemple, le résumé Somme est la somme de tous les points de données affichés dans sa ligne. Les colonnes récapitulatives ne sont pas les mêmes que CloudWatch les statistiques. Représenté dans la source comme suit :

```
 "table": {
        "summaryColumns": [
            "MIN",
            "MAX",
            "SUM",
            "AVG"
        ]
    },
```

Seuils

Utilisez cette option pour appliquer des seuils à votre tableau. Lorsqu'un point de données se situe en deçà d'un seuil, sa cellule est surlignée avec la couleur du seuil. Représenté dans la source comme suit :

```
"annotations": {
    "horizontal": [
        {
            "label": string,
            "value": int,
            "fill": "above" | "below"
        }
    ]
}
```

Unité dans la colonne d’étiquettes

Pour afficher l’unité associée à la métrique, vous pouvez activer cette option pour afficher l’unité dans la colonne d’étiquette à côté de l’étiquette. Représenté dans la source comme suit :

```
"yAxis": {
    "left": {
        "showUnits": true | false
    }
}
```

Inverser les lignes et les colonnes

Cela transforme le tableau de telle sorte que les points de données passent des colonnes aux lignes et que les métriques deviennent des colonnes. Représenté dans la source comme suit :

```
 "table": {
    "layout": "vertical" | "horizontal"
}
```

Colonnes de résumé autocollantes

Cela rend les colonnes de résumé persistantes, de sorte qu’elles restent visibles lorsque vous faites défiler la page. L’étiquette est déjà collante. Représenté dans la source comme suit :

```
"table": {
    "stickySummary": true | false
}
```

Afficher uniquement les colonnes de résumé

Cela empêche l’affichage des colonnes des points de données, de sorte que seules les colonnes d’étiquette et de résumé sont affichées. Représenté dans la source comme suit :

```
 "table": {
    "showTimeSeriesData": false | true
}
```

Données en direct

Affiche le point de données le plus récent, même s’il n’est pas encore complètement agrégé. Représenté dans la source comme suit :

```
"liveData": true | false
```

Format du widget de nombre

Affiche autant de chiffres que peut contenir la cellule, avant d'arrondir et de convertir. Représenté dans la source comme suit :

```
"singleValueFullPrecision": true | false
```

Pour ajouter un widget de tableau de données à 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 Tableaux de bord, puis choisissez un tableau de bord.
3. Cliquez sur le bouton +, sélectionnez Tableau de données, puis cliquez sur Suivant.
4. Dans l'onglet Parcourir, recherchez ou parcourez les métriques que vous souhaitez afficher dans le widget de tableau. Sélectionnez ensuite les métriques.
5. (Facultatif) Pour modifier la disposition du tableau, cliquez sur l'onglet Options et sélectionnez Inverser les lignes et les colonnes.

   Vous pouvez également utiliser l'onglet Options pour modifier les colonnes qui apparaissent dans le tableau et afficher l'unité utilisée dans la colonne Étiquette.

   > **Tip**
   >
   > Pour afficher des seuils plus précis, choisissez Afficher autant de chiffres que possible avant d'arrondir.

6. (Facultatif) Pour modifier l'intervalle de temps de votre widget de tableau de données, sélectionnez l'un des intervalles de temps prédéfinis dans la partie supérieure du widget. L'intervalle de temps varie de 1 heure à 1 semaine. Pour définir votre propre plage de temps, choisissez Custom (Personnalisée).
7. (Facultatif) Pour modifier l'intervalle de temps de votre widget de tableau de données, sélectionnez l'un des intervalles de temps prédéfinis dans la partie supérieure du widget. L'intervalle de temps varie de 1 heure à 1 semaine. Pour définir votre propre plage de temps, choisissez Custom (Personnalisée).

8. (Facultatif) Pour que ce widget continue d'utiliser l'intervalle de temps que vous sélectionnez, même si l'intervalle de temps pour le reste du tableau de bord est modifié ultérieurement, choisissez Conserver l'intervalle de temps.
9. Choisissez Créer un widget, puis choisissez Enregistrer le tableau de bord.

Pour supprimer un widget de tableau d'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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Dans le coin supérieur droit du widget que vous souhaitez supprimer, choisissez Actions du widget, puis choisissez Supprimer.
4. Choisissez Save dashboard (Enregistrer le tableau de bord).

Lier et dissocier des graphiques sur un tableau de bord CloudWatch

Vous pouvez lier tous les graphiques de votre tableau de bord ; ainsi quand vous zoomez en avant ou en arrière sur un graphique, la même action opère simultanément sur les autres graphiques. Vous pouvez supprimer des liens de graphiques pour limiter le zoom à un seul graphique.

Ajouter des liens à des graphiques à 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Choisissez Actions, puis Link graphs (Lier des graphiques).

Pour supprimer les liens entre les graphiques 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 Dashboards (Tableaux de bord), puis choisissez un tableau de bord.
3. Effacez Actions, puis Link graphs (Lier des graphiques).

Partage de CloudWatch tableaux de bord

Vous pouvez partager vos CloudWatch tableaux de bord avec des personnes qui n'ont pas un accès direct à votre AWS compte. Ainsi, vous pouvez partager des tableaux de bord entre les équipes, les parties prenantes et les personnes externes à votre organisation. Vous pouvez même afficher des tableaux de bord sur de grands écrans dans les espaces réservés aux équipes, ou les incorporer dans des Wikis et d'autres pages web.

> **Warning**
>
> Toutes les personnes avec lesquelles vous partagez le tableau de bord bénéficient des autorisations répertoriées dans Autorisations accordées aux personnes avec lesquelles vous partagez le tableau de bord pour le compte. Si vous partagez le tableau de bord publiquement, tous ceux qui ont le lien vers le tableau de bord disposent de ces autorisations. Les `ec2:DescribeTags` autorisations `cloudwatch:GetMetricData` et ne peuvent pas être limitées à des métriques ou à des instances EC2 spécifiques. Les personnes ayant accès au tableau de bord peuvent donc interroger toutes les CloudWatch métriques ainsi que les noms et balises de toutes les instances EC2 du compte.

Lorsque vous partagez des tableaux de bord, vous pouvez désigner qui peut afficher le tableau de bord de trois manières différentes :

- Partagez un tableau de bord unique et désignez jusqu'à cinq adresses e-mail aux personnes autorisées à consulter le tableau de bord. Chacun de ces utilisateurs crée son propre mot de passe qu'il doit saisir afin de pouvoir afficher le tableau de bord.
- Partagez publiquement un tableau de bord unique, afin que tous ceux qui ont le lien puissent consulter le tableau de bord.
- Partagez tous les CloudWatch tableaux de bord de votre compte et spécifiez un fournisseur d'authentification unique (SSO) tiers pour accéder aux tableaux de bord. Tous les utilisateurs membres de la liste de ce fournisseur SSO peuvent accéder à tous les tableaux de bord du compte. Pour ce faire, intégrez le fournisseur SSO à Amazon Cognito. Le fournisseur SSO doit prendre en charge le langage SAML (Security Assertion Markup Language). Pour plus d'informations sur Amazon Cognito, consultez la page Qu'est-ce qu'Amazon Cognito ?.

Le partage d'un tableau de bord n'entraîne pas de frais, mais les widgets d'un tableau de bord partagé entraînent des frais aux tarifs standard CloudWatch . Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Lorsque vous partagez un tableau de bord, les ressources Amazon Cognito sont créées dans la région USA Est (Virginie du Nord).

> **Important**
>
> Ne modifiez pas les noms et les identifiants des ressources créés par le processus de partage de tableau de bord. Celles-ci incluent les ressources Amazon Cognito et IAM. La modification de ces ressources peut entraîner des fonctionnalités non voulues et incorrectes dans les tableaux de bord partagés.

> **Note**
>
> Si vous partagez un tableau de bord contenant des widgets de métriques avec des annotations d'alerte, les personnes avec lesquelles vous partagez ce tableau de bord ne verront pas ces widgets. Ils verront plutôt un widget vide avec du texte indiquant que le widget n'est pas disponible. Vous verrez toujours les widgets de métriques avec des annotations d'alerte lorsque vous visualiserez vous-même le tableau de bord.

Autorisations requises pour partager un tableau de bord

Pour pouvoir partager des tableaux de bord à l'aide de l'une des méthodes suivantes et pour déterminer quels tableaux de bord ont déjà été partagés, vous devez être connecté en tant qu'utilisateur ou avec un rôle IAM disposant de certaines autorisations.

Pour pouvoir partager des tableaux de bord, votre utilisateur ou votre rôle IAM doit inclure les autorisations incluses dans l'instruction de stratégie suivante :

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateRole",
        "iam:CreatePolicy",
        "iam:AttachRolePolicy",
```

```
        "iam:PassRole"
    ],
    "Resource": [
        "arn:aws:iam::*:role/service-role/CWDBSharing*",
        "arn:aws:iam::*:policy/*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*",
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "cloudwatch:GetDashboard",
    ],
    "Resource": [
        "*"
        // or the ARNs of dashboards that you want to share
    ]
}
```

Pour déterminer quels tableaux de bord sont partagés sans pour autant pouvoir les partager, un utilisateur ou un rôle IAM peut inclure une déclaration de stratégie semblable à la suivante :

```
{
    "Effect": "Allow",
    "Action": [
        "cognito-idp:*",
        "cognito-identity:*"
    ],
    "Resource": [
        "*"
    ]
},
{
    "Effect": "Allow",
```

```
    "Action": [
        "cloudwatch:ListDashboards",
    ],
    "Resource": [
        "*"
    ]
}
```

Autorisations accordées aux personnes avec lesquelles vous partagez le tableau de bord

Lorsque vous partagez un tableau de bord, CloudWatch crée un rôle IAM dans le compte qui donne les autorisations suivantes aux personnes avec lesquelles vous partagez le tableau de bord :

- `cloudwatch:GetInsightRuleReport`
- `cloudwatch:GetMetricData`
- `cloudwatch:DescribeAlarms`
- `ec2:DescribeTags`

> **Warning**
>
> Toutes les personnes avec lesquelles vous partagez le tableau de bord bénéficient de ces autorisations pour le compte. Si vous partagez le tableau de bord publiquement, tous ceux qui ont le lien vers le tableau de bord disposent de ces autorisations.
> Les `ec2:DescribeTags` autorisations `cloudwatch:GetMetricData` et ne peuvent pas être limitées à des métriques ou à des instances EC2 spécifiques. Les personnes ayant accès au tableau de bord peuvent donc interroger toutes les CloudWatch métriques ainsi que les noms et balises de toutes les instances EC2 du compte.

Lorsque vous partagez un tableau de bord, par défaut, les autorisations qui le CloudWatch créent limitent l'accès aux seules alarmes et aux règles Contributor Insights présentes sur le tableau de bord lorsqu'il est partagé. Si vous ajoutez de nouvelles alertes ou règles Contributor Insights au tableau de bord et souhaitez qu'elles soient également visibles par les personnes avec lesquelles vous avez partagé le tableau de bord, vous devez mettre à jour la stratégie pour autoriser ces ressources.

Partage d'un tableau de bord unique avec des utilisateurs spécifiques

Suivez les étapes décrites dans cette section pour partager un tableau de bord contenant jusqu'à cinq adresses e-mail de votre choix.

> **Note**
>
> Par défaut, les widgets CloudWatch Logs du tableau de bord ne sont pas visibles par les personnes avec lesquelles vous partagez le tableau de bord. Pour plus d'informations, consultez Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les widgets de table des journaux.
> Par défaut, les widgets d'alerte composites du tableau de bord ne sont pas visibles par les personnes avec lesquelles vous partagez le tableau de bord. Pour de plus amples informations, consultez Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les alertes composites.

Pour partager un tableau de bord unique avec des utilisateurs spécifiques

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 le nom de votre tableau de bord.
4. Choisissez Actions, puis Share dashboard (Partager un tableau de bord).
5. À côté de Share your dashboard and require a username and password (Partager votre tableau de bord et exiger un nom d'utilisateur et un mot de passe), choisissez Start sharing (Commencer le partage).
6. Sous Add email addresses (Ajouter des adresses e-mail), saisissez les adresses e-mail des personnes avec lesquelles vous souhaitez partager le tableau de bord. Vous pouvez inclure jusqu'à cinq adresses e-mail.
7. Lorsque toutes les adresses e-mail sont saisies, lisez l'accord et cochez la case de confirmation. Sélectionnez ensuite Preview policy (Prévisualiser une stratégie).
8. Confirmez que les ressources qui seront partagées correspondent à ce que vous souhaitez, puis sélectionnez Confirm and generate shareable link (Confirmer et générer un lien partageable).
9. Sur la page suivante, sélectionnez Copy link to clipboard (Copier le lien dans le presse-papiers). Vous pouvez ensuite coller ce lien dans un e-mail et l'envoyer aux utilisateurs invités. Ces

derniers reçoivent automatiquement un e-mail séparé avec leur nom d'utilisateur et un mot de passe temporaire à utiliser pour se connecter au tableau de bord.

Partage public d'un tableau de bord unique

Suivez les étapes de cette section pour partager un tableau de bord publiquement. Cela peut être utile pour afficher le tableau de bord sur un grand écran dans un espace réservé aux équipes, ou l'intégrer dans une page Wiki.

> **Important**
>
> En partageant un tableau de bord publiquement, vous le rendez accessible à toute personne en ayant le lien, sans authentification. Effectuez cette opération uniquement pour les tableaux de bord qui ne contiennent pas d'informations sensibles.

> **Note**
>
> Par défaut, les widgets CloudWatch Logs du tableau de bord ne sont pas visibles par les personnes avec lesquelles vous partagez le tableau de bord. Pour plus d'informations, consultez Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les widgets de table des journaux.
> Par défaut, les widgets d'alerte composites du tableau de bord ne sont pas visibles par les personnes avec lesquelles vous partagez le tableau de bord. Pour de plus amples informations, consultez Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les alertes composites.

Pour partager un tableau de bord publiquement

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 le nom de votre tableau de bord.
4. Choisissez Actions, puis Share dashboard (Partager un tableau de bord).
5. À côté de Share your dashboard publicly (Partager votre tableau de bord), choisissez Start sharing (Commencer le partage).

6. Saisissez **`Confirm`** dans la zone de texte.
7. Lisez l'accord et cochez la case de confirmation. Sélectionnez ensuite Preview policy (Prévisualiser une stratégie).
8. Confirmez que les ressources qui seront partagées correspondent à ce que vous souhaitez, puis sélectionnez Confirm and generate shareable link (Confirmer et générer un lien partageable).
9. Sur la page suivante, sélectionnez Copy link to clipboard (Copier le lien dans le presse-papiers). Vous pouvez ensuite partager ce lien. Toute personne avec laquelle vous partagez le lien peut accéder au tableau de bord sans fournir d'informations d'identification.

Partagez tous les CloudWatch tableaux de bord du compte à l'aide de l'authentification unique

Suivez les étapes décrites dans cette section pour partager tous les tableaux de bord de votre compte avec les utilisateurs à l'aide de l'authentification unique (SSO).

> **Note**
>
> Par défaut, les widgets CloudWatch Logs du tableau de bord ne sont pas visibles par les personnes avec lesquelles vous partagez le tableau de bord. Pour plus d'informations, consultez Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les widgets de table des journaux.
> Par défaut, les widgets d'alerte composites du tableau de bord ne sont pas visibles par les personnes avec lesquelles vous partagez le tableau de bord. Pour plus d'informations, consultez Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les alertes composites.

Pour partager vos CloudWatch tableaux de bord avec les utilisateurs figurant dans la liste d'un fournisseur SSO

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 le nom de votre tableau de bord.
4. Choisissez Actions, puis Share dashboard (Partager un tableau de bord).
5. Choisissez Accéder aux CloudWatch paramètres.

6. Si le fournisseur SSO que vous souhaitez n'est pas répertorié dans Available SSO providers (Fournisseurs SSO disponibles), choisissez Manage SSO providers (Gérer les fournisseurs SSO) et suivez les instructions reprises dans Configurer le SSO pour le partage de CloudWatch tableaux de bord.

   Retournez ensuite à la CloudWatch console et actualisez le navigateur. Le fournisseur SSO que vous avez activé doit maintenant apparaître dans la liste.
7. Choisissez le fournisseur SSO de votre choix dans la liste Available SSO providers (Fournisseurs SSO disponibles).
8. Choisissez Save changes (Enregistrer les modifications).

Configurer le SSO pour le partage de CloudWatch tableaux de bord

Pour configurer le partage de tableau de bord via un fournisseur tiers d'authentification unique prenant en charge le langage SAML, procédez comme suit.

> **Important**
>
> Nous vous recommandons fortement de ne pas partager de tableaux de bord à l'aide d'un fournisseur SSO non-SAML. Cela risquerait de permettre par inadvertance à des tiers d'accéder aux tableaux de bord de votre compte.

Pour configurer un fournisseur SSO afin d'activer le partage de tableau de bord

1. Intégrez le fournisseur SSO à Amazon Cognito. Pour de plus amples informations, consultez Intégration de fournisseurs d'identité SAML tiers avec des groupes d'utilisateurs Amazon Cognito.
2. Téléchargez le fichier XML de métadonnées à partir de votre fournisseur SSO.
3. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
4. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
5. Dans la sectionDashboard sharing (Partage de tableau de bord), choisissez Configure (Configurer).
6. Choisissez Manage SSO providers (Gérer les fournisseurs SSO).

Cela a pour effet d'ouvrir la console Amazon Cognito dans la région USA Est (Virginie du Nord) (us-east-1). Si vous ne voyez pas User Pools (Groupes d'utilisateurs), la console Amazon Cognito peut s'être ouverte dans une autre région. Dans ce cas, changez la région en USA Est (Virginie du Nord) us-east-1 et passez aux étapes suivantes.

7. Choisissez le CloudWatchDashboardSharingpool.
8. Dans le panneau de navigation, sélectionnez Identity providers (Fournisseurs d'identité).
9. Choisissez SAML.
10. Saisissez un nom pour votre fournisseur SSO dans Provider name (Nom du fournisseur).
11. Choisissez Select file (Sélectionner un fichier), puis sélectionnez le fichier XML de métadonnées que vous avez téléchargé à l'étape 1.
12. Choisissez Create provider (Créer un fournisseur).

Déterminer combien de vos tableaux de bord sont partagés

Vous pouvez utiliser la CloudWatch console pour voir combien de vos CloudWatch tableaux de bord sont actuellement partagés avec d'autres personnes.

Pour déterminer combien de vos tableaux de bord sont partagés

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
3. La section Dashboard sharing (Partage de tableau) affiche le nombre de tableaux de bord partagés.
4. Pour déterminer quels tableaux de bord sont partagés, choisissez ***number*** dashboards shared (nombre tableaux de bord partagés) sous Username and password (Nom d'utilisateur et mot de passe) et sous Public dashboards (Tableaux de bord publics).

Déterminer lesquels de vos tableaux de bord sont partagés

Vous pouvez utiliser la CloudWatch console pour voir quels tableaux de bord sont actuellement partagés avec d'autres personnes.

Pour déterminer lesquels de vos tableaux de bord sont partagés

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. Dans la liste des tableaux de bord, consultez la colonne Share (Partager). Les tableaux de bord dont l'icône est remplie dans cette colonne sont actuellement partagés.
4. Pour identifier les utilisateurs avec lesquels un tableau de bord est partagé, choisissez le nom du tableau de bord, puis sélectionnez Actions, Share dashboard (Partager un tableau de bord).

   La page Share dashboard ***dashboard name*** (Partager le tableau de bord nom du tableau de bord) affiche le mode de partage du tableau de bord. Si vous le souhaitez, vous pouvez arrêter de partager le tableau de bord en sélectionnant Stop sharing (Arrêter le partage).

Arrêt du partage d'un ou de plusieurs tableaux de bord

Vous pouvez arrêter le partage d'un tableau de bord partagé unique ou arrêter le partage de tous les tableaux de bord partagés à la fois.

Pour arrêter de partager un tableau de bord unique

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 le nom du tableau de bord partagé.
4. Choisissez Actions, puis Share dashboard (Partager un tableau de bord).
5. Choisissez Stop sharing (Arrêter le partage).
6. Dans la boîte de confirmation, sélectionnez Stop sharing (Arrêter le partage).

Pour arrêter le partage de tous les tableaux de bord partagés

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
3. Dans la section Dashboard sharing (Partage de tableau), sélectionnez Stop sharing all dashboards (Arrêter le partage de tous les tableaux de bord).
4. Dans la boîte de confirmation, sélectionnez Stop sharing all dashboards (Arrêter le partage de tous les tableaux de bord).

Vérification des autorisations du tableau de bord et modification de la portée des autorisations

Suivez les étapes de cette section si vous souhaitez vérifier les autorisations des utilisateurs de vos tableaux de bord partagés ou modifier la portée des autorisations des tableaux de bord partagés.

Pour vérifier les autorisations des tableaux de bord partagés

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 le nom du tableau de bord partagé.
4. Choisissez Actions, puis Share dashboard (Partager un tableau de bord).
5. Sous Resources (Ressources), choisissez IAM Role (Rôle IAM).
6. Dans la console IAM, choisissez la stratégie affichée.
7. (Facultatif) Pour limiter les alertes que les utilisateurs du tableau de bord partagé peuvent voir, choisissez Edit policy (Modifier la stratégie) et déplacez l'autorisation `cloudwatch:DescribeAlarms` de sa position actuelle vers une nouvelle instruction `Allow`, qui répertorie les ARN des alertes que vous souhaitez que les utilisateurs du tableau de bord partagé puissent consulter. Consultez l'exemple suivant.

   ```
   {
      "Effect": "Allow",
       "Action": "cloudwatch:DescribeAlarms",
       "Resource": [
           "AlarmARN1",
           "AlarmARN2"
       ]
   }
   ```

   Dans ce cas, veillez à supprimer l'autorisation `cloudwatch:DescribeAlarms` d'une section de la stratégie actuelle, qui ressemble à ceci :

   ```
   {
      "Effect": "Allow",
       "Action": [
           "cloudwatch:GetInsightRuleReport",
           "cloudwatch:GetMetricData",
           "cloudwatch:DescribeAlarms",
   ```

```
        "ec2:DescribeTags"
    ],
    "Resource": "*"
}
```

8. (Facultatif) Pour limiter la portée des règles Contributor Insights que les utilisateurs du tableau de bord partagé peuvent voir, choisissez Edit policy (Modifier la stratégie) et déplacez `cloudwatch:GetInsightRuleReport` de sa position actuelle vers une nouvelle instruction `Allow`, qui répertorie les ARN des règles Contributor Insights que vous souhaitez que les utilisateurs du tableau de bord partagé puissent consulter. Consultez l'exemple suivant.

```
{
   "Effect": "Allow",
    "Action": "cloudwatch:GetInsightRuleReport",
    "Resource": [
        "PublicContributorInsightsRuleARN1",
        "PublicContributorInsightsRuleARN2"
    ]
}
```

Dans ce cas, veillez à supprimer `cloudwatch:GetInsightRuleReport` d'une section de la stratégie actuelle, qui ressemble à ceci :

```
{
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetInsightRuleReport",
                "cloudwatch:GetMetricData",
                "cloudwatch:DescribeAlarms",
                "ec2:DescribeTags"
            ],
            "Resource": "*"
        }
```

Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les alertes composites

Lorsque vous partagez un tableau de bord, les widgets d'alerte composites du tableau de bord ne sont par défaut pas visibles par les personnes avec lesquelles vous partagez le tableau de

bord. Pour que les widgets d'alerte composites soient visibles, vous devez ajouter une autorisation `DescribeAlarms: *` à la stratégie de partage du tableau de bord. Cette autorisation se présente comme suit :

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
}
```

Warning

L'instruction de stratégie précédente donne accès à toutes les alertes du compte. Pour réduire la portée de `cloudwatch:DescribeAlarms`, vous devez utiliser une instruction `Deny`. Vous pouvez ajouter une instruction `Deny` à la stratégie et spécifier les ARN des alertes que vous souhaitez verrouiller. Cette instruction de refus doit ressembler à l'exemple ci-dessous :

```
{
    "Effect": "Allow",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": "*"
},
{
    "Effect": "Deny",
    "Action": "cloudwatch:DescribeAlarms",
    "Resource": [
        "SensitiveAlarm1ARN",
        "SensitiveAlarm1ARN"
    ]
}
```

Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les widgets de table des journaux

Lorsque vous partagez un tableau de bord, les widgets CloudWatch Logs Insights qui se trouvent sur le tableau de bord ne sont pas visibles par les personnes avec lesquelles vous partagez le tableau de

bord. Cela concerne à la fois les widgets CloudWatch Logs Insights qui existent actuellement et ceux qui sont ajoutés au tableau de bord une fois que vous l'avez partagé.

Si vous souhaitez que ces personnes puissent voir les widgets CloudWatch Logs, vous devez ajouter des autorisations au rôle IAM pour le partage du tableau de bord.

Pour permettre aux personnes avec lesquelles vous partagez un tableau de bord de voir les widgets CloudWatch Logs

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 le nom du tableau de bord partagé.
4. Choisissez Actions, puis Share dashboard (Partager un tableau de bord).
5. Sous Resources (Ressources), choisissez IAM Role (Rôle IAM).
6. Dans la console IAM, choisissez la stratégie affichée.
7. Choisissez Edit policy (Modifier la stratégie) et ajoutez l'instruction suivante. Dans la nouvelle instruction, nous vous recommandons de spécifier les ARN des groupes de journaux que vous souhaitez partager uniquement. Consultez l'exemple suivant.

```
{
            "Effect": "Allow",
            "Action": [
                "logs:FilterLogEvents",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:GetLogRecord",
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "SharedLogGroup1ARN",
                "SharedLogGroup2ARN"
           ]
        },
```

8. Choisissez Save Changes (Enregistrer les modifications).

Si votre politique IAM pour le partage de tableau de bord inclut déjà ces cinq autorisations avec * comme ressource, nous vous recommandons fortement de modifier la stratégie et de spécifier

uniquement les ARN des groupes de journaux que vous souhaitez partager. Par exemple, si votre section `Resource` pour ces autorisations était la suivante :

```
"Resource": "*"
```

Modifiez la stratégie pour spécifier uniquement les ARN des groupes de journaux que vous souhaitez partager, comme dans l'exemple suivant :

```
"Resource": [
  "SharedLogGroup1ARN",
  "SharedLogGroup2ARN"
]
```

Autoriser les personnes avec lesquelles vous partagez les tableaux de bord à voir les widgets personnalisés

Lorsque vous partagez un tableau de bord, les widgets personnalisés du tableau de bord ne sont par défaut pas visibles par les personnes avec lesquelles vous partagez le tableau de bord. Cela affecte à la fois les widgets personnalisés qui existent actuellement et ceux qui sont ajoutés au tableau de bord après le partage.

Si vous souhaitez que ces personnes puissent consulter les widgets personnalisés, vous devez ajouter des autorisations au rôle IAM pour le partage de tableau de bord.

Pour permettre aux personnes avec lesquelles vous partagez un tableau de bord de voir les widgets personnalisés

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 le nom du tableau de bord partagé.
4. Choisissez Actions, puis Share dashboard (Partager un tableau de bord).
5. Sous Resources (Ressources), choisissez IAM Role (Rôle IAM).
6. Dans la console IAM, choisissez la stratégie affichée.
7. Choisissez Edit policy (Modifier la stratégie) et ajoutez l'instruction suivante. Dans la nouvelle instruction, nous vous recommandons de spécifier les ARN des fonctions Lambda que vous souhaitez partager uniquement. Consultez l'exemple suivant.

```
{
  "Sid": "Invoke",
  "Effect": "Allow",
  "Action": [
      "lambda:InvokeFunction"
  ],
  "Resource": [
    "LambdaFunction1ARN",
    "LambdaFunction2ARN"
  ]
 }
```

8. Choisissez Save Changes (Enregistrer les modifications).

Si votre stratégie IAM pour le partage de tableau de bord inclut déjà cette autorisation avec * comme ressource, nous vous recommandons fortement de modifier la stratégie et de spécifier uniquement les ARN des fonctions Lambda que vous souhaitez partager. Par exemple, si votre section `Resource` pour ces autorisations était la suivante :

```
"Resource": "*"
```

Modifiez la stratégie pour spécifier uniquement les ARN des widgets personnalisés que vous souhaitez partager, comme dans l'exemple suivant :

```
"Resource": [
  "LambdaFunction1ARN",
  "LambdaFunction2ARN"
]
```

Utilisation des données en direct

Vous pouvez choisir si vos widgets de métriques affichent des données en direct. Les données en direct sont des données publiées au cours de la dernière minute et qui n'ont pas été entièrement agrégées.

- Si les données en direct sont désactivées, seuls les points de données ayant une période d'agrégation d'au moins une minute dans le passé sont affichés. Par exemple, lorsque vous utilisez

des périodes de 5 minutes, le point de données de 12h35 est agrégé de 12h35 à 12h40, et affiché à 12h41.

- Si les données en direct sont activées, le point de données le plus récent est affiché dès que les données sont publiées dans l'intervalle d'agrégation correspondant. Chaque fois que vous actualisez l'affichage, le point de données le plus récent peut changer lorsque de nouvelles données de cette période d'agrégation sont publiées. Si vous utilisez une statistique cumulative telle que Somme ou Nombre d'échantillons, l'utilisation de données en direct peut entraîner une chute à la fin de votre graphique.

Vous pouvez choisir d'activer les données en direct pour un tableau de bord entier ou pour des widgets individuels sur le tableau de bord.

Pour choisir d'utiliser des données en direct sur l'ensemble de votre tableau de bord

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), puis choisissez un tableau de bord.
3. Pour activer ou désactiver définitivement les données en direct pour tous les widgets du tableau de bord, procédez comme suit :

   a. Choisissez Actions, Paramètres, Bulk update live data (Mettre à jour en bloc les données en direct).

   b. Choisissez Live Data on (Données en direct activées) ou Live Data off (Données en direct désactivées), puis choisissez Set (Définir).
4. Pour remplacer temporairement les paramètres des données en direct de chaque widget, choisissez Actions. Ensuite, sous Overrides (Remplacements), à côté de Live data (Données en direct), effectuez l'une des actions suivantes :

   - Choisissez Activé pour activer temporairement les données en direct pour tous les widgets.
   - Choisissez Désactivé pour désactiver temporairement les données en direct pour tous les widgets.
   - Choisissez Do not override (Ne pas remplacer) pour conserver le paramètre de données en direct de chaque widget.

Pour choisir d'utiliser des données en direct sur un seul widget

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), puis choisissez un tableau de bord.
3. Sélectionnez un widget et choisissez Actions, Modifier.
4. Choisissez l'onglet Options de graphique.
5. Activez ou désactivez la case à cocher sous Live Data (Données en direct).

Affichage d'un tableau de bord animé

Vous pouvez afficher un tableau de bord animé qui reproduit les données CloudWatch métriques capturées au fil du temps. Cela peut vous aider à consulter les tendances, à faire des présentations ou à analyser les problèmes après qu'ils se sont produits.

Les widgets animés du tableau de bord incluent des widgets linéaires, des widgets en aires empilées, des widgets avec chiffres et des widgets Explorateur de métriques. Les graphiques circulaires, les graphiques à barres, les widgets de texte et les widgets de journaux sont affichés dans le tableau de bord, mais ne sont pas animés.

Pour afficher un tableau de bord animé

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 le nom du tableau de bord.
4. Choisissez Actions, puis Replay dashboard (Relire le tableau de bord).
5. (Facultatif) Par défaut, lorsque vous démarrez l'animation, celle-ci apparaît sous la forme d'une fenêtre défilante. Si vous souhaitez plutôt que l'animation apparaisse sous forme d' point-by-point animation, choisissez l'icône en forme de loupe pendant que l'animation est en pause et réinitialisez le zoom.
6. Pour démarrer l'animation, cliquez sur le bouton Play (Lecture). Vous pouvez également choisir les boutons Back (Précédent) et Forward (Suivant) pour passer à d'autres points dans le temps.
7. (Facultatif) Pour modifier la fenêtre horaire de l'animation, choisissez le calendrier et sélectionnez la période.

8. Pour modifier la vitesse de l'animation, choisissez Auto speed (Vitesse automatique) et sélectionnez la nouvelle vitesse.
9. Lorsque vous avez terminé, choisissez Exit animate (Quitter l'animation).

Ajouter un CloudWatch tableau de bord à votre liste de favoris

Dans la CloudWatch console, vous pouvez ajouter des tableaux de bord, des alarmes et des groupes de journaux à une liste de favoris. Vous pouvez accéder à la liste de favoris à partir du panneau de navigation. La procédure suivante indique comment ajouter un tableau de bord à la liste de favoris.

Pour ajouter un tableau de bord à la liste de favoris

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. Dans la liste des tableaux de bord, sélectionnez l'étoile en regard du nom du tableau de bord que vous souhaitez ajouter à vos favoris.
   - (Facultatif) Vous pouvez également ajouter un tableau de bord à vos favoris en sélectionnant en le sélectionnant dans la liste et en choisissant l'étoile en regard de son nom.
4. Pour accéder à la liste de favoris, choisissez Favorites and recents (Favoris et récents) dans le panneau de navigation. Le menu contient deux colonnes. La première contient vos tableaux de bord, alertes et groupes de journaux favoris, et la deuxième contient les tableaux de bord, les alertes et les groupes de journaux que vous avez récemment consultés.

> **Tip**
>
> Vous pouvez ajouter des tableaux de bord, des alarmes et des groupes de journaux à vos favoris dans le menu Favoris et récents du volet de navigation de la CloudWatch console. Dans la colonne Recently visited (Récemment consulté), passez la souris sur le tableau de bord que vous souhaitez ajouter à vos favoris et choisissez l'étoile à côté de celui-ci.

Modifier le paramètre de dérogation aux périodes ou l'intervalle d'actualisation du tableau de CloudWatch bord

Vous pouvez spécifier la façon dont le paramètre de période des graphiques ajoutés à ce tableau de bord est conservé ou modifié.

Lorsqu'une période automatique ou une plage de temps persistante est appliquée à un widget, la plage de temps globale du graphique peut affecter les périodes que vous avez définies.

- Si la plage horaire est inférieure ou égale à un jour, les paramètres de période ne sont pas modifiés.
- Si l'intervalle de temps est compris entre un et trois jours, les périodes définies en dessous de cinq minutes sont ramenées à 5 minutes.
- Si la plage horaire est supérieure à trois jours, les périodes définies en dessous d'une heure sont ramenées à une heure.

Les étapes suivantes expliquent comment utiliser la console pour modifier les options de remplacement de période. Vous pouvez également les modifier en utilisant le champ `period0verride` dans la structure JSON du tableau de bord. Pour plus d'informations, veuillez consulter la rubrique Dashboard Body Overall Structure.

Pour modifier les options de remplacement de période

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Actions.
3. Sous Period override (Remplacement de période), choisissez l'un des éléments suivants :
   - Choisissez Auto pour que la période des métriques sur chaque graphique s'adapte automatiquement à la plage de temps du tableau de bord.
   - Choisissez Do not override (Ne pas remplacer) pour vous assurer que le paramètre de période de chaque graphique est toujours respecté.
   - Choisissez l'une des autres options pour que les graphiques ajoutés au tableau de bord adaptent toujours cette période choisie en tant que leur paramètre de période.

L'option Period override (Remplacement de période) redevient toujours Auto lorsque le tableau de bord est fermé ou que le navigateur est actualisé. Il est impossible d'enregistrer différents paramètres pour Period override (Remplacement de période).

Vous pouvez modifier la fréquence à laquelle les données de votre CloudWatch tableau de bord sont actualisées.

Pour modifier l'intervalle d'actualisation du tableau de bord

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), puis choisissez un tableau de bord.
3. Dans le menu Refresh options (en haut à droite), sélectionnez 10 secondes, 1 minute, 2 minutes, 5 minutes ou 15 minutes.

Modifier la plage horaire ou le format du fuseau horaire d'un CloudWatch tableau de bord

Vous pouvez modifier la plage de temps pour afficher des données du tableau de bord sur plusieurs minutes, heures, jours ou semaines. Vous pouvez également modifier le format de fuseau horaire pour afficher des données du tableau de bord en UTC ou en heure locale. L'heure locale est le fuseau horaire spécifié dans le système d'exploitation de votre ordinateur.

> **Note**
>
> Si vous créez un tableau de bord avec des graphiques qui contiennent 100 métriques haute résolution ou plus, nous vous recommandons de ne pas définir une plage de temps supérieure à 1 heure. Pour plus d'informations, consultez Métriques haute résolution.

> **Note**
>
> Si la plage de temps d'un tableau de bord est inférieure à la période utilisée pour un widget du tableau de bord, les événements suivants se produisent :

- Le widget est modifié pour afficher la quantité de données correspondant à une période complète pour ce widget, même si cette période est plus longue que celle du tableau de bord. Cela garantit la présence d'au moins un point de données sur le graphique.
- L'heure de début de la période pour ce point de données est ajustée à rebours pour s'assurer qu'au moins un point de données puisse être affiché.

New console

Pour modifier la plage de temps du tableau de bord

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), puis choisissez un tableau de bord.
3. Depuis l'écran du tableau de bord, effectuez l'une des actions suivantes :
   - Dans la partie supérieure du tableau de bord, sélectionnez l'un des intervalles de temps prédéfinis. Celles-ci durent de 1 heure à 1 semaine (1h, 3h, 12h, 1j ou 1 sem).
   - Vous pouvez également choisir l'une des options de plage de temps personnalisées suivantes :
     - Choisissez Custom (Personnalisée), puis choisissez l'onglet Relative. Choisissez une plage de temps de 1 minute à 15 mois.
     - Choisissez Custom (Personnalisée), puis choisissez l'onglet Absolute (Absolue). Utilisez le calendrier ou les champs de texte pour spécifier votre plage de temps.

> **Tip**
>
> Si la période d'agrégation est définie sur Auto lorsque vous modifiez la plage de temps d'un graphique, CloudWatch cela peut modifier la période. Pour définir la période manuellement, choisissez le menu déroulant Actions, puis choisissez Period override (Remplacement de la période).

Pour modifier le format du fuseau horaire du tableau de bord

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), puis choisissez un tableau de bord.
3. Dans la partie supérieure du tableau de bord, choisissez Personnalisé.

4. Dans le coin supérieur droit de la boîte de dialogue qui s'affiche, choisissez UTC ou Local time (Heure locale).
5. Choisissez Appliquer.

Old console

Pour modifier la plage de temps du tableau de bord

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), puis choisissez un tableau de bord.
3. Depuis l'écran du tableau de bord, effectuez l'une des actions suivantes :
   - Dans la partie supérieure du tableau de bord, sélectionnez l'un des intervalles de temps prédéfinis. Celles-ci durent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j, ou1 sem).
   - Vous pouvez également choisir l'une des options de plage de temps personnalisées suivantes :
     - Cliquez sur l'onglet custom (personnalisée) puis choisissez l'onglet Relative. Sélectionnez l'une des plages prédéfinies, qui s'étendent de 1 minute à 15 mois.
     - Cliquez sur l'onglet custom (personnalisée) puis choisissez l'onglet Absolute (Absolue). Utilisez le calendrier ou les champs de texte pour spécifier votre plage de temps.

> **Tip**
>
> Si la période d'agrégation est définie sur Auto lorsque vous modifiez la plage de temps d'un graphique, CloudWatch cela peut modifier la période. Pour définir la période manuellement, choisissez le menu déroulant Actions, puis choisissez Period override (Remplacement de la période).

Pour modifier le format du fuseau horaire du tableau de bord

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), puis choisissez un tableau de bord.
3. Dans le coin supérieur droit de l'écran du tableau de bord, choisissez le menu déroulant Custom (Personnalisée).
4. Dans le coin supérieur droit de la boîte de dialogue qui s'affiche, choisissez UTC ou Local timezone (Fuseau horaire local) dans le menu déroulant.

Utiliser les CloudWatch métriques Amazon

Les métriques sont des données sur les performances de vos systèmes. Par défaut, de nombreux services offrent des métriques gratuites pour les ressources (telles que des instances Amazon EC2, des volumes Amazon EBS et des instances Amazon RDS DB). Vous pouvez activer la surveillance détaillée de certaines ressources, telles que vos instances Amazon EC2, ou ajouter vos propres métriques d'applications. Amazon CloudWatch peut charger toutes les statistiques de votre compte (à la fois les mesures relatives aux AWS ressources et les mesures relatives aux applications que vous fournissez) à des fins de recherche, de représentation graphique et d'alarmes.

Les données métriques sont conservées pendant 15 mois, ce qui vous permet de consulter à la fois les up-to-the-minute données et les données historiques.

Pour représenter graphiquement les indicateurs dans la console, vous pouvez utiliser CloudWatch Metrics Insights, un moteur de requêtes SQL performant qui vous permet d'identifier les tendances et les modèles au sein de tous vos indicateurs en temps réel.

Table des matières

Surveillance basique et surveillance détaillée

CloudWatch propose deux catégories de surveillance : la surveillance de base et la surveillance détaillée.

De nombreux AWS services offrent une surveillance de base en publiant un ensemble de mesures par défaut sans frais pour les clients. CloudWatch Par défaut, lorsque vous commencez à utiliser l'un d'entre eux Services AWS, la surveillance de base est automatiquement activée. Pour obtenir une liste des services offrant une surveillance basique, veuillez consulter AWS services qui publient CloudWatch des statistiques.

Une surveillance détaillée n'est offerte que par certains services. Elle entraîne également des frais. Pour l'utiliser pour un AWS service, vous devez choisir de l'activer. Pour plus d'informations sur les tarifs, consultez les CloudWatch tarifs Amazon.

Les options de surveillance détaillées diffèrent en fonction des services qui l'offrent. Par exemple, la surveillance détaillée Amazon EC2 fournit des métriques plus fréquentes, publiées à intervalles d'une minute, au lieu d'intervalles de cinq minutes utilisés dans la surveillance basique Amazon EC2. Une surveillance détaillée pour Simple Storage Service (Amazon S3) et Amazon Managed Streaming for Apache Kafka signifie des métriques plus précises.

Dans différents AWS services, la surveillance détaillée porte également des noms différents. Par exemple, dans Amazon EC2, cela s'appelle surveillance détaillée, surveillance améliorée et dans Amazon S3, mesures de demande. AWS Elastic Beanstalk

L'utilisation d'une surveillance détaillée pour Amazon EC2 vous aide à mieux gérer vos ressources Amazon EC2, de sorte que vous puissiez trouver les tendances et agir plus rapidement. Pour Simple Storage Service (Amazon S3), les métriques de demandes sont disponibles à intervalles d'une minute pour vous aider à identifier rapidement les problèmes opérationnels et agir en conséquence. Sur Amazon MSK, lorsque vous activez la surveillance de niveau `PER_BROKER`, `PER_TOPIC_PER_BROKER` ou `PER_TOPIC_PER_PARTITION`, vous obtenez des métriques supplémentaires qui offrent plus de visibilité.

Le tableau suivant répertorie les services offrant une surveillance détaillée. Il inclut également des liens vers la documentation de ces services qui expliquent plus en détail la surveillance détaillée et fournissent des instructions sur la manière de l'activer.

Service	Documentation
Amazon API Gateway	Dimensions pour les métriques API Gateway

Service	Documentation
Amazon CloudFront	Afficher des métriques CloudFront de distribution supplémentaires
Amazon EC2	Activer ou désactiver la surveillance détaillée pour vos instances
Elastic Beanstalk	Surveillance et création de rapports d'état améliorée
Amazon Kinesis Data Streams	Métriques avancées au niveau des partitions
Amazon MSK	Amazon MSK Metrics à surveiller avec CloudWatch
Amazon S3	Mesures de demande Amazon S3 dans CloudWatch

Service	Documentation
Amazon SES	Collectez CloudWatch des métriques de surveillance détaillées à l'aide de la publication d'événements Amazon SES.

En outre, CloudWatch propose des solutions de out-of-the-box surveillance avec des métriques plus détaillées et des tableaux de bord pré-créés pour certains AWS services, comme indiqué dans le tableau suivant.

Service	Documentation des fonctionnalités
Lambda	Informations sur Lambda
Amazon ECS	Informations sur les conteneurs pour Amazon ECS
Amazon EKS	Container Insights pour Amazon EKS et Kubernetes

Interrogez vos indicateurs avec CloudWatch Metrics Insights

CloudWatch Metrics Insights est un puissant moteur de requêtes SQL très performant que vous pouvez utiliser pour interroger vos métriques à grande échelle. Vous pouvez identifier les tendances et les modèles au sein de tous vos CloudWatch indicateurs en temps réel.

Vous pouvez également définir des alarmes sur toutes les requêtes Metrics Insights qui renvoient une seule série temporelle. Cela peut être particulièrement utile pour créer des alarmes qui surveillent des métriques agrégées sur une flotte de votre infrastructure ou de vos applications. Créez l'alarme une fois, et elle s'ajuste dynamiquement au fur et à mesure que des ressources sont ajoutées ou retirées de la flotte.

Vous pouvez effectuer une requête CloudWatch Metrics Insights dans la console à l'aide de l'éditeur de requêtes CloudWatch Metrics Insights. Vous pouvez également effectuer une requête CloudWatch Metrics Insights avec le AWS CLI ou un AWS SDK en exécutant `GetMetricData` ou`PutDashboard`. Les requêtes que vous exécutez avec l'éditeur de requêtes CloudWatch Metrics Insights sont gratuites. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Avec l'éditeur de requêtes CloudWatch Metrics Insights, vous pouvez choisir parmi une variété d'exemples de requêtes prédéfinis et également créer vos propres requêtes. Lorsque vous créez vos requêtes, vous pouvez utiliser une vue du générateur pour parcourir vos statistiques et dimensions existantes. Vous pouvez également utiliser une vue d’éditeur pour écrire des requêtes manuellement.

Vous pouvez également utiliser le langage naturel pour créer des requêtes CloudWatch Metrics 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 Metrics Insights.

Avec Metrics Insights, vous pouvez exécuter des requêtes à grande échelle. En utilisant la clause GROUP BY, vous pouvez regrouper de manière flexible vos métriques en temps réel dans des séries temporelles distinctes par valeur de dimension spécifique, en fonction de vos cas d’utilisation. Comme les requêtes Metrics Insights incluent une fonctionnalité ORDER BY, vous pouvez utiliser Metrics Insights pour effectuer des requêtes de type « Top N ». Par exemple, les requêtes de type « Top N » peuvent analyser des millions de métriques de votre compte et renvoyer les 10 instances qui consomment le plus de processeur. Cela peut vous aider à identifier et à résoudre les problèmes de latence dans vos applications.

Rubriques

Création de vos requêtes

Vous pouvez exécuter une requête CloudWatch Metrics Insights à l'aide de la CloudWatch console AWS CLI, des AWS SDK ou des kits de développement logiciel. Les requêtes exécutées dans la console sont gratuites. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Pour plus d'informations sur l'utilisation AWS des SDK pour effectuer une requête Metrics Insights, consultez GetMetricData.

Pour exécuter une requête à l'aide de la CloudWatch console, procédez comme suit :

Pour interroger vos métriques à l'aide de Metrics Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Choisissez l'onglet Query (Requête).
4. (Facultatif) Pour exécuter un exemple de requête préétablie, choisissez Add query (Ajouter une requête) et sélectionnez la requête à exécuter. Si vous êtes satisfait de cette requête, vous pouvez ignorer le reste de cette procédure. Ou, vous pouvez choisir Editor (Éditeur) pour modifier l'exemple de requête, puis choisir Run (Exécuter) pour exécuter la requête modifiée.

5. Pour créer votre propre requête, vous pouvez utiliser la vue Builder (Générateur), la vue Editor (Éditeur), et également utiliser une combinaison des deux. Vous pouvez passer d'une vue à l'autre à tout moment et voir votre travail en cours dans les deux vues.

   Dans la vue Builder (Générateur), vous pouvez parcourir et sélectionner l'espace de noms des métriques, le nom de la métrique, le filtre, le groupe et les options de commande. Pour chacune de ces options, le générateur de requêtes vous propose une liste de sélections possibles de votre environnement parmi lesquelles choisir.

   Dans la vue Editor (Éditeur), vous pouvez commencer à écrire votre requête. Au fur et à mesure que vous tapez, l'éditeur propose des suggestions basées sur les caractères que vous avez saisis jusqu'à présent.
6. Lorsque vous êtes satisfait de votre requête, sélectionnez Run (Exécuter).
7. (Facultatif) Une autre façon de modifier une requête que vous avez représentée graphiquement consiste à choisir l'onglet Graphed metrics (Graphique des métriques) et choisir l'icône de modification à côté de la formule de requête dans la colonne Details (Détails).
8. (Facultatif) Pour supprimer une requête du graphique, choisissez Graphed metrics (Graphique des métriques) et choisissez l‘icône X à droite de la ligne qui affiche votre requête.

Composants et syntaxe de requête Metrics Insights

CloudWatch La syntaxe de Metrics Insights est la suivante.

```
SELECT FUNCTION(metricName)
FROM namespace | SCHEMA(...)
[ WHERE labelKey OPERATOR labelValue [AND ... ] ]
[ GROUP BY labelKey [ , ... ] ]
[ ORDER BY FUNCTION() [ DESC | ASC ] ]
[ LIMIT number ]
```

Les clauses possibles dans une requête Metrics Insights sont les suivantes. Aucun des mots-clés n'est sensible à la casse, mais les identificateurs tels que les noms des métriques, des espaces de noms et des dimensions sont sensibles à la casse.

SELECT

Obligatoire. Spécifie la fonction à utiliser pour agréger les observations dans chaque compartiment temporel (déterminé par la période fournie). Spécifie également le nom de la métrique à interroger.

Les valeurs valides pour FUNCTION (FONCTION) sont `AVG`, `COUNT`, `MAX`, `MIN` et `SUM`.

- `AVG` calcule la moyenne des observations correspondant à la requête.
- `COUNT` renvoie le nombre d'observations correspondant à la requête.
- `MAX` renvoie la valeur maximale des observations correspondant à la requête.
- `MIN` renvoie la valeur minimale des observations correspondant à la requête.
- `SUM` calcule la somme des observations correspondant à la requête.

FROM

Obligatoire. Spécifie la source de la métrique. Vous pouvez spécifier soit l'espace de noms de métrique qui contient la métrique à interroger, soit une fonction de table SCHEMA (SCHÉMA). Voici des exemples d'espaces de noms de métriques : `"AWS/EC2"`, `"AWS/Lambda"`, et des espaces de noms de métriques que vous avez créés pour vos métriques personnalisées.

Les espaces de noms de métriques qui incluent / ou tout autre caractère qui n'est pas une lettre, un chiffre ou un tiret bas doivent être entourés de guillemets doubles. Pour de plus amples informations, consultez Qu'est-ce qui nécessite des guillemets ou des caractères d'échappement ?.

SCHEMA (SCHÉMA)

Une fonction de table facultative pouvant être utilisée dans une clause FROM (À PARTIR DE). Utilisez SCHEMA (SCHÉMA) pour réduire les résultats de la requête aux métriques qui correspondent parfaitement à une liste de dimensions, ou aux métriques qui n'ont pas de dimensions.

Si vous utilisez une clause SCHEMA (SCHÉMA), elle doit contenir au moins un argument, et ce premier argument doit être l'espace de noms de la métrique interrogée. Si vous spécifiez SCHEMA (SCHÉMA) avec cet argument d'espace de noms uniquement, les résultats sont limités à des métriques qui n'ont aucune dimension.

Si vous spécifiez SCHEMA (SCHÉMA) avec des arguments supplémentaires, les arguments supplémentaires après l'argument d'espace de noms doivent être des clés d'étiquette. Les clés d'étiquette doivent être des noms de dimension. Si vous spécifiez une ou plusieurs de ces clés

d'étiquette, les résultats sont limités uniquement aux métriques qui possèdent cet ensemble exact de dimensions. L'ordre de ces clés d'étiquette n'a pas d'importance.

Par exemple :

- SELECT AVG(CPUUtilization) FROM "AWS/EC2" correspond à toutes les métriques `CPUUtilization` dans l'espace de noms `AWS/EC2`, quelles que soient leurs dimensions, et renvoie une série temporelle agrégée unique.
- SELECT AVG(CPUUtilization) FROM SCHEMA("AWS/EC2") ne correspond qu'aux métriques `CPUUtilization` dans l'espace de noms `AWS/EC2` pour lequel aucune dimension n'est définie.
- SELECT AVG (CPUUtilization) FROM SCHEMA (« AWS/EC2 », InstanceId) ne correspond qu'aux `CPUUtilization` CloudWatch métriques signalées avec exactement une dimension,. `InstanceId`
- SELECT SUM (RequestCount) FROM SCHEMA (« AWS/ApplicationElb » LoadBalancer,, AvailabilityZone) correspond uniquement aux `RequestCount` métriques signalées à CloudWatch from `AWS/ApplicationELB` avec exactement deux dimensions, et. `LoadBalancer` `AvailabilityZone`

WHERE

Facultatif. Filtre les résultats en ne retenant que les métriques qui correspondent à l'expression spécifiée à l'aide de valeurs d'étiquette spécifiques pour une ou plusieurs clés d'étiquette. Par exemple, WHERE InstanceType = 'c3.4xlarge' filtre les résultats uniquement pour les types d'**`c3.4xlarge`**instances, et WHERE ! InstanceType = 'c3.4xlarge' filtre les résultats pour tous les types d'instances sauf. `c3.4xlarge`

Lorsque vous exécutez une requête dans un compte de surveillance, vous pouvez l'utiliser `WHERE AWS.AccountId` pour limiter les résultats au compte que vous spécifiez. Par exemple, `WHERE AWS.AccountId=444455556666` n'interroge les métriques que pour le compte `444455556666`. Pour limiter votre requête aux seuls indicateurs du compte de surveillance lui-même, utilisez `WHERE AWS.AccountId=CURRENT_ACCOUNT_ID()`.

Les valeurs des étiquettes doivent toujours être entourées de guillemets simples.

Opérateurs pris en charge

La clause WHERE (OÙ) prend en charge les opérateurs suivants :

- = La valeur de l'étiquette doit correspondre à la chaîne spécifiée.
- != La valeur de l'étiquette ne doit pas correspondre à la chaîne spécifiée.

- AND (ET) Les deux conditions spécifiées doivent être vraies pour correspondre. Vous pouvez utiliser plusieurs mots-clés AND (ET) afin de spécifier deux conditions ou plus.

GROUP BY

Facultatif. Regroupe les résultats de la requête en plusieurs séries temporelles, chacune correspondant à une valeur différente pour la ou les clés d'étiquette spécifiées. Par exemple, l'utilisation de `GROUP BY InstanceId` renvoie une série temporelle différente pour chaque valeur de `InstanceId`. L'utilisation de `GROUP BY ServiceName, Operation` crée une série temporelle différente pour chaque combinaison possible des valeurs de `ServiceName` et `Operation`.

Avec une clause GROUP BY (REGROUPER PAR), les résultats sont classés par ordre alphabétique croissant par défaut, en utilisant la séquence d'étiquettes spécifiée dans la clause GROUP BY (REGROUPER PAR). Pour modifier l'ordre des résultats, ajoutez une clause ORDER BY (CLASSER PAR) à votre requête.

Lorsque vous exécutez une requête dans un compte de surveillance, vous pouvez l'utiliser `GROUP BY AWS.AccountId` pour regrouper les résultats en fonction des comptes dont ils proviennent.

> **Note**
>
> Si certaines des métriques correspondantes n'incluent pas de clé d'étiquette spécifique spécifiée dans la clause GROUP BY (REGROUPER PAR), un groupe nul nommé `Other` est renvoyé. Par exemple, si vous spécifiez `GROUP BY ServiceName, Operation` et que certaines des métriques renvoyées ne comprennent pas `ServiceName` en tant que dimension, alors ces métriques sont affichées comme ayant `Other` comme valeur pour `ServiceName`.

ORDER BY

Facultatif. Spécifie l'ordre à utiliser pour les séries temporelles renvoyées si la requête renvoie plusieurs séries temporelles. L'ordre est basé sur les valeurs trouvées par la clause FUNCTION (FONCTION) que vous spécifiez dans la clause ORDER BY (CLASSER PAR). La clause FUNCTION (FONCTION) est utilisée pour calculer une valeur scalaire unique à partir de chaque série temporelle renvoyée, et cette valeur est utilisée pour déterminer l'ordre.

Vous spécifiez également s'il faut utiliser l'ordre croissant ASC ou décroissant DESC. Si vous l'omettez, la valeur par défaut est croissante, ASC.

Par exemple, l'ajout d'une clause `ORDER BY MAX() DESC` classe les résultats en fonction du point de données maximal observé dans la plage temporelle, dans l'ordre décroissant. En d'autres mots, la série temporelle ayant le point de données maximal le plus élevé est renvoyée en premier.

Les fonctions valides à utiliser dans une clause ORDER BY (CLASSER PAR) sont `AVG()`, `COUNT()`, `MAX()`, `MIN()` et `SUM()`.

Si vous utilisez une clause ORDER BY (CLASSER PAR) avec une clause LIMIT (LIMITE), la requête résultante est une requête « Top N ». ORDER BY (CLASSER PAR) est également utile pour les requêtes qui peuvent renvoyer un grand nombre de métriques, car chaque requête ne peut pas renvoyer plus de 500 séries temporelles. Si une requête correspond à plus de 500 séries temporelles et que vous utilisez une clause ORDER BY (CLASSER PAR), les séries temporelles sont triées, puis les 500 séries temporelles qui arrivent en premier dans l'ordre de tri sont celles qui sont renvoyées.

LIMIT

Facultatif. Limite le nombre de séries temporelles renvoyées par la requête à la valeur que vous spécifiez. La valeur maximale que vous pouvez spécifier est 500, et une requête qui ne spécifie pas de LIMIT (LIMITE) ne peut pas non plus renvoyer plus de 500 séries temporelles.

L'utilisation d'une clause LIMIT (LIMITE) avec une clause ORDER BY (CLASSER PAR) vous donne une requête « Top N ».

Qu'est-ce qui nécessite des guillemets ou des caractères d'échappement ?

Dans une requête, les valeurs d'étiquette doivent toujours être entourées de guillemets simples. Par exemple, SELECT MAX (CPUUtilization) FROM « AWS/EC2" WHERE = « ». AutoScalingGroupName my-production-fleet

Les espaces de noms de métriques, les noms de métriques et les clés d'étiquette contenant des caractères autres que des lettres, des chiffres et des tirets bas (_) doivent être entourés de guillemets doubles. Par exemple, SELECT MAX("My.Metric").

Si l'un d'entre eux contient un guillemet double ou un guillemet simple lui-même (par exemple : `Bytes"Input"`), vous devez séparer chaque guillemet par une barre oblique inverse, comme dans SELECT AVG("Bytes\"Input\"").

Si un espace de noms de métrique, un nom de métrique ou une clé d'étiquette contient un mot qui est un mot-clé réservé dans Metrics Insights, ceux-ci doivent également être entre guillemets doubles. Par exemple, si vous disposez d'une métrique nommée `LIMIT`, vous devez utiliser `SELECT AVG("LIMIT")`. Il est également valable de placer n'importe quel espace de noms, nom de métrique ou étiquette entre guillemets doubles, même s'il n'inclut pas de mot-clé réservé.

Pour obtenir la liste complète des mots-clés réservés, consultez Mots-clés réservés.

Créer une requête riche étape par étape

Cette section illustre la création d'un exemple complet qui utilise toutes les clauses possibles, étape par étape.

Nous démarrons par la requête suivante, qui rassemble l'ensemble des métriques `RequestCount` de l'Application Load Balancer qui sont collectées avec les deux dimensions `LoadBalancer` et `AvailabilityZone`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
```

Désormais, si nous voulons voir les métriques uniquement à partir d'un équilibreur de charge spécifique, nous pouvons ajouter une clause WHERE (OÙ) pour limiter les métriques renvoyées en ne retenant que les métriques pour lesquelles la valeur de la dimension `LoadBalancer` est `app/load-balancer-1`.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
```

La requête précédente agrège les métriques `RequestCount` de toutes les zones de disponibilité de cet équilibreur de charge en une seule série temporelle. Si nous voulons voir différentes séries temporelles pour chaque zone de disponibilité, nous pouvons ajouter une clause GROUP BY (REGROUPER PAR).

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
```

Ensuite, nous pourrions vouloir classer ces résultats pour voir d'abord les valeurs les plus élevées. La clause ORDER BY (CLASSER PAR) suivante classe les séries temporelles par ordre décroissant, en fonction de la valeur maximale signalée par chaque série temporelle pendant la période de requête :

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
```

Enfin, si nous sommes avant tout intéressés par un type de requête « Top N », nous pouvons utiliser une clause LIMIT (LIMITE). Cet exemple final limite les résultats aux séries temporelles avec les cinq valeurs `MAX` les plus élevées.

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer, AvailabilityZone)
WHERE LoadBalancer = 'app/load-balancer-1'
GROUP BY AvailabilityZone
ORDER BY MAX() DESC
LIMIT 5
```

Exemple de requête entre comptes

Ces exemples sont valides lorsqu'ils sont exécutés dans un compte configuré en tant que compte de surveillance dans le cadre de l' CloudWatch observabilité entre comptes.

L’exemple suivant montre comment rechercher toutes les instances Amazon EC2 du compte source 123456789012 et en afficher la moyenne.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId ='123456789012'
```

L’exemple suivant interroge la métrique `CPUUtilization` dans `AWS/EC2` pour tous les comptes source liés et regroupe les résultats par ID de compte et type d’instance.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
GROUP BY AWS.AccountId, InstanceType
```

L'exemple suivant interroge le `CPUUtilization` dans le compte de surveillance lui-même.

```
SELECT AVG(CpuUtilization)
FROM "AWS/EC2"
WHERE AWS.AccountId = CURRENT_ACCOUNT_ID()
```

Mots-clés réservés

Les mots clés suivants sont réservés dans CloudWatch Metrics Insights. Si l'un de ces mots se trouve dans un espace de noms, un nom de métrique ou une clé d'étiquette dans une requête, vous devez les placer entre guillemets doubles. Les mots-clés réservés ne respectent pas la casse.

```
"ABORT" "ABORTSESSION" "ABS" "ABSOLUTE" "ACCESS" "ACCESSIBLE" "ACCESS_LOCK" "ACCOUNT"
 "ACOS" "ACOSH" "ACTION" "ADD" "ADD_MONTHS"
"ADMIN" "AFTER" "AGGREGATE" "ALIAS" "ALL" "ALLOCATE" "ALLOW" "ALTER" "ALTERAND" "AMP"
 "ANALYSE" "ANALYZE" "AND" "ANSIDATE" "ANY" "ARE" "ARRAY",
"ARRAY_AGG" "ARRAY_EXISTS" "ARRAY_MAX_CARDINALITY" "AS" "ASC" "ASENSITIVE" "ASIN"
 "ASINH" "ASSERTION" "ASSOCIATE" "ASUTIME" "ASYMMETRIC" "AT",
"ATAN" "ATAN2" "ATANH" "ATOMIC" "AUDIT" "AUTHORIZATION" "AUX" "AUXILIARY" "AVE"
 "AVERAGE" "AVG" "BACKUP" "BEFORE" "BEGIN" "BEGIN_FRAME" "BEGIN_PARTITION",
"BETWEEN" "BIGINT" "BINARY" "BIT" "BLOB" "BOOLEAN" "BOTH" "BREADTH" "BREAK" "BROWSE"
 "BT" "BUFFERPOOL" "BULK" "BUT" "BY" "BYTE" "BYTEINT" "BYTES" "CALL",
"CALLED" "CAPTURE" "CARDINALITY" "CASCADE" "CASCADED" "CASE" "CASESPECIFIC" "CASE_N"
 "CAST" "CATALOG" "CCSID" "CD" "CEIL" "CEILING" "CHANGE" "CHAR",
"CHAR2HEXINT" "CHARACTER" "CHARACTERS" "CHARACTER_LENGTH" "CHARS" "CHAR_LENGTH" "CHECK"
 "CHECKPOINT" "CLASS" "CLASSIFIER" "CLOB" "CLONE" "CLOSE" "CLUSTER",
"CLUSTERED" "CM" "COALESCE" "COLLATE" "COLLATION" "COLLECT" "COLLECTION" "COLLID"
 "COLUMN" "COLUMN_VALUE" "COMMENT" "COMMIT" "COMPLETION" "COMPRESS" "COMPUTE",
"CONCAT" "CONCURRENTLY" "CONDITION" "CONNECT" "CONNECTION" "CONSTRAINT" "CONSTRAINTS"
 "CONSTRUCTOR" "CONTAINS" "CONTAINSTABLE" "CONTENT" "CONTINUE" "CONVERT",
"CONVERT_TABLE_HEADER" "COPY" "CORR" "CORRESPONDING" "COS" "COSH" "COUNT" "COVAR_POP"
 "COVAR_SAMP" "CREATE" "CROSS" "CS" "CSUM" "CT" "CUBE" "CUME_DIST",
"CURRENT" "CURRENT_CATALOG" "CURRENT_DATE" "CURRENT_DEFAULT_TRANSFORM_GROUP"
 "CURRENT_LC_CTYPE" "CURRENT_PATH" "CURRENT_ROLE" "CURRENT_ROW" "CURRENT_SCHEMA",
"CURRENT_SERVER" "CURRENT_TIME" "CURRENT_TIMESTAMP" "CURRENT_TIMEZONE"
 "CURRENT_TRANSFORM_GROUP_FOR_TYPE" "CURRENT_USER" "CURRVAL" "CURSOR" "CV" "CYCLE"
 "DATA",
"DATABASE" "DATABASES" "DATABLOCKSIZE" "DATE" "DATEFORM" "DAY" "DAYS" "DAY_HOUR"
 "DAY_MICROSECOND" "DAY_MINUTE" "DAY_SECOND" "DBCC" "DBINFO" "DEALLOCATE" "DEC",
"DECFLOAT" "DECIMAL" "DECLARE" "DEFAULT" "DEFERRABLE" "DEFERRED" "DEFINE" "DEGREES"
 "DEL" "DELAYED" "DELETE" "DENSE_RANK" "DENY" "DEPTH" "DEREF" "DESC" "DESCRIBE",
"DESCRIPTOR" "DESTROY" "DESTRUCTOR" "DETERMINISTIC" "DIAGNOSTIC" "DIAGNOSTICS"
 "DICTIONARY" "DISABLE" "DISABLED" "DISALLOW" "DISCONNECT" "DISK" "DISTINCT",
```

```
"DISTINCTROW" "DISTRIBUTED" "DIV" "DO" "DOCUMENT" "DOMAIN" "DOUBLE" "DROP" "DSSIZE"
 "DUAL" "DUMP" "DYNAMIC" "EACH" "ECHO" "EDITPROC" "ELEMENT" "ELSE" "ELSEIF",
"EMPTY" "ENABLED" "ENCLOSED" "ENCODING" "ENCRYPTION" "END" "END-EXEC" "ENDING"
 "END_FRAME" "END_PARTITION" "EQ" "EQUALS" "ERASE" "ERRLVL" "ERROR" "ERRORFILES",
"ERRORTABLES" "ESCAPE" "ESCAPED" "ET" "EVERY" "EXCEPT" "EXCEPTION" "EXCLUSIVE" "EXEC"
 "EXECUTE" "EXISTS" "EXIT" "EXP" "EXPLAIN" "EXTERNAL" "EXTRACT" "FALLBACK
"FALSE" "FASTEXPORT" "FENCED" "FETCH" "FIELDPROC" "FILE" "FILLFACTOR" "FILTER" "FINAL"
 "FIRST" "FIRST_VALUE" "FLOAT" "FLOAT4" "FLOAT8" "FLOOR"
"FOR" "FORCE" "FOREIGN" "FORMAT" "FOUND" "FRAME_ROW" "FREE" "FREESPACE" "FREETEXT"
 "FREETEXTTABLE" "FREEZE" "FROM" "FULL" "FULLTEXT" "FUNCTION"
"FUSION" "GE" "GENERAL" "GENERATED" "GET" "GIVE" "GLOBAL" "GO" "GOTO" "GRANT" "GRAPHIC"
 "GROUP" "GROUPING" "GROUPS" "GT" "HANDLER" "HASH"
"HASHAMP" "HASHBAKAMP" "HASHBUCKET" "HASHROW" "HAVING" "HELP" "HIGH_PRIORITY" "HOLD"
 "HOLDLOCK" "HOUR" "HOURS" "HOUR_MICROSECOND" "HOUR_MINUTE"
"HOUR_SECOND" "IDENTIFIED" "IDENTITY" "IDENTITYCOL" "IDENTITY_INSERT" "IF" "IGNORE"
 "ILIKE" "IMMEDIATE" "IN" "INCLUSIVE" "INCONSISTENT" "INCREMENT"
"INDEX" "INDICATOR" "INFILE" "INHERIT" "INITIAL" "INITIALIZE" "INITIALLY" "INITIATE"
 "INNER" "INOUT" "INPUT" "INS" "INSENSITIVE" "INSERT" "INSTEAD"
"INT" "INT1" "INT2" "INT3" "INT4" "INT8" "INTEGER" "INTEGERDATE" "INTERSECT"
 "INTERSECTION" "INTERVAL" "INTO" "IO_AFTER_GTIDS" "IO_BEFORE_GTIDS"
"IS" "ISNULL" "ISOBID" "ISOLATION" "ITERATE" "JAR" "JOIN" "JOURNAL" "JSON_ARRAY"
 "JSON_ARRAYAGG" "JSON_EXISTS" "JSON_OBJECT" "JSON_OBJECTAGG"
"JSON_QUERY" "JSON_TABLE" "JSON_TABLE_PRIMITIVE" "JSON_VALUE" "KEEP" "KEY" "KEYS"
 "KILL" "KURTOSIS" "LABEL" "LAG" "LANGUAGE" "LARGE" "LAST"
"LAST_VALUE" "LATERAL" "LC_CTYPE" "LE" "LEAD" "LEADING" "LEAVE" "LEFT" "LESS" "LEVEL"
 "LIKE" "LIKE_REGEX" "LIMIT" "LINEAR" "LINENO" "LINES"
"LISTAGG" "LN" "LOAD" "LOADING" "LOCAL" "LOCALE" "LOCALTIME" "LOCALTIMESTAMP" "LOCATOR"
 "LOCATORS" "LOCK" "LOCKING" "LOCKMAX" "LOCKSIZE" "LOG"
"LOG10" "LOGGING" "LOGON" "LONG" "LONGBLOB" "LONGTEXT" "LOOP" "LOWER" "LOW_PRIORITY"
 "LT" "MACRO" "MAINTAINED" "MAP" "MASTER_BIND"
"MASTER_SSL_VERIFY_SERVER_CERT" "MATCH" "MATCHES" "MATCH_NUMBER" "MATCH_RECOGNIZE"
 "MATERIALIZED" "MAVG" "MAX" "MAXEXTENTS" "MAXIMUM" "MAXVALUE"
"MCHARACTERS" "MDIFF" "MEDIUMBLOB" "MEDIUMINT" "MEDIUMTEXT" "MEMBER" "MERGE" "METHOD"
 "MICROSECOND" "MICROSECONDS" "MIDDLEINT" "MIN" "MINDEX"
"MINIMUM" "MINUS" "MINUTE" "MINUTES" "MINUTE_MICROSECOND" "MINUTE_SECOND" "MLINREG"
 "MLOAD" "MLSLABEL" "MOD" "MODE" "MODIFIES" "MODIFY"
"MODULE" "MONITOR" "MONRESOURCE" "MONSESSION" "MONTH" "MONTHS" "MSUBSTR" "MSUM"
 "MULTISET" "NAMED" "NAMES" "NATIONAL" "NATURAL" "NCHAR" "NCLOB"
"NE" "NESTED_TABLE_ID" "NEW" "NEW_TABLE" "NEXT" "NEXTVAL" "NO" "NOAUDIT" "NOCHECK"
 "NOCOMPRESS" "NONCLUSTERED" "NONE" "NORMALIZE" "NOT" "NOTNULL"
"NOWAIT" "NO_WRITE_TO_BINLOG" "NTH_VALUE" "NTILE" "NULL" "NULLIF" "NULLIFZERO" "NULLS"
 "NUMBER" "NUMERIC" "NUMPARTS" "OBID" "OBJECT" "OBJECTS"
"OCCURRENCES_REGEX" "OCTET_LENGTH" "OF" "OFF" "OFFLINE" "OFFSET" "OFFSETS" "OLD"
 "OLD_TABLE" "OMIT" "ON" "ONE" "ONLINE" "ONLY" "OPEN" "OPENDATASOURCE"
```

```
"OPENQUERY" "OPENROWSET" "OPENXML" "OPERATION" "OPTIMIZATION" "OPTIMIZE"
 "OPTIMIZER_COSTS" "OPTION" "OPTIONALLY" "OR" "ORDER" "ORDINALITY" "ORGANIZATION"
"OUT" "OUTER" "OUTFILE" "OUTPUT" "OVER" "OVERLAPS" "OVERLAY" "OVERRIDE" "PACKAGE" "PAD"
 "PADDED" "PARAMETER" "PARAMETERS" "PART" "PARTIAL" "PARTITION"
"PARTITIONED" "PARTITIONING" "PASSWORD" "PATH" "PATTERN" "PCTFREE" "PER" "PERCENT"
 "PERCENTILE" "PERCENTILE_CONT" "PERCENTILE_DISC" "PERCENT_RANK" "PERIOD" "PERM"
"PERMANENT" "PIECESIZE" "PIVOT" "PLACING" "PLAN" "PORTION" "POSITION" "POSITION_REGEX"
 "POSTFIX" "POWER" "PRECEDES" "PRECISION" "PREFIX" "PREORDER"
"PREPARE" "PRESERVE" "PREVVAL" "PRIMARY" "PRINT" "PRIOR" "PRIQTY" "PRIVATE"
 "PRIVILEGES" "PROC" "PROCEDURE" "PROFILE" "PROGRAM" "PROPORTIONAL"
"PROTECTION" "PSID" "PTF" "PUBLIC" "PURGE" "QUALIFIED" "QUALIFY" "QUANTILE" "QUERY"
 "QUERYNO" "RADIANS" "RAISERROR" "RANDOM" "RANGE" "RANGE_N" "RANK"
"RAW" "READ" "READS" "READTEXT" "READ_WRITE" "REAL" "RECONFIGURE" "RECURSIVE" "REF"
 "REFERENCES" "REFERENCING" "REFRESH" "REGEXP" "REGR_AVGX" "REGR_AVGY"
"REGR_COUNT" "REGR_INTERCEPT" "REGR_R2" "REGR_SLOPE" "REGR_SXX" "REGR_SXY" "REGR_SYY"
 "RELATIVE" "RELEASE" "RENAME" "REPEAT" "REPLACE" "REPLICATION"
"REPOVERRIDE" "REQUEST" "REQUIRE" "RESIGNAL" "RESOURCE" "RESTART" "RESTORE" "RESTRICT"
 "RESULT" "RESULT_SET_LOCATOR" "RESUME" "RET" "RETRIEVE" "RETURN"
"RETURNING" "RETURNS" "REVALIDATE" "REVERT" "REVOKE" "RIGHT" "RIGHTS" "RLIKE" "ROLE"
 "ROLLBACK" "ROLLFORWARD" "ROLLUP" "ROUND_CEILING" "ROUND_DOWN"
"ROUND_FLOOR" "ROUND_HALF_DOWN" "ROUND_HALF_EVEN" "ROUND_HALF_UP" "ROUND_UP" "ROUTINE"
 "ROW" "ROWCOUNT" "ROWGUIDCOL" "ROWID" "ROWNUM" "ROWS" "ROWSET"
"ROW_NUMBER" "RULE" "RUN" "RUNNING" "SAMPLE" "SAMPLEID" "SAVE" "SAVEPOINT" "SCHEMA"
 "SCHEMAS" "SCOPE" "SCRATCHPAD" "SCROLL" "SEARCH" "SECOND" "SECONDS"
"SECOND_MICROSECOND" "SECQTY" "SECTION" "SECURITY" "SECURITYAUDIT" "SEEK" "SEL"
 "SELECT" "SEMANTICKEYPHRASETABLE" "SEMANTICSIMILARITYDETAILSTABLE"
"SEMANTICSIMILARITYTABLE" "SENSITIVE" "SEPARATOR" "SEQUENCE" "SESSION" "SESSION_USER"
 "SET" "SETRESRATE" "SETS" "SETSESSRATE" "SETUSER" "SHARE" "SHOW"
"SHUTDOWN" "SIGNAL" "SIMILAR" "SIMPLE" "SIN" "SINH" "SIZE" "SKEW" "SKIP" "SMALLINT"
 "SOME" "SOUNDEX" "SOURCE" "SPACE" "SPATIAL" "SPECIFIC" "SPECIFICTYPE"
"SPOOL" "SQL" "SQLEXCEPTION" "SQLSTATE" "SQLTEXT" "SQLWARNING" "SQL_BIG_RESULT"
 "SQL_CALC_FOUND_ROWS" "SQL_SMALL_RESULT" "SQRT" "SS" "SSL" "STANDARD"
"START" "STARTING" "STARTUP" "STAT" "STATE" "STATEMENT" "STATIC" "STATISTICS" "STAY"
 "STDDEV_POP" "STDDEV_SAMP" "STEPINFO" "STOGROUP" "STORED" "STORES"
"STRAIGHT_JOIN" "STRING_CS" "STRUCTURE" "STYLE" "SUBMULTISET" "SUBSCRIBER" "SUBSET"
 "SUBSTR" "SUBSTRING" "SUBSTRING_REGEX" "SUCCEEDS" "SUCCESSFUL"
"SUM" "SUMMARY" "SUSPEND" "SYMMETRIC" "SYNONYM" "SYSDATE" "SYSTEM" "SYSTEM_TIME"
 "SYSTEM_USER" "SYSTIMESTAMP" "TABLE" "TABLESAMPLE" "TABLESPACE" "TAN"
"TANH" "TBL_CS" "TEMPORARY" "TERMINATE" "TERMINATED" "TEXTSIZE" "THAN" "THEN"
 "THRESHOLD" "TIME" "TIMESTAMP" "TIMEZONE_HOUR" "TIMEZONE_MINUTE" "TINYBLOB"
"TINYINT" "TINYTEXT" "TITLE" "TO" "TOP" "TRACE" "TRAILING" "TRAN" "TRANSACTION"
 "TRANSLATE" "TRANSLATE_CHK" "TRANSLATE_REGEX" "TRANSLATION" "TREAT"
"TRIGGER" "TRIM" "TRIM_ARRAY" "TRUE" "TRUNCATE" "TRY_CONVERT" "TSEQUAL" "TYPE" "UC"
 "UESCAPE" "UID" "UNDEFINED" "UNDER" "UNDO" "UNION" "UNIQUE"
```

```
"UNKNOWN" "UNLOCK" "UNNEST" "UNPIVOT" "UNSIGNED" "UNTIL" "UPD" "UPDATE" "UPDATETEXT"
 "UPPER" "UPPERCASE" "USAGE" "USE" "USER" "USING" "UTC_DATE"
"UTC_TIME" "UTC_TIMESTAMP" "VALIDATE" "VALIDPROC" "VALUE" "VALUES" "VALUE_OF"
 "VARBINARY" "VARBYTE" "VARCHAR" "VARCHAR2" "VARCHARACTER" "VARGRAPHIC"
"VARIABLE" "VARIADIC" "VARIANT" "VARYING" "VAR_POP" "VAR_SAMP" "VCAT" "VERBOSE"
 "VERSIONING" "VIEW" "VIRTUAL" "VOLATILE" "VOLUMES" "WAIT" "WAITFOR"
"WHEN" "WHENEVER" "WHERE" "WHILE" "WIDTH_BUCKET" "WINDOW" "WITH" "WITHIN"
 "WITHIN_GROUP" "WITHOUT" "WLM" "WORK" "WRITE" "WRITETEXT" "XMLCAST" "XMLEXISTS"
"XMLNAMESPACES" "XOR" "YEAR" "YEARS" "YEAR_MONTH" "ZEROFILL" "ZEROIFNULL" "ZONE"
```

Création d'alarmes sur les requêtes Metrics Insights

Vous pouvez créer des alarmes sur des requêtes Metrics Insights. Cela vous permet d'avoir des alarmes qui suivent plusieurs ressources sans avoir besoin d'être mises à jour ultérieurement. La requête capte les nouvelles ressources et les ressources qui changent. Par exemple, vous pouvez créer une alerte qui surveille l'utilisation du CPU de votre flotte, et l'alerte évalue automatiquement les nouvelles instances que vous lancez après avoir créé l'alerte.

Dans un compte de surveillance configuré pour l'observabilité CloudWatch entre comptes, vos alarmes Metrics Insights peuvent surveiller les ressources des comptes sources et du compte de surveillance lui-même. Pour plus d'informations sur la façon de limiter vos requêtes d'alarme à un compte spécifique ou de regrouper les résultats par identifiant de compte, veuillez consulter les sections `WHERE` et `GROUP BY` dans Composants et syntaxe de requête Metrics Insights.

Table des matières

Création d'une alarme Metrics Insights

Pour créer une alarme sur une requête Metrics Insights à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Choisissez l'onglet Query (Requête).
4. (Facultatif) Pour exécuter un exemple de requête préétablie, choisissez Add query (Ajouter une requête) et sélectionnez la requête à exécuter. Ou, vous pouvez choisir Editor (Éditeur) pour modifier l'exemple de requête, puis choisir Run (Exécuter) pour exécuter la requête modifiée.

5. Pour créer votre propre requête, vous pouvez utiliser la vue Builder (Générateur), la vue Editor (Éditeur), ou une combinaison des deux. Vous pouvez passer d'une vue à l'autre à tout moment et voir votre travail en cours dans les deux vues.

   Dans la vue Builder (Générateur), vous pouvez parcourir et sélectionner l'espace de noms des métriques, le nom de la métrique, le filtre, le groupe et les options de commande. Pour chacune de ces options, le générateur de requêtes vous propose une liste de sélections possibles de votre environnement parmi lesquelles choisir.

   Dans la vue Editor (Éditeur), vous pouvez commencer à écrire votre requête. Au fur et à mesure que vous tapez, l'éditeur propose des suggestions basées sur les caractères que vous avez saisis jusqu'à présent.

   > **Important**
   >
   > Pour créer une alarme sur une requête Metrics Insights, la requête doit retourner une seule série temporelle. Si elle contient une instruction GROUP BY, celle-ci doit être contenue dans une expression mathématique métrique qui ne renvoie qu'une seule série temporelle comme résultat final de l'expression.

6. Lorsque vous êtes satisfait de votre requête, sélectionnez Run (Exécuter).
7. Sélectionnez Créer une alerte.
8. Sous Conditions, spécifiez les éléments suivants :

   a. Pour Whenever ***metric*** is (À chaque fois que la métrique est), spécifiez si la métrique doit être supérieure à, inférieure à ou égale au seuil. Dans than... (à...), spécifiez la valeur de seuil.

   b. 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 pour la première valeur un nombre inférieur à celui de la seconde valeur. Pour plus d'informations, consultez . Évaluation d'une alerte.

   c. 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, consultez . Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.

9. Choisissez Next (Suivant).
10. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alerte passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Pour que l'alerte envoie plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, choisissez Add notification (Ajouter une notification).

    Pour que l'alerte n'envoie pas de notifications, choisissez Remove (Supprimer).
11. Pour que l'alerte exécute Auto Scaling, EC2 ou des actions du Systems Manager, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer. Les alertes peuvent effectuer des actions du Systems Manager uniquement lorsqu'elles passent à l'état ALARM. Pour plus d'informations sur les actions de Systems Manager, consultez les sections Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

    > **Note**
    >
    > Pour créer une alerte qui exécute une action SSM Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager.
12. Lorsque vous avez terminé, choisissez Next (Suivant).
13. Saisissez un nom et une description pour l'alerte. Le nom ne doit contenir que des caractères ASCII. Sélectionnez ensuite Next (Suivant).
14. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

Pour créer une alarme sur une requête Metrics Insights à l'aide du AWS CLI

- Utilisez la commande `put-metric-alarm` et spécifiez une requête Metrics Insights dans le paramètre `metrics`. Par exemple, la commande suivante définit une alarme qui passe à l'état ALARM si l'une de vos instances dépasse 50 % d'utilisation du CPU.

```
aws cloudwatch put-metric-alarm --alarm-name Metrics-Insights-alarm --
evaluation-periods 1 --comparison-operator GreaterThanThreshold --metrics
```

```
'[{"Id":"m1","Expression":"SELECT MAX(CPUUtilization) FROM SCHEMA(\"AWS/EC2\",
InstanceId)", "Period":60}]' --threshold 50
```

Cas de données partielles

Si la requête Metrics Insights utilisée pour l'alarme correspond à plus de 10 000 métriques, l'alarme est évaluée sur la base des 10 000 premières métriques trouvées par la requête. Cela signifie que l'alarme est évaluée sur des données partielles.

Vous pouvez utiliser les méthodes suivantes pour savoir si une alarme Metrics Insights est en train d'évaluer son état d'alarme sur la base de données partielles :

- Dans la console, si vous choisissez une alarme pour voir la page Details (Détails), le message Evaluation warning: Not evaluating all data (Avertissement d'évaluation : toutes les données ne sont pas évaluées) apparaît sur cette page.
- La valeur s'affiche `PARTIAL_DATA` dans le `EvaluationState` champ lorsque vous utilisez la AWS CLI commande describe-alarm ou l' DescribeAlarmsAPI.

Les alarmes publient également des événements sur Amazon EventBridge lorsqu'elles passent à l'état de données partielles. Vous pouvez donc créer une EventBridge règle pour surveiller ces événements. Dans ces cas, le champ `evaluationState` possède la valeur `PARTIAL_DATA`. Voici un exemple.

```
{
    "version": "0",
    "id": "12345678-3bf9-6a09-dc46-12345EXAMPLE",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-11-08T11:26:05Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:my-alarm-name"
    ],
    "detail": {
        "alarmName": "my-alarm-name",
        "state": {
            "value": "ALARM",
```

```
            "reason": "Threshold Crossed: 3 out of the last 3 datapoints [20000.0
 (08/11/22 11:25:00), 20000.0 (08/11/22 11:24:00), 20000.0 (08/11/22 11:23:00)] were
 greater than the threshold (0.0) (minimum 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2022-11-08T11:26:05.399+0000\",\"startDate\":\"2022-11-08T11:23:00.000+0000\",
\"period\":60,\"recentDatapoints\":[20000.0,20000.0,20000.0],\"threshold\":0.0,
\"evaluatedDatapoints\":[{\"timestamp\":\"2022-11-08T11:25:00.000+0000\",\"value
\":20000.0}]}",
            "timestamp": "2022-11-08T11:26:05.401+0000",
            "evaluationState": "PARTIAL_DATA"
        },
        "previousState": {
            "value": "INSUFFICIENT_DATA",
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2022-11-08T11:25:51.227+0000"
        },
        "configuration": {
            "metrics": [
                {
                    "id": "m2",
                    "expression": "SELECT SUM(PartialDataTestMetric) FROM
 partial_data_test",
                    "returnData": true,
                    "period": 60
                }
            ]
        }
    }
}
```

Si la requête pour l'alarme comprend une instruction GROUP BY qui renvoie initialement plus de 500 séries temporelles, l'alarme est évaluée sur la base des 500 premières séries temporelles que la requête trouve. Cependant, si vous utilisez une clause ORDER BY, toutes les séries temporelles trouvées par la requête sont triées, et les 500 qui ont les valeurs les plus élevées ou les plus basses selon votre clause ORDER BY sont utilisées pour évaluer l'alarme.

Utiliser des requêtes Metrics Insights avec des mathématiques appliquées aux métriques

Vous pouvez utiliser une requête Metrics Insights en entrée dans une fonction mathématique de métrique. Pour plus d'informations sur les mathématiques appliquées aux métriques, consultez Utilisation des mathématiques appliquées aux métriques.

Une requête Metrics Insights qui n'inclut pas de clause GROUP BY (REGROUPER PAR) renvoie une seule série temporelle. Par conséquent, les résultats renvoyés peuvent être utilisés avec n'importe quelle fonction mathématique de métrique qui prend une seule série temporelle en entrée.

Une requête Metrics Insights qui inclut une clause GROUP BY (REGROUPER PAR) renvoie plusieurs séries temporelles. Par conséquent, les résultats renvoyés peuvent être utilisés avec n'importe quelle fonction mathématique de métrique qui prend en entrée un tableau de séries temporelles.

Par exemple, la requête suivante renvoie le nombre total d'octets téléchargés pour chaque compartiment de la région, sous forme de tableau de séries temporelles :

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
```

Sur un graphique de la console ou lors d'une GetMetricDataopération, les résultats de cette requête sontq1. Cette requête renvoie le résultat en octets. Par conséquent, si vous souhaitez voir le résultat en Mo, vous pouvez utiliser la fonction mathématique suivante :

```
q1/1024/1024
```

Utiliser le langage naturel pour générer et mettre à jour CloudWatch les requêtes Metrics Insights

> Cette fonctionnalité est en version préliminaire aux États-Unis Est (Virginie du Nord), à l'Ouest des États-Unis (Oregon) et en Asie-Pacifique (Tokyo) CloudWatch et est sujette à modification.

CloudWatch prend en charge une fonctionnalité de requête en langage naturel pour vous aider à générer et à mettre à jour des requêtes pour CloudWatch Metrics Insights et CloudWatch Logs Insights.

Grâce à cette fonctionnalité, vous pouvez poser des questions ou décrire les CloudWatch données que vous recherchez dans un langage simple. 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 « Which Amazon Elastic Compute Cloud instance has the highest network out? » et « Show me the top 10 Amazon DynamoDB Tables by consumed reads ».

Pour générer une requête CloudWatch Metrics Insights avec cette fonctionnalité, ouvrez l'éditeur de requêtes CloudWatch Metrics Insights dans la vue du générateur ou de l'éditeur et choisissez Generate query.

> **Important**
>
> Pour utiliser la fonctionnalité de requête en langage naturel, vous devez utiliser la CloudWatchFullAccessReadOnlyAccesspolitique, AdministratorAccess, CloudWatchFullAccessV2 ou. CloudWatchReadOnlyAccess
> Vous pouvez également inclure l'action `cloudwatch:GenerateQuery` dans une politique en ligne ou gérée par le client, qu'elle soit nouvelle ou existante.

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 CloudWatch Metrics Insights, voir Composants et syntaxe des requêtes Metrics Insights.

Exemple : 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 une requête qui effectue une recherche de base.

Invite

Voici un exemple d'invite qui indique à la fonctionnalité de rechercher les 10 tables DynamoDB les plus gourmandes en capacité de lecture.

```
Show top 10 DynamoDB Tables by consumed reads
```

Requête

Voici un exemple de requête générée 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.

```
# Show top 10 DynamoDB Tables by consumed reads
SELECT SUM("ConsumedReadCapacityUnits")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query selects the sum of consumed read capacity units for each DynamoDB table,
 groups the results by table name, orders the results from highest to lowest read
 capacity consumption, and limits the results to the top 10 tables.
```

> **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.

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 principales tables DynamoDB consommant le plus de capacité de lecture, cette invite indique désormais la possibilité de trier les résultats en fonction du nombre d'octets renvoyés.

```
Sort by bytes returned instead
```

Requête mise à jour

Voici un exemple de la requête mise à jour. 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.

```
# Sort by bytes returned instead
SELECT SUM("ReturnedBytes")
FROM "AWS/DynamoDB"
GROUP BY TableName
ORDER BY SUM() DESC
LIMIT 10
# This query modifies the original query to select the sum of returned bytes instead
 of consumed read capacity units, and orders the results from highest to lowest sum of
 returned bytes, limiting the results to the top 10 tables.
```

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 Metrics 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 l' CloudWatch IA, y compris la capacité de génération de requêtes, vous devez créer une politique de désinscription pour CloudWatch. Pour plus d'informations, veuillez consulter la rubrique Politiques de désactivation des services IA dans le Guide de l'utilisateur AWS Organizations .

Inférence SQL

CloudWatch Metrics Insights utilise plusieurs mécanismes pour déduire l'intention d'une requête SQL donnée.

Rubriques

- Segmentation temporelle
- Projection des champs
- Agrégation globale ORDER BY (CLASSER PAR)

Segmentation temporelle

Les données en séries chronologiques résultant d'une requête sont regroupées dans des compartiments de temps en fonction de la période demandée. Pour agréger des valeurs dans SQL standard, une clause GROUP BY (REGROUPER PAR) explicite doit être définie pour collecter toutes les observations d'une période donnée. Comme il s'agit de la méthode standard pour interroger des données de séries chronologiques, CloudWatch Metrics Insights en déduit le découpage temporel sans qu'il soit nécessaire d'exprimer une clause GROUP BY explicite.

Par exemple, lorsqu'une requête est exécutée avec une période d'une minute, toutes les observations appartenant à cette minute jusqu'à la prochaine (exclue) sont cumulées jusqu'à l'heure de démarrage du compartiment temporel. Cela rend les instructions SQL de Metrics Insights plus concises et moins détaillées.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

La requête précédente renvoie une seule série temporelle (paires horodatage-valeurs), représentant l'utilisation moyenne de CPU de toutes les instances Amazon EC2. En supposant que la période demandée soit d'une minute, chaque point de données renvoyé représente la moyenne de toutes les observations mesurées dans un intervalle spécifique d'une minute (heure de démarrage incluse, heure de fin exclue). L'horodatage lié au point de données spécifique est l'heure de démarrage du compartiment

Projection des champs

Les requêtes Metrics Insights renvoient toujours la projection de l'horodatage. Vous n'avez pas besoin de spécifier de colonne d'horodatage dans la clause SELECT (SÉLECTIONNER) pour obtenir l'horodatage de chaque valeur de point de données correspondante. Pour plus d'informations sur le calcul de l'horodatage, consultez Segmentation temporelle.

Lorsque vous utilisez GROUP BY (REGROUPER PAR), chaque nom de groupe est également déduit et projeté dans le résultat, de sorte que vous pouvez regrouper la série temporelle renvoyée.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
```

La requête précédente renvoie une série temporelle pour chaque instance EC2 d'Amazon. Chaque série temporelle est étiquetée après la valeur de l'ID d'instance.

Agrégation globale ORDER BY (CLASSER PAR)

Lorsque vous utilisez ORDER BY (CLASSER PAR), FUNCTION() (FONCTION ()) détermine la fonction d'agrégation que vous souhaitez classer (les valeurs des points de données des métriques interrogées). L'opération d'agrégation est effectuée sur tous les points de données correspondants de chaque série temporelle individuelle dans la fenêtre temporelle interrogée.

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX()
LIMIT 10
```

La requête précédente renvoie l'utilisation de CPU pour chaque instance Amazon EC2, ce qui limite le jeu de résultats à 10 entrées. Les résultats sont classés en fonction de la valeur maximale des séries temporelles individuelles dans la fenêtre de temps interrogée. La clause ORDER BY (CLASSER PAR) est appliquée avant LIMIT (LIMITE), de sorte que la commande est calculée sur plus de 10 séries temporelles.

Exemples de requêtes Metrics Insights

Cette section contient des exemples de requêtes CloudWatch Metrics Insights utiles que vous pouvez copier et utiliser directement ou copier et modifier dans l'éditeur de requêtes. Certains de ces exemples sont déjà disponibles dans la console et vous pouvez y accéder en choisissant Add query (Ajouter une requête) dans la vue Metrics (Métriques).

Application Load Balancer

Nombre total de requêtes pour tous les équilibreurs de charge

```
SELECT SUM(RequestCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
```

Top 10 des équilibreurs de charge les plus actifs

```
SELECT MAX(ActiveConnectionCount)
FROM SCHEMA("AWS/ApplicationELB", LoadBalancer)
GROUP BY LoadBalancer
ORDER BY SUM() DESC
LIMIT 10
```

AWS Exemples d'utilisation de l'API

Les 20 meilleures AWS API selon le nombre d'appels sur votre compte

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API'
GROUP BY Service, Resource
ORDER BY COUNT() DESC
LIMIT 20
```

CloudWatch API triées par appels

```
SELECT COUNT(CallCount)
FROM SCHEMA("AWS/Usage", Class, Resource, Service, Type)
WHERE Type = 'API' AND Service = 'CloudWatch'
GROUP BY Resource
ORDER BY COUNT() DESC
```

Exemples DynamoDB

Top 10 des tables par lectures consommées

```
SELECT SUM(ProvisionedWriteCapacityUnits)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 des tables par octets renvoyés

```
SELECT SUM(ReturnedBytes)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 des tables par erreurs utilisateur

```
SELECT SUM(UserErrors)
FROM SCHEMA("AWS/DynamoDB", TableName)
GROUP BY TableName
ORDER BY MAX() DESC LIMIT 10
```

Exemples Amazon Elastic Block Store

Top 10 des volumes Amazon EBS par octets écrits

```
SELECT SUM(VolumeWriteBytes)
FROM SCHEMA("AWS/EBS", VolumeId)
GROUP BY VolumeId
ORDER BY SUM() DESC
LIMIT 10
```

Temps d'écriture moyen du volume Amazon EBS

```
SELECT AVG(VolumeTotalWriteTime)
FROM SCHEMA("AWS/EBS", VolumeId)
```

Exemples Amazon EC2

Utilisation de CPU des instances EC2 triées par niveau le plus élevé

```
SELECT AVG(CPUUtilization)
  FROM SCHEMA("AWS/EC2", InstanceId)
  GROUP BY InstanceId
  ORDER BY AVG() DESC
```

Utilisation moyenne de CPU sur l'ensemble de la flotte

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
```

Top 10 des instances selon l'utilisation de CPU la plus élevée

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/EC2", InstanceId)
GROUP BY InstanceId
ORDER BY MAX() DESC
LIMIT 10
```

Dans ce cas, l' CloudWatch agent collecte une **`CPUUtilization`** métrique par application. Cette requête filtre la moyenne de cette métrique pour un nom d'application spécifique.

```
SELECT AVG(CPUUtilization)
FROM "AWS/CWAgent"
WHERE ApplicationName = 'eCommerce'
```

Exemples Amazon Elastic Container Service

Utilisation moyenne de CPU sur tous les clusters ECS

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
```

Top 10 des clusters par utilisation de la mémoire

```
SELECT AVG(MemoryUtilization)
FROM SCHEMA("AWS/ECS", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC
LIMIT 10
```

Top 10 des services par utilisation de CPU

```
SELECT AVG(CPUUtilization)
FROM SCHEMA("AWS/ECS", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Top 10 des services par exécution des tâches (Container Insights)

```
SELECT AVG(RunningTaskCount)
FROM SCHEMA("ECS/ContainerInsights", ClusterName, ServiceName)
GROUP BY ClusterName, ServiceName
ORDER BY AVG() DESC
LIMIT 10
```

Exemples de conteneur Insights Amazon Elastic Kubernetes Service

Utilisation moyenne de CPU dans tous les clusters EKS

```
SELECT AVG(pod_cpu_utilization)
```

```
FROM SCHEMA("ContainerInsights", ClusterName)
```

Top 10 des clusters par utilisation de CPU des nœuds

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Top 10 des clusters par utilisation de la mémoire des pods

```
SELECT AVG(pop_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName)
GROUP BY ClusterName
ORDER BY AVG() DESC LIMIT 10
```

Top 10 des nœuds par utilisation de CPU

```
SELECT AVG(node_cpu_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, NodeName)
GROUP BY ClusterName, NodeName
ORDER BY AVG() DESC LIMIT 10
```

Top 10 des pods par utilisation de la mémoire

```
SELECT AVG(pod_memory_utilization)
FROM SCHEMA("ContainerInsights", ClusterName, PodName)
GROUP BY ClusterName, PodName
ORDER BY AVG() DESC LIMIT 10
```

EventBridge exemples

Top 10 des règles par invocations

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 des règles par échec des invocations

```
SELECT SUM(FailedInvocations)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Top 10 des règles par règles correspondantes

```
SELECT SUM(MatchedEvents)
FROM SCHEMA("AWS/Events", RuleName)
GROUP BY RuleName
ORDER BY MAX() DESC LIMIT 10
```

Exemples Kinesis

Top 10 des flux par octets écrits

```
SELECT SUM("PutRecords.Bytes")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY SUM() DESC LIMIT 10
```

Top 10 des flux par les premiers éléments du flux

```
SELECT MAX("GetRecords.IteratorAgeMilliseconds")
FROM SCHEMA("AWS/Kinesis", StreamName)
GROUP BY StreamName
ORDER BY MAX() DESC LIMIT 10
```

Exemples Lambda

Fonctions Lambda classées en fonction du nombre d'invocations

```
SELECT SUM(Invocations)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
```

Top 10 des fonctions Lambda par exécution la plus longue

```
SELECT AVG(Duration)
```

```
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY MAX() DESC
LIMIT 10
```

Top 10 des fonctions Lambda par nombre d'erreurs

```
SELECT SUM(Errors)
FROM SCHEMA("AWS/Lambda", FunctionName)
GROUP BY FunctionName
ORDER BY SUM() DESC
LIMIT 10
```

CloudWatch Exemples de journaux

Top 10 des groupes de journaux par événements entrants

```
SELECT SUM(IncomingLogEvents)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Top 10 des groupes de journaux par octets écrits

```
SELECT SUM(IncomingBytes)
FROM SCHEMA("AWS/Logs", LogGroupName)
GROUP BY LogGroupName
ORDER BY SUM() DESC LIMIT 10
```

Exemples Amazon RDS

Top 10 des instances Amazon RDS par utilisation de CPU la plus élevée

```
SELECT MAX(CPUUtilization)
FROM SCHEMA("AWS/RDS", DBInstanceIdentifier)
GROUP BY DBInstanceIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Top 10 des clusters Amazon RDS par écritures

```
SELECT SUM(WriteIOPS)
FROM SCHEMA("AWS/RDS", DBClusterIdentifier)
GROUP BY DBClusterIdentifier
ORDER BY MAX() DESC
LIMIT 10
```

Exemples Amazon Simple Storage Service

Latence moyenne par compartiment

```
SELECT AVG(TotalRequestLatency)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY AVG() DESC
```

Top 10 des compartiments par octets téléchargés

```
SELECT SUM(BytesDownloaded)
FROM SCHEMA("AWS/S3", BucketName, FilterId)
WHERE FilterId = 'EntireBucket'
GROUP BY BucketName
ORDER BY SUM() DESC
LIMIT 10
```

Exemples Amazon Simple Notification Service

Nombre total de messages publiés par les rubriques SNS

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
```

Top 10 des rubriques par messages publiés

```
SELECT SUM(NumberOfMessagesPublished)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Top 10 des rubriques par échecs de livraison des messages

```
SELECT SUM(NumberOfNotificationsFailed)
FROM SCHEMA("AWS/SNS", TopicName)
GROUP BY TopicName
ORDER BY SUM() DESC
LIMIT 10
```

Exemples Amazon SQS

Les 10 files d'attente les plus fréquentes selon le nombre de messages visibles

```
SELECT AVG(ApproximateNumberOfMessagesVisible)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Top 10 des files d'attente les plus actives

```
SELECT SUM(NumberOfMessagesSent)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY SUM() DESC
LIMIT 10
```

Top 10 des files d'attente par âge du premier message

```
SELECT AVG(ApproximateAgeOfOldestMessage)
FROM SCHEMA("AWS/SQS", QueueName)
GROUP BY QueueName
ORDER BY AVG() DESC
LIMIT 10
```

Limites Metrics Insights

CloudWatch Les limites de Metrics Insights sont actuellement les suivantes :

- Actuellement, vous ne pouvez interroger que les trois dernières heures de données.
- Une seule requête ne peut traiter plus de 10 000 métriques. Cela signifie que si les clauses SELECT (SÉLECTIONNER), FROM (À PARTIR DE), et WHERE (OÙ) correspondent à plus de 10 000 métriques, la requête ne traite que les 10 000 premières métriques trouvées.

- Une seule requête ne peut pas renvoyer plus de 500 séries temporelles. Cela signifie que si la requête renvoie plus de 500 métriques, toutes les métriques ne seront pas renvoyées dans les résultats de la requête. Si vous utilisez une clause ORDER BY (CLASSER PAR), alors toutes les métriques en cours de traitement sont triées, et les 500 qui ont les valeurs les plus élevées ou les plus basses selon votre clause ORDER BY (CLASSER PAR) sont renvoyées.

  Si vous n'incluez pas de clause ORDER BY (CLASSER PAR), vous ne pouvez pas contrôler les 500 métriques correspondantes qui sont renvoyées.
- Vous pouvez avoir jusqu'à 200 alarmes Metrics Insights par région.
- Metrics Insights ne prend pas en charge les données haute résolution, c'est-à-dire les données de métriques rapportées avec une granularité inférieure à une minute. Si vous demandez des données haute résolution, la demande n'échoue pas, mais la sortie est agrégée à une granularité d'une minute.
- Chaque GetMetricDataopération ne peut comporter qu'une seule requête, mais vous pouvez avoir plusieurs widgets dans un tableau de bord qui incluent chacun une requête.

Glossaire de Metrics Insights

étiquette

Dans Metrics Insights, une étiquette est une paire valeur-clé utilisée pour étendre une requête afin de renvoyer un ensemble de données particulier, ou pour définir des critères selon lesquels les résultats de la requête doivent être séparés en séries temporelles distinctes. Une clé d'étiquette est similaire à un nom de colonne dans SQL. Actuellement, les étiquettes doivent être de dimensions CloudWatch métriques.

observation

Une observation est une valeur enregistrée pour une métrique donnée à un moment donné.

Dépannage Metrics Insights

Les résultats incluent « Autre », mais je n'ai pas cela comme dimension

Cela signifie que la requête inclut une clause GROUP BY (REGROUPER PAR) qui spécifie une clé d'étiquette qui n'est pas utilisée dans certaines des métriques qui sont renvoyées par la requête. Dans ce cas, un groupe nul nommé `Other` est renvoyé. Les métriques qui n'incluent pas cette clé

d'étiquette sont probablement des métriques agrégées qui renvoient des valeurs agrégées sur toutes les valeurs de cette clé d'étiquette.

Par exemple, supposons que nous ayons la requête suivante :

```
SELECT AVG(Faults)
FROM MyCustomNamespace
GROUP BY Operation, ServiceName
```

Si certaines des métriques renvoyées ne comprennent pas `ServiceName` en tant que dimension, alors ces métriques sont affichées comme ayant `Other` comme valeur pour `ServiceName`.

Pour éviter de voir « Autre » dans vos résultats, utilisez SCHEMA (SCHÉMA) dans votre clause FROM (À PARTIR DE), comme dans l'exemple suivant :

```
SELECT AVG(Faults)
FROM SCHEMA(MyCustomNamespace, Operation)
GROUP BY Operation, ServiceName
```

Cela limite les résultats renvoyés aux métriques qui ont à la fois les dimensions `Operation` et `ServiceName`.

L'horodatage le plus ancien de mon graphique a une valeur de métrique inférieure à celle des autres

CloudWatch Metrics Insights ne prend actuellement en charge que les trois dernières heures de données. Lorsque vous effectuez un graphique avec une période supérieure à une minute, il peut arriver que le point de données le plus ancien diffère de la valeur attendue. Cela est dû au fait que les requêtes Metrics Insights ne renvoient que les trois dernières heures de données. Dans ce cas, le point de données le plus ancien de la requête renvoie uniquement les observations qui ont été mesurées au cours des trois dernières heures, au lieu de renvoyer toutes les observations de la période correspondant au point de données.

Utilisez l'explorateur de métriques pour contrôler les ressources en fonction de leurs étiquettes et de leurs propriétés

L'explorateur de métriques est un outil basé sur des balises qui vous permet de filtrer, d'agréger et de visualiser vos métriques par balises et propriétés de ressources, afin d'améliorer l'observabilité de vos services. Cela vous offre une expérience de dépannage flexible et dynamique, de sorte que vous

pouvez créer plusieurs graphiques à la fois et utiliser ces graphiques pour créer vos tableaux de bord d'état des applications.

Les visualisations de l'explorateur de mesures sont dynamiques. Ainsi, si une ressource correspondante est créée après avoir créé un widget d'explorateur de mesures et que vous l'avez ajouté à un CloudWatch tableau de bord, la nouvelle ressource apparaît automatiquement dans le widget d'explorateur.

Par exemple, si toutes vos instances de production EC2 ont l'identification **`production`**, vous pouvez utiliser l'explorateur de métriques pour filtrer et agréger les métriques de toutes ces instances afin de comprendre leur état et leurs performances. Si une nouvelle instance avec une identification correspondante est créée ultérieurement, elle est automatiquement ajoutée au widget de l'explorateur de métriques.

> **Note**
>
> L'explorateur de métriques fournit une point-in-time expérience. Les ressources qui ont été résiliées ou qui n'existent plus avec la propriété ou la balise que vous avez spécifié ne sont pas affichées dans la visualisation. Cependant, vous pouvez toujours trouver les métriques de ces ressources dans CloudWatch les vues des métriques.

Avec l'explorateur de métriques, vous pouvez choisir comment agréger les métriques à partir des ressources qui correspondent aux critères et si elles doivent toutes être affichées dans un seul graphique ou sur différents graphiques au sein d'un widget d'explorateur de métriques.

L'explorateur de métriques inclut des modèles que vous pouvez utiliser pour afficher des graphiques de visualisation utiles en un seul clic. Vous pouvez également étendre ces modèles pour créer des widgets d'explorateur de métriques entièrement personnalisés.

L'explorateur de métriques prend en charge les métriques émises par AWS et les métriques EC2 publiées par l' CloudWatch agent, notamment les métriques de mémoire, de disque et de processeur. Pour utiliser l'explorateur de métriques afin de voir les métriques publiées par l' CloudWatch agent, vous devrez peut-être mettre à jour le fichier de configuration de CloudWatch l'agent. Pour de plus amples informations, veuillez consulter CloudWatch configuration de l'agent pour l'explorateur de métriques

Pour créer une visualisation avec l'explorateur de métriques et éventuellement l'ajouter à un tableau de bord, procédez comme suit.

Pour créer une visualisation à l'aide de l'explorateur de métriques

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Explorer (Explorateur).
3. Effectuez l'une des actions suivantes :
   - Pour utiliser un modèle, sélectionnez-le dans la zone qui affiche actuellementEmpty Explorer (Explorateur vide).

     Selon le modèle, l'explorateur peut immédiatement afficher des graphiques de métriques. Si ce n'est pas le cas, choisissez une ou plusieurs identifications ou propriétés dans la liste From (De), puis les données doivent apparaître. Si ce n'est pas le cas, utilisez les options en haut de la page pour afficher une plage de temps plus longue dans les graphiques.
   - Pour créer une visualisation personnalisée, sous Metrics (Métriques), choisissez une seule métrique ou toutes les métriques disponibles à partir d'un service.

     Une fois que vous avez choisi une métrique, vous pouvez éventuellement répéter cette étape pour ajouter d'autres métriques.
4. Pour chaque métrique sélectionnée, CloudWatch affiche la statistique qu'elle utilisera immédiatement après le nom de la métrique. Pour la modifier, sélectionnez le nom de la statistique, puis sélectionnez la statistique de votre choix.
5. Sous From (De), choisissez une identification ou une propriété de ressource pour filtrer vos résultats.

   Si vous le souhaitez, vous pouvez ensuite répéter cette étape pour choisir d'autres balises ou propriétés de ressource.

   Si vous choisissez plusieurs valeurs de la même propriété, telles que deux types d'instances EC2, l'explorateur affiche toutes les ressources correspondant à l'une ou l'autre des propriétés choisies. L'opération est alors considérée comme une opération OR.

   Si vous choisissez des propriétés ou des identifications différentes, telles que l'identification **`Production`** et le type d'instance M5, seules les ressources correspondant à toutes ces sélections sont affichées. L'opération est alors considérée comme une opération AND.
6. (Facultatif) Pour Aggregate by (Regrouper par), choisissez une statistique à utiliser pour regrouper les métriques. Ensuite, à côté de for (Pour), choisissez le mode de regroupement de la métrique de la liste. Vous pouvez regrouper toutes les ressources actuellement affichées ou les regrouper par une seule balise ou propriété de ressource.

Selon la méthode de regroupement que vous sélectionnez, le résultat peut être donner lieu à une seule série temporelle ou à plusieurs séries temporelles.

7. Sous Split by (Fractionner par), vous pouvez choisir de fractionner un seul graphique avec plusieurs séries temporelles en plusieurs graphiques. Le fractionnement peut être réalisé selon une variété de critères, que vous choisissez sous l'option Split by (Fractionner par).
8. Sous Graph options (Options de graphique), vous pouvez affiner le graphique en modifiant la période, le type de graphique, le placement de la légende et la mise en page.
9. Pour ajouter cette visualisation sous forme de widget à un CloudWatch tableau de bord, choisissez Ajouter au tableau de bord.

CloudWatch configuration de l'agent pour l'explorateur de métriques

Pour permettre à l'explorateur de mesures de découvrir les métriques EC2 publiées par l' CloudWatch agent, assurez-vous que le fichier de configuration de l' CloudWatch agent contient les valeurs suivantes :

- Dans la section `metrics`, assurez-vous que le paramètre `aggregation_dimensions` inclut `["InstanceId"]`. Il peut également contenir d'autres dimensions.
- Dans la section `metrics`, assurez-vous que le paramètre `append_dimensions` inclut une ligne `{"InstanceId":"${aws:InstanceId}"}`. Il peut également contenir d'autres lignes.
- Dans la section `metrics`, à l'intérieur de la section `metrics_collected`, vérifiez les sections de chaque type de ressource que l'explorateur de métriques doit découvrir, telles que les sections `cpu`, `disk` et `memory`. Assurez-vous que chacune de ces sections possède `"resources": [ "*"] line.`.
- Dans la section `cpu` de la section `metrics_collected`, assurez-vous qu'une ligne `"totalcpu": true` est présente.
- Vous devez utiliser l'espace de `CWAgent` noms par défaut pour les métriques collectées par l' CloudWatch agent, au lieu d'un espace de noms personnalisé.

Les paramètres de la liste précédente obligent l' CloudWatch agent à publier des métriques agrégées pour les disques, les processeurs et les autres ressources qui peuvent être tracées dans l'explorateur de métriques pour toutes les instances qui l'utilisent.

Ces paramètres republient les métriques que vous aviez précédemment configurées pour être publiées avec plusieurs dimensions, ce qui ajoute à vos coûts de métriques.

Pour plus d'informations sur la modification du fichier de configuration de l' CloudWatch agent, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

Utiliser les flux de métriques

Vous pouvez utiliser les flux métriques pour diffuser en continu CloudWatch les métriques vers la destination de votre choix, avec une near-real-time diffusion et une faible latence. Les destinations prises en charge incluent des AWS destinations telles qu'Amazon Simple Storage Service et plusieurs destinations de fournisseurs de services tiers.

Il existe trois principaux scénarios d'utilisation des flux CloudWatch métriques :

- Configuration personnalisée avec Firehose : créez un flux de métriques et dirigez-le vers un flux de diffusion Amazon Data Firehose qui transmet vos CloudWatch métriques là où vous le souhaitez. Vous pouvez les diffuser vers un lac de données tel qu'Amazon S3, ou vers n'importe quelle destination ou point de terminaison pris en charge par Firehose, y compris des fournisseurs tiers. Les formats JSON, OpenTelemetry 1.0.0 et OpenTelemetry 0.7.0 sont pris en charge de manière native, ou vous pouvez configurer des transformations dans votre flux de diffusion Firehose pour convertir les données dans un autre format tel que Parquet. Avec un flux de mesures, vous pouvez continuellement mettre à jour les données de surveillance ou combiner ces données CloudWatch métriques avec des données de facturation et de performance pour créer des ensembles de données riches. Vous pouvez ensuite utiliser des outils tels qu'Amazon Athena pour obtenir des informations sur l'optimisation des coûts, les performances des ressources et l'utilisation des ressources.
- Configuration rapide de S3 : diffusez vers Amazon Simple Storage Service grâce à un processus de configuration rapide. Par défaut, CloudWatch crée les ressources nécessaires au flux. Les formats JSON, OpenTelemetry 1.0.0 et OpenTelemetry 0.7.0 sont pris en charge.
- Configuration rapide des AWS partenaires : CloudWatch fournit une expérience de configuration rapide à certains partenaires tiers. Vous pouvez faire appel à des fournisseurs de services tiers pour surveiller, dépanner et analyser vos applications à l'aide des données diffusées CloudWatch en continu. Lorsque vous utilisez le flux de travail de configuration rapide des partenaires, vous devez uniquement fournir une URL de destination et une clé d'API pour votre destination, et vous CloudWatch vous occupez du reste de la configuration. La configuration rapide de partenaire est disponible pour les fournisseurs tiers suivants :

  - Datadog
  - Dynatrace

- New Relic
- Splunk Observability Cloud
- SumoLogic

Vous pouvez diffuser toutes vos CloudWatch statistiques ou utiliser des filtres pour diffuser uniquement des mesures spécifiques. Chaque flux de métriques peut inclure jusqu'à 1 000 filtres qui incluent ou excluent des espaces de noms de métriques ou des métriques spécifiques. Un flux de métriques unique ne peut avoir que des filtres d'inclusion ou d'exclusion, mais pas les deux.

Une fois qu'un flux de métriques est créé, si de nouvelles métriques sont créées qui correspondent aux filtres en place, les nouvelles métriques sont automatiquement incluses dans le flux.

Il n'y a aucune limite sur le nombre de flux de métriques par compte ou par région, ni sur le nombre de mises à jour de métriques diffusées en continu.

Chaque flux peut utiliser le format JSON, OpenTelemetry 1.0.0 ou OpenTelemetry 0.7.0. Vous pouvez modifier le format de sortie d'un flux métrique à tout moment, par exemple pour passer de la OpenTelemetry version 0.7.0 à la version OpenTelemetry 1.0.0. Pour plus d’informations sur les formats de sortie, veuillez consulter la rubrique Formats de sortie de flux de métriques.

Pour les flux de métriques sur les comptes de surveillance, vous pouvez choisir d'inclure ou non les métriques provenant des comptes sources liés à ce compte de surveillance. Pour plus d’informations, consultez CloudWatch observabilité entre comptes.

Les flux de métriques incluent toujours les statistiques `Minimum`, `Maximum`, `SampleCount` et `Sum`. Vous pouvez également choisir d'inclure des statistiques supplémentaires moyennant des frais supplémentaires. Pour plus d’informations, consultez Statistiques pouvant être diffusées.

La tarification des flux de métriques est basée sur le nombre de mises à jour de métrique. Firehose vous facture également des frais pour le flux de diffusion utilisé pour le flux métrique. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

Rubriques

- Configurer un flux de métriques
- Statistiques pouvant être diffusées
- Exploitation et entretien des flux métriques
- Surveillez vos flux de mesures à l'aide de CloudWatch métriques

- Confiance entre Firehose CloudWatch et Firehose
- Formats de sortie de flux de métriques
- Résolution des problèmes

Configurer un flux de métriques

Suivez les étapes décrites dans les sections suivantes pour configurer un flux CloudWatch métrique.

Après la création d'un flux métrique, le temps nécessaire pour que les données métriques apparaissent à la destination dépend des paramètres de mise en mémoire tampon configurés sur le flux de diffusion Firehose. La mise en mémoire tampon est exprimée en taille maximale de la charge utile ou en temps d'attente maximal, selon la première des deux valeurs atteintes. Si celles-ci sont définies sur les valeurs minimales (60 secondes, 1 Mo), la latence attendue est de 3 minutes si les CloudWatch espaces de noms sélectionnés ont des mises à jour métriques actives.

Dans un flux CloudWatch métrique, les données sont envoyées toutes les minutes. Les données peuvent arriver à la destination finale en panne. Toutes les métriques spécifiées dans les espaces de noms spécifiés sont envoyées dans le flux de métriques, à l'exception des métriques dont l'horodatage date de plus de deux jours.

Pour chaque combinaison de nom de la métrique et d'espace de noms que vous diffusez, toutes les combinaisons de dimensions de ce nom de la métrique et de cet espace de noms sont diffusées.

Pour les flux de métriques sur les comptes de surveillance, vous pouvez choisir d'inclure ou non les métriques provenant des comptes sources liés à ce compte de surveillance. Pour plus d’informations, consultez CloudWatch observabilité entre comptes.

Pour créer et gérer des flux de mesures, vous devez être connecté à un compte disposant de la CloudWatchFullAccesspolitique et des `iam:PassRole` autorisations, ou à un compte disposant de la liste d'autorisations suivante :

- `iam:PassRole`
- `cloudwatch:PutMetricStream`
- `cloudwatch:DeleteMetricStream`
- `cloudwatch:GetMetricStream`
- `cloudwatch:ListMetricStreams`
- `cloudwatch:StartMetricStreams`

- `cloudwatch:StopMetricStreams`

Si vous comptez CloudWatch configurer le rôle IAM nécessaire pour les flux métriques, vous devez également disposer des `iam:PutRolePolicy` autorisations `iam:CreateRole` et.

> **Important**
>
> Un utilisateur disposant du `cloudwatch:PutMetricStream` a accès aux données CloudWatch métriques diffusées, même s'il n'en a pas l'`cloudwatch:GetMetricData`autorisation.

Rubriques

- Configuration personnalisée avec Firehose
- Utiliser la configuration rapide d'Amazon S3
- Configuration rapide de partenaire

Configuration personnalisée avec Firehose

Utilisez cette méthode pour créer un flux de mesures et le diriger vers un flux de diffusion Amazon Data Firehose qui transmet vos CloudWatch métriques là où vous le souhaitez. Vous pouvez les diffuser vers un lac de données tel qu'Amazon S3, ou vers n'importe quelle destination ou point de terminaison pris en charge par Firehose, y compris des fournisseurs tiers.

Les formats JSON, OpenTelemetry 1.0.0 et OpenTelemetry 0.7.0 sont pris en charge de manière native, ou vous pouvez configurer des transformations dans votre flux de diffusion Firehose pour convertir les données dans un autre format tel que Parquet. Avec un flux de mesures, vous pouvez continuellement mettre à jour les données de surveillance ou combiner ces données CloudWatch métriques avec des données de facturation et de performance pour créer des ensembles de données riches. Vous pouvez ensuite utiliser des outils tels qu'Amazon Athena pour obtenir des informations sur l'optimisation des coûts, les performances des ressources et l'utilisation des ressources.

Vous pouvez utiliser la CloudWatch console, le AWS CLI AWS CloudFormation, ou le AWS Cloud Development Kit (AWS CDK) pour configurer un flux métrique.

Le flux de diffusion Firehose que vous utilisez pour votre flux métrique doit se trouver dans le même compte et dans la même région que ceux où vous avez configuré le flux métrique. Pour bénéficier

de la fonctionnalité inter-régions, vous pouvez configurer le flux de diffusion Firehose pour qu'il soit diffusé vers une destination finale située dans un autre compte ou une autre région.

CloudWatch console

Cette section décrit comment utiliser la CloudWatch console pour configurer un flux métrique à l'aide de Firehose.

Pour configurer un flux métrique personnalisé à l'aide de Firehose

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), Streams (Flux). Choisissez ensuite Create metric stream (Créer un flux de métriques).
3. (Facultatif) 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 choisir d'inclure ou non les métriques des comptes sources liés dans ce flux de métriques. Pour inclure les métriques des comptes sources, choisissez Include source account metrics (Inclure les métriques des comptes sources).
4. Choisissez Configuration personnalisée avec Firehose.
5. Pour Sélectionnez votre flux Kinesis Data Firehose, sélectionnez le flux de diffusion Firehose à utiliser. Il doit se trouver sur le même compte. Le format par défaut de cette option est OpenTelemetry 0.7.0, mais vous pourrez le modifier ultérieurement au cours de cette procédure.

   Sélectionnez ensuite le flux de diffusion Firehose à utiliser sous Sélectionnez votre flux de diffusion Firehose.
6. (Facultatif) Vous pouvez choisir Sélectionner un rôle de service existant pour utiliser un rôle IAM existant au lieu d'en CloudWatch créer un nouveau pour vous.
7. (Facultatif) Pour modifier le format de sortie par défaut de votre scénario, choisissez Change output format (Modifier le format de sortie). Les formats pris en charge sont JSON, OpenTelemetry 1.0.0 et OpenTelemetry 0.7.0.
8. Pour que les métriques soient diffusées, sélectionnez Toutes les métriques ou Select metrics.

   Si vous sélectionnez Toutes les métriques, toutes les métriques de ce compte seront incluses dans le stream.

   Considérez attentivement s'il faut diffuser toutes les métriques, car plus vous diffusez de métriques, plus vos frais de flux de métriques seront élevés.

Si vous choisissez Sélectionner les métriques, effectuez l'une des opérations suivantes :

- Pour diffuser la plupart des espaces de noms métriques, choisissez Exclude et sélectionnez les espaces de noms ou les métriques à exclure. Lorsque vous spécifiez un espace de noms dans Exclude, vous pouvez éventuellement sélectionner certaines métriques spécifiques de cet espace de noms à exclure. Si vous choisissez d'exclure un espace de noms mais que vous ne sélectionnez pas de métriques dans cet espace de noms, toutes les métriques de cet espace de noms sont exclues.
- Pour inclure uniquement quelques espaces de noms ou métriques dans le flux de métriques, choisissez Inclure, puis sélectionnez les espaces de noms ou les métriques à inclure. Si vous choisissez d'inclure un espace de noms mais que vous ne sélectionnez pas de métriques dans cet espace de noms, toutes les métriques de cet espace de noms sont incluses.

9. (Facultatif) Pour diffuser des statistiques supplémentaires pour certaines de ces mesures au-delà du minimum, du maximum et de la somme, sélectionnez Ajouter des statistiques supplémentaires. SampleCount Choisissez soit Add recommended metrics (Ajouter des métriques recommandées) pour ajouter des statistiques couramment utilisées, ou sélectionnez manuellement l'espace de noms et le nom de la métrique pour lesquels diffuser des statistiques supplémentaires. Ensuite, sélectionnez les statistiques supplémentaires à diffuser.

   Pour ensuite choisir un autre groupe de métriques pour lequel diffuser un autre ensemble de statistiques supplémentaires, choisissez Add additional statistics (Ajouter des statistiques supplémentaires). Chaque métrique peut inclure jusqu'à 20 statistiques supplémentaires, et jusqu'à 100 métriques dans un flux de métriques peuvent inclure des statistiques supplémentaires.

   Le streaming de statistiques supplémentaires entraîne des frais supplémentaires. Pour plus d’informations, consultez Statistiques pouvant être diffusées.

   Pour obtenir des définitions des statistiques supplémentaires, veuillez consulter CloudWatch définitions des statistiques.

10. (Facultatif) Personnalisez le nom du nouveau flux de métriques sous Metric stream name (Nom du flux de métriques).
11. Choisissez Create metric stream (Créer un filtre de métriques).

AWS CLI ou AWS API

Suivez les étapes ci-dessous pour créer un flux CloudWatch métrique.

Pour utiliser l' AWS API AWS CLI or pour créer un flux métrique

1. Si vous diffusez en continu sur Amazon S3, créez d'abord le compartiment. Pour de plus amples informations, veuillez consulter Créer un compartiment dans.
2. Créez le flux de diffusion Firehose. Pour plus d'informations, consultez Création d'un flux Firehose.
3. Créez un rôle IAM qui permet d' CloudWatch écrire dans le flux de diffusion Firehose. Pour plus d'informations sur le contenu de ce rôle, consultez Confiance entre Firehose CloudWatch et Firehose.
4. Utilisez la commande `aws cloudwatch put-metric-stream` CLI ou l'`PutMetricStream`API pour créer le flux CloudWatch métrique.

AWS CloudFormation

Vous pouvez l' AWS CloudFormation utiliser pour configurer un flux métrique. Pour plus d'informations, consultez AWS::CloudWatch::MetricStream.

À utiliser AWS CloudFormation pour créer un flux métrique

1. Si vous diffusez en continu sur Amazon S3, créez d'abord le compartiment. Pour de plus amples informations, veuillez consulter Créer un compartiment dans.
2. Créez le flux de diffusion Firehose. Pour plus d'informations, consultez Création d'un flux Firehose.
3. Créez un rôle IAM qui permet d' CloudWatch écrire dans le flux de diffusion Firehose. Pour plus d'informations sur le contenu de ce rôle, consultez Confiance entre Firehose CloudWatch et Firehose.
4. Créez le stream dans AWS CloudFormation. Pour plus d'informations, consultez AWS::CloudWatch::MetricStream.

AWS Cloud Development Kit (AWS CDK)

Vous pouvez l' AWS Cloud Development Kit (AWS CDK) utiliser pour configurer un flux métrique.

Pour utiliser le AWS CDK pour créer un flux métrique

1. Si vous diffusez en continu sur Amazon S3, créez d'abord le compartiment. Pour de plus amples informations, veuillez consulter Créer un compartiment dans.
2. Créez le flux de diffusion Firehose. Pour plus d'informations, consultez Création d'un flux de diffusion Amazon Data Firehose.
3. Créez un rôle IAM qui permet d' CloudWatch écrire dans le flux de diffusion Firehose. Pour plus d'informations sur le contenu de ce rôle, consultez Confiance entre Firehose CloudWatch et Firehose.
4. Créez le flux de métriques. La ressource du flux métrique est disponible AWS CDK sous la forme d'une construction de niveau 1 (L1) nomméeCfnMetricStream. Pour de plus amples informations, consultez Utilisation des constructions L1.

Utiliser la configuration rapide d'Amazon S3

La méthode Quick S3 Setup fonctionne bien si vous souhaitez configurer rapidement un flux vers Amazon S3 et si vous n'avez besoin d'aucune transformation de formatage autre que les formats JSON, OpenTelemetry 1.0.0 et OpenTelemetry 0.7.0 pris en charge. CloudWatch créera toutes les ressources nécessaires, y compris le flux de diffusion Firehose et les rôles IAM nécessaires. Le format par défaut de cette option est JSON, mais vous pouvez le modifier lors de la configuration du flux.

Par contre, si vous souhaitez que le format final soit le format Parquet ou Optimized Row Columnar (ORC), il convient de suivre les étapes décrites dans Configuration personnalisée avec Firehose.

CloudWatch console

Cette section décrit comment utiliser la CloudWatch console pour configurer un flux métrique Amazon S3 à l'aide de la configuration rapide de S3.

Pour configurer un flux de métriques à l'aide de la Configuration rapide de S3

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), Streams (Flux). Choisissez ensuite Create metric stream (Créer un flux de métriques).
3. (Facultatif) 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 choisir d'inclure ou non les métriques des comptes sources liés dans ce flux de métriques. Pour inclure les métriques

des comptes sources, choisissez Include source account metrics (Inclure les métriques des comptes sources).

4. Choisissez Configuration rapide de S3. CloudWatch créera toutes les ressources nécessaires, y compris le flux de diffusion Firehose et les rôles IAM nécessaires. Le format par défaut de cette option est JSON, mais vous pouvez le modifier ultérieurement dans cette procédure.
5. (Facultatif) Choisissez Sélectionner les ressources existantes pour utiliser un compartiment S3 existant ou des rôles IAM existants au lieu d'en CloudWatch créer de nouveaux pour vous.
6. (Facultatif) Pour modifier le format de sortie par défaut de votre scénario, choisissez Change output format (Modifier le format de sortie). Les formats pris en charge sont JSON, OpenTelemetry 1.0.0 et OpenTelemetry 0.7.0.
7. Pour que les métriques soient diffusées, sélectionnez Toutes les métriques ou Select metrics.

   Si vous sélectionnez Toutes les métriques, toutes les métriques de ce compte seront incluses dans le stream.

   Considérez attentivement s'il faut diffuser toutes les métriques, car plus vous diffusez de métriques, plus vos frais de flux de métriques seront élevés.

   Si vous choisissez Sélectionner les métriques, effectuez l'une des opérations suivantes :

   - Pour diffuser la plupart des espaces de noms métriques, choisissez Exclude et sélectionnez les espaces de noms ou les métriques à exclure. Lorsque vous spécifiez un espace de noms dans Exclude, vous pouvez éventuellement sélectionner certaines métriques spécifiques de cet espace de noms à exclure. Si vous choisissez d'exclure un espace de noms mais que vous ne sélectionnez pas de métriques dans cet espace de noms, toutes les métriques de cet espace de noms sont exclues.
   - Pour inclure uniquement quelques espaces de noms ou métriques dans le flux de métriques, choisissez Inclure, puis sélectionnez les espaces de noms ou les métriques à inclure. Si vous choisissez d'inclure un espace de noms mais que vous ne sélectionnez pas de métriques dans cet espace de noms, toutes les métriques de cet espace de noms sont incluses.
8. (Facultatif) Pour diffuser des statistiques supplémentaires pour certaines de ces mesures au-delà du minimum, du maximum et de la somme, sélectionnez Ajouter des statistiques supplémentaires. SampleCount Choisissez soit Add recommended metrics (Ajouter des métriques recommandées) pour ajouter des statistiques couramment utilisées, ou sélectionnez manuellement l'espace de noms et le nom de la métrique pour lesquels diffuser des statistiques supplémentaires. Ensuite, sélectionnez les statistiques supplémentaires à diffuser.

Pour ensuite choisir un autre groupe de métriques pour lequel diffuser un autre ensemble de statistiques supplémentaires, choisissez Add additional statistics (Ajouter des statistiques supplémentaires). Chaque métrique peut inclure jusqu'à 20 statistiques supplémentaires, et jusqu'à 100 métriques dans un flux de métriques peuvent inclure des statistiques supplémentaires.

Le streaming de statistiques supplémentaires entraîne des frais supplémentaires. Pour plus d'informations, consultez Statistiques pouvant être diffusées.

Pour obtenir des définitions des statistiques supplémentaires, veuillez consulter CloudWatch définitions des statistiques.

9. (Facultatif) Personnalisez le nom du nouveau flux de métriques sous Metric stream name (Nom du flux de métriques).
10. Choisissez Create metric stream (Créer un filtre de métriques).

Configuration rapide de partenaire

CloudWatch fournit une expérience de configuration rapide aux partenaires tiers suivants. Pour utiliser ce flux de travail, vous devez uniquement fournir une URL de destination et une clé d'API pour votre destination. CloudWatch gère le reste de la configuration, y compris la création du flux de diffusion Firehose et des rôles IAM nécessaires.

> **Important**
>
> Avant d'utiliser la configuration rapide de partenaire pour créer un flux de métriques, nous vous recommandons vivement de lire la documentation de ce partenaire, dont le lien figure dans la liste suivante.

- Datadog
- Dynatrace
- New Relic
- Splunk Observability Cloud
- SumoLogic

Lorsque vous configurez un flux métrique pour l'un de ces partenaires, le flux est créé avec certains paramètres par défaut, comme indiqué dans les sections suivantes.

Rubriques

Configurer un flux de métriques à l'aide de la configuration rapide de partenaire

CloudWatch fournit une option de configuration rapide pour certains partenaires tiers. Avant d'exécuter la procédure indiquée dans cette section, vous devez satisfaire à certaines exigences du partenaire. Ces informations peuvent inclure une URL de destination et/ou une clé d'API pour la destination de votre partenaire. Vous devriez également lire la documentation sur le site Web du partenaire dont le lien figure dans la section précédente, ainsi que les paramètres par défaut pour ce partenaire répertoriés dans les sections suivantes.

Pour diffuser vers une destination tierce non prise en charge par la configuration rapide, vous pouvez suivre les instructions de la section Configuration personnalisée avec Firehose Pour configurer un flux à l'aide de Firehose, puis envoyer ces métriques de Firehose à la destination finale.

Pour utiliser la configuration rapide de partenaire afin de créer un flux de métriques destiné à un fournisseur tiers

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), Streams (Flux). Choisissez ensuite Create metric stream (Créer un flux de métriques).
3. (Facultatif) 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 choisir d'inclure ou non les métriques des comptes sources liés dans ce flux de métriques. Pour inclure les métriques des comptes sources, choisissez Include source account metrics (Inclure les métriques des comptes sources).
4. Choisissez Quick Amazon Web Services Partner Setup

5. Sélectionnez le nom du partenaire vers lequel vous souhaitez diffuser les métriques.
6. Pour l'URL du point de terminaison, saisissez l'URL de destination.
7. Pour la clé d'accès ou la clé d'API, saisissez la clé d'accès du partenaire. Une clé d'accès n'est pas exigée pour tous les partenaires.
8. Pour que les métriques soient diffusées, sélectionnez Toutes les métriques ou Select metrics.

   Si vous sélectionnez Toutes les métriques, toutes les métriques de ce compte seront incluses dans le stream.

   Considérez attentivement s'il faut diffuser toutes les métriques, car plus vous diffusez de métriques, plus vos frais de flux de métriques seront élevés.

   Si vous choisissez Sélectionner les métriques, effectuez l'une des opérations suivantes :

   - Pour diffuser la plupart des espaces de noms métriques, choisissez Exclude et sélectionnez les espaces de noms ou les métriques à exclure. Lorsque vous spécifiez un espace de noms dans Exclude, vous pouvez éventuellement sélectionner certaines métriques spécifiques de cet espace de noms à exclure. Si vous choisissez d'exclure un espace de noms mais que vous ne sélectionnez pas de métriques dans cet espace de noms, toutes les métriques de cet espace de noms sont exclues.
   - Pour inclure uniquement quelques espaces de noms ou métriques dans le flux de métriques, choisissez Inclure, puis sélectionnez les espaces de noms ou les métriques à inclure. Si vous choisissez d'inclure un espace de noms mais que vous ne sélectionnez pas de métriques dans cet espace de noms, toutes les métriques de cet espace de noms sont incluses.
9. (Facultatif) Pour diffuser des statistiques supplémentaires pour certaines de ces mesures au-delà du minimum, du maximum et de la somme, sélectionnez Ajouter des statistiques supplémentaires. SampleCount Choisissez soit Add recommended metrics (Ajouter des métriques recommandées) pour ajouter des statistiques couramment utilisées, ou sélectionnez manuellement l'espace de noms et le nom de la métrique pour lesquels diffuser des statistiques supplémentaires. Ensuite, sélectionnez les statistiques supplémentaires à diffuser.

   Pour ensuite choisir un autre groupe de métriques pour lequel diffuser un autre ensemble de statistiques supplémentaires, choisissez Add additional statistics (Ajouter des statistiques supplémentaires). Chaque métrique peut inclure jusqu'à 20 statistiques supplémentaires, et jusqu'à 100 métriques dans un flux de métriques peuvent inclure des statistiques supplémentaires.

Le streaming de statistiques supplémentaires entraîne des frais supplémentaires. Pour plus d'informations, consultez Statistiques pouvant être diffusées.

Pour obtenir des définitions des statistiques supplémentaires, veuillez consulter CloudWatch définitions des statistiques.

10. (Facultatif) Personnalisez le nom du nouveau flux de métriques sous Metric stream name (Nom du flux de métriques).
11. Choisissez Create metric stream (Créer un filtre de métriques).

Valeurs par défaut du flux Datadog

Les flux de configuration rapide de partenaire vers Datadog utilisent les valeurs par défaut suivantes :

- Format de sortie : OpenTelemetry 0.7.0
- Encodage du contenu du stream Firehose (GZIP)
- Options de mise en mémoire tampon des flux Firehose Intervalle de 60 secondes, taille de 4 Mo
- Option Firehose Stream Retry (durée de 60 secondes)

Lorsque vous utilisez la configuration rapide de partenaire pour créer un flux de métriques vers Datadog et que vous diffusez certaines métriques, celles-ci incluent par défaut des statistiques supplémentaires. La diffusion de statistiques supplémentaires entraîne des frais supplémentaires. Pour plus d'informations sur les statistiques et leurs frais, veuillez consulter Statistiques pouvant être diffusées.

La liste suivante indique les métriques pour lesquelles des statistiques supplémentaires sont diffusées par défaut, si vous choisissez de diffuser ces statistiques. Vous pouvez choisir de désélectionner ces statistiques supplémentaires avant de démarrer le flux.

- **`Duration`** dans **`AWS/Lambda`** : p50, p80, p95, p99, p99.9
- **`PostRuntimeExtensionDuration`** dans **`AWS/Lambda`** : p50, p99
- **`FirstByteLatency`** et **`TotalRequestLatency`** dans **`AWS/S3`** : p50, p90, p95, p99, p99.9
- **`ResponseLatency`** dans **`AWS/Polly`** et **`TargetResponseTime`** dans AWS/ApplicationELB : p50, p90, p95, p99
- **`Latency`** et **`IntegrationLatency`** dans **`AWS/ApiGateway`** : p90, p95, p99
- **`Latency`** et **`TargetResponseTime`** dans **`AWS/ELB`** : p95, p99

- **`RequestLatency`** dans **`AWS/AppRunner`** : p50, p95, p99
- **`ActivityTime`**, **`ExecutionTime`**, **`LambdaFunctionRunTime`**, **`LambdaFunctionScheduleTime`**, **`LambdaFunctionTime`**, **`ActivityRunTime`** et **`ActivityScheduleTime`** dans **`AWS/States`** : p95, p99
- **`EncoderBitRate`**, **`ConfiguredBitRate`** et **`ConfiguredBitRateAvailable`** dans **`AWS/MediaLive`** : p90
- **`Latency`** dans **`AWS/AppSync`** : p90

Paramètres par défaut du flux Dynatrace

Les flux de configuration rapide de partenaire vers Dynatrace utilisent les valeurs par défaut suivantes :

- Format de sortie : OpenTelemetry 0.7.0
- Encodage du contenu du stream Firehose (GZIP)
- Options de mise en mémoire tampon des flux Firehose Intervalle de 60 secondes, taille de 5 Mo
- Option Firehose Stream Retry (durée de 600 secondes)

Paramètres par défaut du fluc New Relic

Les flux de configuration rapide de partenaire vers New Relic utilisent les valeurs par défaut suivantes :

- Format de sortie : OpenTelemetry 0.7.0
- Encodage du contenu du stream Firehose (GZIP)
- Options de mise en mémoire tampon des flux Firehose Intervalle de 60 secondes, taille de 1 Mo
- Option Firehose Stream Retry (durée de 60 secondes)

Paramètres par défaut du flux Splunk Observability Cloud

Les flux de configuration rapide de partenaire vers Splunk Observability Cloud utilisent les valeurs par défaut suivantes :

- Format de sortie : OpenTelemetry 0.7.0
- Encodage du contenu du stream Firehose (GZIP)
- Options de mise en mémoire tampon des flux Firehose Intervalle de 60 secondes, taille de 1 Mo

- Option Firehose Stream Retry (durée de 300 secondes)

Paramètres par défaut du flux Sumo Logic

Les flux de configuration rapide de partenaire vers Sumo Logic utilisent les valeurs par défaut suivantes :

- Format de sortie : OpenTelemetry 0.7.0
- Encodage du contenu du stream Firehose (GZIP)
- Options de mise en mémoire tampon des flux Firehose Intervalle de 60 secondes, taille de 1 Mo
- Option Firehose Stream Retry (durée de 60 secondes)

Statistiques pouvant être diffusées

Les flux de métriques incluent toujours les statistiques suivantes : `Minimum`, `Maximum`, `SampleCount` et `Sum`. Vous pouvez également choisir d'inclure les statistiques supplémentaires suivantes dans un flux de métriques. Ce choix se fait sur une base par métrique. Pour de plus amples informations sur ces statistiques, veuillez consulter CloudWatch définitions des statistiques.

- Valeurs percentiles telles que p95 ou p99 (pour les flux au format JSON ou au format) OpenTelemetry
- Moyenne ajustée (uniquement pour les flux au format JSON)
- Moyenne winsorisée (uniquement pour les flux au format JSON)
- Nombre ajusté (uniquement pour les flux au format JSON)
- Somme ajustée (uniquement pour les flux au format JSON)
- Rang centile (uniquement pour les flux au format JSON)
- Moyenne interquartile (uniquement pour les flux au format JSON)

Le streaming de statistiques supplémentaires entraîne des frais supplémentaires. La diffusion de une à cinq de ces statistiques supplémentaires pour une métrique particulière est facturée sous forme de mise à jour de métrique supplémentaire. Par la suite, chaque ensemble supplémentaire comprenant jusqu'à cinq de ces statistiques est facturé comme une autre mise à jour de métrique.

Par exemple, supposons que pour une métrique, vous diffusez les six statistiques supplémentaires suivantes : p95, p99, p99,9, moyenne ajustée, moyenne winsorisée et Somme ajustée. Chaque

mise à jour de cette métrique est facturée sous la forme de trois mises à jour de métriques : une pour la mise à jour des métriques qui inclut les statistiques par défaut, une pour les cinq premières statistiques supplémentaires et une pour la sixième statistique supplémentaire. L'ajout de quatre statistiques supplémentaires pour un total de dix statistiques n'augmenterait pas la facturation, mais une onzième statistique supplémentaire le ferait.

Lorsque vous spécifiez un nom de la métrique et une combinaison d'espaces de noms pour diffuser des statistiques supplémentaires, toutes les combinaisons de dimensions de ce nom de la métrique et de cet espace de noms sont diffusées avec les statistiques supplémentaires.

CloudWatch metric streams publie une nouvelle métrique`TotalMetricUpdate`, qui reflète le nombre de base de mises à jour de métriques plus les mises à jour de métriques supplémentaires liées à la diffusion de statistiques supplémentaires. Pour plus d'informations, consultez Surveillez vos flux de mesures à l'aide de CloudWatch métriques.

Pour plus d'informations, consultez Amazon CloudWatch Pricing.

> **Note**
>
> Certaines métriques ne prennent pas en charge les centiles. Les statistiques de centile de ces métriques sont exclues du flux et n'entraînent pas de frais de flux de métriques. Un exemple de ces statistiques qui ne prennent pas en charge les centiles sont des métriques de l'espace de noms `AWS/ECS`.

Les statistiques supplémentaires que vous configurez sont diffusées uniquement si elles correspondent aux filtres du flux. Par exemple, si vous créez un flux qui contient uniquement `EC2` et `RDS` dans les filtres d'inclusion, puis dans les listes de configuration de vos statistiques `EC2` et `Lambda`, alors le flux inclut les métriques `EC2` avec statistiques supplémentaires, les métriques `RDS` avec uniquement les statistiques par défaut, et ne comprennent aucunes statistiques `Lambda`.

Exploitation et entretien des flux métriques

Les flux de métriques sont toujours dans l'un des deux états, Running (En cours d'exécution) ou Stopped (Arrêt).

- Running (En cours d'exécution) : le flux de métriques s'exécute correctement. Il se peut que les données de métrique ne soient pas diffusées vers la destination en raison des filtres présents sur le flux.

- Stopped (Arrêt) : le flux de métriques a été explicitement arrêté par quelqu'un, et non à cause d'une erreur. Il peut être utile d'arrêter votre flux pour interrompre temporairement la diffusion de données sans supprimer le flux.

Si vous arrêtez et redémarrez un flux métrique, les données métriques publiées CloudWatch lors de l'arrêt du flux métrique ne sont pas rechargées dans le flux métrique.

Si vous modifiez le format de sortie d'un flux de métriques, dans certains cas, vous pouvez voir une petite quantité de données de métrique écrites vers la destination dans l'ancien format et le nouveau format. Pour éviter cette situation, vous pouvez créer un nouveau flux de diffusion Firehose avec la même configuration que votre flux de diffusion actuel, puis passer au nouveau flux de diffusion Firehose et modifier le format de sortie en même temps. De cette façon, les enregistrements Kinesis avec un format de sortie différent sont stockés sur Amazon S3 dans des objets distincts. Plus tard, vous pourrez rediriger le trafic vers le flux de diffusion Firehose d'origine et supprimer le second flux de diffusion.

Pour afficher, modifier, arrêter et démarrer vos flux de métriques

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), Streams (Flux).

   La liste des flux s'affiche, et la colonne Status (État) indique si chaque flux est en cours d'exécution ou arrêté.
3. Pour arrêter ou démarrer un flux de métriques, sélectionnez le flux et choisissez Stop (Arrêter) ou Start (Démarrer).
4. Pour afficher les détails d'un flux de métriques, sélectionnez le flux et choisissez View details (Afficher les détails).
5. Pour modifier le format de sortie, les filtres, le flux Firehose de destination ou les rôles du flux, choisissez Modifier et apportez les modifications souhaitées.

   Si vous modifiez les filtres, il peut y avoir des lacunes dans les données de métrique pendant la transition.

Surveillez vos flux de mesures à l'aide de CloudWatch métriques

Les flux métriques émettent CloudWatch des métriques concernant leur état et leur fonctionnement dans l'espace de `AWS/CloudWatch/MetricStreams` noms. Les métriques suivantes sont émises.

Les deux métriques sont émises avec une dimension `MetricStreamName` et sans dimension. Vous pouvez utiliser les métriques sans dimensions pour afficher les métriques agrégées pour tous vos flux de métriques. Vous pouvez utiliser les métriques avec la dimension `MetricStreamName` pour afficher les métriques concernant uniquement ce flux de métrique.

Pour ces deux métriques, les valeurs ne sont émises que pour les flux de métriques qui se trouvent dans l'état Running (En cours d'exécution).

Métrique	Description
`MetricUpdate`	Nombre de mises à jour de métriques envoyées au flux de métrique. Si aucune mise à jour de métrique n'est diffusée en continu pendant une période donnée, cette métrique n'est pas émise pendant cette période. Si vous arrêtez le flux de métrique, cette métrique cesse d'être émise jusqu'à ce que le flux de métrique soit redémarré. Statistique valide : `Sum` Unités : aucune
`TotalMetricUpdate`	Ce chiffre est calculé sous la forme de MetricUpdate +, un nombre basé sur des statistiques supplémentaires diffusées en continu. Pour chaque combinaison unique d'espace de noms et de noms de la métrique, la diffusion de 1 à 5 statistiques supplémentaires ajoute 1 au `TotalMetricUpdate` , la diffusion de 6 à 10 statistiques supplémentaires ajoute 2 au `TotalMetricUpdate` , etc. Statistique valide : `Sum` Unités : aucune
`PublishErrorRate`	Le nombre d'erreurs irrécupérables qui se produisent lors de l'introduction de données dans le flux de diffusion Firehose. Si aucune erreur ne se produit pendant une période donnée, cette métrique n'est pas émise pendant cette période. Si vous arrêtez le flux de métrique, cette métrique cesse d'être émise jusqu'à ce que le flux de métrique soit redémarré.

Métrique	Description
	Statistique valide : `Average` pour voir le taux de mises à jour des métriques qui ne peuvent pas être écrites. La valeur doit être comprise entre 0,0 et 1,0. Unités : aucune

Confiance entre Firehose CloudWatch et Firehose

Le flux de diffusion Firehose doit être fiable CloudWatch via un rôle IAM doté d'autorisations d'écriture sur Firehose. Ces autorisations peuvent être limitées au seul flux de diffusion Firehose utilisé par le flux CloudWatch métrique. Le rôle IAM doit faire confiance au service principal `streams.metrics.cloudwatch.amazonaws.com`.

Si vous utilisez la CloudWatch console pour créer un flux de mesures, vous pouvez avoir CloudWatch créé le rôle avec les autorisations appropriées. Si vous utilisez une autre méthode pour créer un flux de métriques ou si vous souhaitez créer le rôle IAM lui-même, elle doit contenir la stratégie d'autorisations et la stratégie d'approbation suivantes.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "firehose:PutRecord",
                "firehose:PutRecordBatch"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:firehose:region:account-id:deliverystream/*"
        }
    ]
}
```

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

```
        "Service": "streams.metrics.cloudwatch.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Les données métriques sont diffusées CloudWatch vers le flux de diffusion Firehose de destination pour le compte de la source propriétaire de la ressource du flux métrique.

Formats de sortie de flux de métriques

Les données d'un flux CloudWatch métrique peuvent être au format JSON ou au OpenTelemetry format. Actuellement, les formats OpenTelemetry 1.0.0 et 0.7.0 sont pris en charge.

Table des matières

Format JSON

Dans un flux CloudWatch métrique utilisant le format JSON, chaque enregistrement Firehose contient plusieurs objets JSON séparés par un caractère de nouvelle ligne (\n). Chaque objet comprend un point de données unique d'une seule métrique.

Le format JSON utilisé est entièrement compatible avec AWS Glue et avec Amazon Athena. Si vous disposez d'un flux de diffusion Firehose et d'un AWS Glue tableau correctement formaté, le format peut être automatiquement transformé au format Parquet ou au format ORC (Optimized Row Columnar) avant d'être stocké dans S3. Pour plus d'informations sur la transformation du format, consultez la section Conversion du format de votre enregistrement d'entrée dans Firehose. Pour plus

d'informations sur le format approprié pour AWS Glue, consultezQuel schéma AWS Glue devrais-je utiliser pour le format de sortie JSON ?.

Au format JSON, les valeurs valides pour `unit` sont identiques à celles de `unit`dans la structure d'API `MetricDatum`. Pour plus d'informations, consultez MetricDatum. La valeur du champ `timestamp` est en millisecondes d'époque, comme `1616004674229`.

Voici un exemple du format. Dans cet exemple, le format JSON est mis en forme afin de faciliter la lecture. En pratique, l'ensemble du format se trouve sur une seule ligne.

```
{
    "metric_stream_name": "MyMetricStream",
    "account_id": "1234567890",
    "region": "us-east-1",
    "namespace": "AWS/EC2",
    "metric_name": "DiskWriteOps",
    "dimensions": {
        "InstanceId": "i-123456789012"
    },
    "timestamp": 1611929698000,
    "value": {
        "count": 3.0,
        "sum": 20.0,
        "max": 18.0,
        "min": 0.0,
        "p99": 17.56,
        "p99.9": 17.8764,
        "TM(25%:75%)": 16.43
    },
    "unit": "Seconds"
}
```

Quel schéma AWS Glue devrais-je utiliser pour le format de sortie JSON ?

Voici un exemple de représentation JSON de la AWS Glue table `StorageDescriptor` for an, qui serait ensuite utilisée par Firehose. Pour plus d'informations sur`StorageDescriptor`, voir StorageDescriptor.

```
{
  "Columns": [
    {
      "Name": "metric_stream_name",
```

```
      "Type": "string"
    },
    {
      "Name": "account_id",
      "Type": "string"
    },
    {
      "Name": "region",
      "Type": "string"
    },
    {
      "Name": "namespace",
      "Type": "string"
    },
    {
      "Name": "metric_name",
      "Type": "string"
    },
    {
      "Name": "timestamp",
      "Type": "timestamp"
    },
    {
      "Name": "dimensions",
      "Type": "map<string,string>"
    },
    {
      "Name": "value",
      "Type":
"struct<min:double,max:double,count:double,sum:double,p99:double,p99.9:double>"
    },
    {
      "Name": "unit",
      "Type": "string"
    }
  ],
  "Location": "s3://my-s3-bucket/",
  "InputFormat": "org.apache.hadoop.mapred.TextInputFormat",
  "OutputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
  "SerdeInfo": {
    "SerializationLibrary": "org.apache.hive.hcatalog.data.JsonSerDe"
  },
  "Parameters": {
    "classification": "json"
```

```
  }
}
```

L'exemple précédent concerne les données écrites sur Amazon S3 au format JSON. Remplacez les valeurs des champs suivants par les valeurs indiquées pour stocker les données au format Parquet ou ORC (Optimized Row Columnar).

- Parquet :
  - Format d'entrée : org.apache.hadoop.hive.ql.io.parquet. MapredParquetInputFormat
  - Format de sortie : org.apache.hadoop.hive.ql.io.parquet. MapredParquetOutputFormat
  - SerDeInfo.serializationLib : org.apache.hadoop.hive.ql.io.parquet.serde. ParquetHiveSerDe
  - parameters.classification : parquet
- ORC :
  - Format d'entrée : org.apache.hadoop.hive.ql.io.orc. OrcInputFormat
  - Format de sortie : org.apache.hadoop.hive.ql.io.orc. OrcOutputFormat
  - SerDeInfo.serializationLib : org.apache.hadoop.hive.ql.io.orc. OrcSerde
  - parameters.classification : orc

OpenTelemetry Format 1.0.0

> **Note**
>
> Avec le format OpenTelemetry 1.0.0, les attributs métriques sont codés sous forme de liste d'`KeyValue`objets au lieu du `StringKeyValue` type utilisé dans le format 0.7.0. En tant que consommateur, il s'agit du seul changement majeur entre les formats 0.7.0 et 1.0.0. Un analyseur généré à partir des fichiers proto 0.7.0 n'analysera pas les attributs métriques codés au format 1.0.0. Il en va de même en sens inverse, un analyseur généré à partir des fichiers proto 1.0.0 n'analysera pas les attributs métriques codés au format 0.7.0.

OpenTelemetry est un ensemble d'outils, d'API et de SDK. Vous pouvez l'utiliser pour instrumenter, générer, collecter et exporter des données de télémétrie (métriques, journaux et traces) à des fins d'analyse. OpenTelemetry fait partie de la Cloud Native Computing Foundation. Pour plus d'informations, consultez OpenTelemetry.

Pour plus d'informations sur la spécification OpenTelemetry 1.0.0 complète, voir Release version 1.0.0.

Un enregistrement Kinesis peut contenir une ou plusieurs structures de `ExportMetricsServiceRequest` OpenTelemetry données. Chaque structure de données commence par un en-tête avec un `UnsignedVarInt32` indiquant la longueur du registre en octets. Chaque `ExportMetricsServiceRequest` peut contenir des données provenant de plusieurs métriques à la fois.

Ce qui suit est une représentation sous forme de chaîne du message de la structure de `ExportMetricsServiceRequest` OpenTelemetry données. OpenTelemetry sérialise le protocole binaire Google Protocol Buffers, ce qui n'est pas lisible par l'homme.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "123456789012"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  scope_metrics {
```

```
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "NoneTranslated"
      summary {
        data_points {
          start_time_unix_nano: 60000000000
          time_unix_nano: 120000000000
          count: 1
          sum: 1.0
          quantile_values {
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
            }
          }
          attributes {
            key: "Dimensions"
            value {
              kvlist_value {
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
```

```
                }
              }
            }
          }
        }
        data_points {
          start_time_unix_nano: 70000000000
          time_unix_nano: 130000000000
          count: 2
          sum: 5.0
          quantile_values {
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
          attributes {
            key: "Namespace"
            value {
              string_value: "AWS/DynamoDB"
            }
          }
          attributes {
            key: "MetricName"
            value {
              string_value: "ConsumedReadCapacityUnits"
            }
          }
          attributes {
            key: "Dimensions"
            value {
              kvlist_value {
                values {
                  key: "TableName"
                  value {
                    string_value: "MyTable"
                  }
                }
              }
            }
          }
        }
```

```
      }
    }
  }
}
```

Objet de haut niveau pour sérialiser les données métriques OpenTelemetry

`ExportMetricsServiceRequest`est le wrapper de haut niveau pour sérialiser la charge utile d'un OpenTelemetry exportateur. Il contient une ou plusieurs `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics`est l'objet de niveau supérieur destiné à représenter les MetricData objets.

```
// A collection of ScopeMetrics from a Resource.
message ResourceMetrics {
  reserved 1000;

  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated ScopeMetrics scope_metrics = 2;

  // This schema_url applies to the data in the "resource" field. It does not apply
  // to the data in the "scope_metrics" field which have their own schema_url field.
  string schema_url = 3;
}
```

L'objet Ressource

`Resource` est un objet de paire de valeurs qui contient des informations sur la ressource qui a généré les métriques. Pour les métriques créées par AWS, la structure de données contient

l'Amazon Resource Name (ARN) de la ressource liée à la métrique, telle qu'une instance EC2 ou un compartiment S3.

L'objet `Resource` contient un attribut appelé `attributes`, qui stocke une liste des paires de valeurs clés.

- `cloud.account.id` contient l'ID de compte
- `cloud.region` contient la région
- `aws.exporter.arn` contient l'ARN du flux de métriques
- `cloud.provider` est toujours `aws`.

```
// Resource information.
message Resource {
  // Set of attributes that describe the resource.
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
 then
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

L' ScopeMetrics objet

Le champ `scope` ne sera pas rempli. Seul le champ des métriques que nous exportons est renseigné.

```
// A collection of Metrics produced by an Scope.
message ScopeMetrics {
  // The instrumentation scope information for the metrics in this message.
  // Semantically when InstrumentationScope isn't set, it is equivalent with
  // an empty instrumentation scope name (unknown).
  opentelemetry.proto.common.v1.InstrumentationScope scope = 1;

  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;

  // This schema_url applies to all metrics in the "metrics" field.
```

```
  string schema_url = 3;
}
```

L'objet Metric

L'objet de métrique contient quelques métadonnées et un champ de données `Summary` qui contient une liste de `SummaryDataPoint`.

Pour les flux de métriques, les métadonnées sont les suivantes :

- `name` sera `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` sera vide.
- `unit` sera rempli en mappant l'unité de référence de la métrique à la variante sensible à la casse du code unifié pour les unités de mesure. Pour de plus amples informations, veuillez consulter Traductions au OpenTelemetry format 1.0.0 and le Code unifié des unités de mesure.
- `type` sera `SUMMARY`

```
message Metric {
  reserved 4, 6, 8;

  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  // Data determines the aggregation type (if any) of the metric, what is the
  // reported value type for the data points, as well as the relatationship to
  // the time interval over which they are reported.
  oneof data {
    Gauge gauge = 5;
    Sum sum = 7;
    Histogram histogram = 9;
    ExponentialHistogram exponential_histogram = 10;
    Summary summary = 11;
  }
```

```
}

message Summary {
  repeated SummaryDataPoint data_points = 1;
}
```

L' SummaryDataPoint objet

L' SummaryDataPoint objet contient la valeur d'un point de données unique dans une série chronologique dans une DoubleSummary métrique.

```
// SummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message SummaryDataPoint {
  reserved 1;

  // The set of key/value pairs that uniquely identify the timeseries from
  // where this point belongs. The list may be empty (may contain 0 elements).
  // Attribute keys MUST be unique (it is not allowed to have more than one
  // attribute with the same key).
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 7;

  // StartTimeUnixNano is optional but strongly encouraged, see the
  // the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 start_time_unix_nano = 2;

  // TimeUnixNano is required, see the detailed comments above Metric.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  //
  // Note: Sum should only be filled out when measuring non-negative discrete
  // events, and is assumed to be monotonic over the values of these events.
```

```
  // Negative events *can* be recorded, but sum should not be filled out when
  // doing so.  This is specifically to enforce compatibility w/ OpenMetrics,
  // see: https://github.com/OpenObservability/OpenMetrics/blob/main/specification/
OpenMetrics.md#summary
  double sum = 5;

  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  //
  // See the following issue for more context:
  // https://github.com/open-telemetry/opentelemetry-proto/issues/125
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    //
    // Quantile values must NOT be negative.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;

  // Flags that apply to this specific data point.  See DataPointFlags
  // for the available flags and their meaning.
  uint32 flags = 8;
}
```

Pour plus d'informations, consultez Traductions au OpenTelemetry format 1.0.0.

Traductions au OpenTelemetry format 1.0.0

CloudWatch effectue certaines transformations pour mettre CloudWatch les données en OpenTelemetry format.

Traduction de l'espace de noms, du nom de la métrique et des dimensions

Ces attributs sont des paires de valeurs clés codées dans le mappage.

- Un attribut possède la clé `Namespace` et sa valeur est l'espace de noms de la métrique.
- Un attribut possède la clé `MetricName` et sa valeur est le nom de la métrique.
- Une paire a la clé `Dimensions` et sa valeur est une liste imbriquée des paires clé-valeur. Chaque paire de cette liste correspond à une dimension CloudWatch métrique, où la clé de la paire est le nom de la dimension et sa valeur est la valeur de la dimension.

Translation de la moyenne, de la somme SampleCount, du minimum et du maximum

Le point de données Summary permet d' CloudWatch exporter toutes ces statistiques en utilisant un seul point de données.

- `startTimeUnixNano`contient le CloudWatch `startTime`
- `timeUnixNano`contient le CloudWatch `endTime`
- `sum` contient la statistique Sum (Somme).
- `count`contient la SampleCount statistique.
- `quantile_values` contient deux objets `valueAtQuantile.value` :
  - `valueAtQuantile.quantile = 0.0` avec `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` avec `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` avec `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` avec `valueAtQuantile.value = Max value`

Les ressources qui consomment le flux métrique peuvent calculer la statistique moyenne sous la forme SampleCountSomme/.

Traduction d'unités

CloudWatch les unités sont mappées selon la variante distinguant majuscules et minuscules du code unifié pour les unités de mesure, comme indiqué dans le tableau suivant. Pour de plus amples informations, veuillez consulter le Code unifié des unités de mesure.

CloudWatch	OpenTelemetry
Seconde	s
Seconde ou secondes	s

CloudWatch	OpenTelemetry
Microsecondes	us
Millisecondes	ms
Octets	Bit
Kilooctets	Ko
Mégaoctets	Mo
Gigaoctets	Go
Teraoctets	To
Bits	bit
Kilobits	Kb
Megabits	Mbps
Gigabits	GBit
Terabits	Tb
Pourcentage	%
Nombre	{Count}
Aucun	1

Les unités combinées avec une barre oblique sont mappées en appliquant la OpenTelemetry conversion des deux unités. Par exemple, Bytes/seconde est mappé à By/s.

Comment analyser les messages OpenTelemetry 1.0.0

Cette section fournit des informations pour vous aider à démarrer avec l'analyse de la OpenTelemetry version 1.0.0.

Tout d'abord, vous devez obtenir des liaisons spécifiques à la langue, qui vous permettent d'analyser les messages OpenTelemetry 1.0.0 dans votre langue préférée.

Pour obtenir des liaisons spécifiques à une langue

- Les étapes à suivre dépendent de votre langue préférée.
  - Pour utiliser Java, ajoutez la dépendance Maven suivante à votre projet Java : OpenTelemetry Java >> 0.14.1.
  - Pour utiliser une autre langue, procédez comme suit :

    a. Vérifiez que votre langue est prise en charge en consultant la liste à l'adresseGénérer vos classes.

    b. Installez le compilateur Protobuf en suivant les étapes indiquées dans Télécharger les tampons de protocole.

    c. Téléchargez les ProtoBuf définitions OpenTelemetry 0.7.0 dans la version 1.0.0.

    d. Vérifiez que vous vous trouvez dans le dossier racine des ProtoBuf définitions OpenTelemetry 0.7.0 téléchargées. Créez ensuite un dossier `src` puis exécutez la commande pour générer des liaisons spécifiques à la langue. Pour de plus amples informations, veuillez consulter Générer vos classes.

    Voici un exemple qui montre comment générer des liaisons Javascript.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

La section suivante présente des exemples d'utilisation des liaisons spécifiques à la langue que vous pouvez créer à l'aide des instructions précédentes.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
```

```
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

JavaScript

Cet exemple suppose que le dossier racine avec les liaisons générées est `./`

L'argument de données de la fonction `parseRecord` peut avoir l'un des types suivants :

- `Uint8Array` c'est optimal
- `Buffer` optimal sous le nœud
- `Array.`*`number`* entier 8 bits

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')
```

```
function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength

        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

Vous devez lire les délimiteurs `var-int` vous-même ou utilisez les méthodes internes `_VarintBytes(size)` et `_DecodeVarint32(buffer, position)`. Ceux-ci retournent la position dans le tampon juste après les octets de taille. Le côté lecture construit un nouveau tampon qui est limité à la lecture uniquement des octets du message.

```
size = my_metric.ByteSize()
```

```
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Utilisez `Buffer.DecodeMessage()`.

C#

Utilisez `CodedInputStream`. Cette classe peut lire des messages délimités par la taille.

C++

Les fonctions décrites dans `google/protobuf/util/delimited_message_util.h` peuvent lire des messages délimités par la taille.

Autres langages

Pour d'autres langues, consultez Télécharger les tampons de protocole.

Lors de l'implémentation de l'analyseur, considérez qu'un registre Kinesis peut contenir plusieurs messages des tampons de protocole `ExportMetricsServiceRequest`, chacun d'entre eux commençant par un en-tête avec un objet `UnsignedVarInt32` indiquant la longueur de l'enregistrement en octets.

OpenTelemetry Format 0.7.0

OpenTelemetry est un ensemble d'outils, d'API et de SDK. Vous pouvez l'utiliser pour instrumenter, générer, collecter et exporter des données de télémétrie (métriques, journaux et traces) à des fins d'analyse. OpenTelemetry fait partie de la Cloud Native Computing Foundation. Pour plus d'informations, consultez OpenTelemetry.

Pour plus d'informations sur la spécification OpenTelemetry 0.7.0 complète, voir la version 0.7.0.

Un enregistrement Kinesis peut contenir une ou plusieurs structures de `ExportMetricsServiceRequest` OpenTelemetry données. Chaque structure de données

commence par un en-tête avec un `UnsignedVarInt32` indiquant la longueur du registre en octets. Chaque `ExportMetricsServiceRequest` peut contenir des données provenant de plusieurs métriques à la fois.

Ce qui suit est une représentation sous forme de chaîne du message de la structure de `ExportMetricsServiceRequest` OpenTelemetry données. OpenTelemetry sérialise le protocole binaire Google Protocol Buffers, ce qui n'est pas lisible par l'homme.

```
resource_metrics {
  resource {
    attributes {
      key: "cloud.provider"
      value {
        string_value: "aws"
      }
    }
    attributes {
      key: "cloud.account.id"
      value {
        string_value: "2345678901"
      }
    }
    attributes {
      key: "cloud.region"
      value {
        string_value: "us-east-1"
      }
    }
    attributes {
      key: "aws.exporter.arn"
      value {
        string_value: "arn:aws:cloudwatch:us-east-1:123456789012:metric-stream/
MyMetricStream"
      }
    }
  }
  instrumentation_library_metrics {
    metrics {
      name: "amazonaws.com/AWS/DynamoDB/ConsumedReadCapacityUnits"
      unit: "1"
      double_summary {
        data_points {
          labels {
```

```
            key: "Namespace"
            value: "AWS/DynamoDB"
          }
          labels {
            key: "MetricName"
            value: "ConsumedReadCapacityUnits"
          }
          labels {
            key: "TableName"
            value: "MyTable"
          }
          start_time_unix_nano: 1604948400000000000
          time_unix_nano: 1604948460000000000
          count: 1
          sum: 1.0
          quantile_values {
            quantile: 0.0
            value: 1.0
          }
          quantile_values {
            quantile: 0.95
            value: 1.0
          }
          quantile_values {
            quantile: 0.99
            value: 1.0
          }
          quantile_values {
            quantile: 1.0
            value: 1.0
          }
        }
        data_points {
          labels {
            key: "Namespace"
            value: "AWS/DynamoDB"
          }
          labels {
            key: "MetricName"
            value: "ConsumedReadCapacityUnits"
          }
          labels {
            key: "TableName"
            value: "MyTable"
```

```
          }
          start_time_unix_nano: 1604948460000000000
          time_unix_nano: 1604948520000000000
          count: 2
          sum: 5.0
          quantile_values {
            quantile: 0.0
            value: 2.0
          }
          quantile_values {
            quantile: 1.0
            value: 3.0
          }
        }
      }
    }
  }
}
```

Objet de haut niveau pour sérialiser les données métriques OpenTelemetry

`ExportMetricsServiceRequest`est le wrapper de haut niveau pour sérialiser la charge utile d'un OpenTelemetry exportateur. Il contient une ou plusieurs `ResourceMetrics`.

```
message ExportMetricsServiceRequest {
  // An array of ResourceMetrics.
  // For data coming from a single resource this array will typically contain one
  // element. Intermediary nodes (such as OpenTelemetry Collector) that receive
  // data from multiple origins typically batch the data before forwarding further and
  // in that case this array will contain multiple elements.
  repeated opentelemetry.proto.metrics.v1.ResourceMetrics resource_metrics = 1;
}
```

`ResourceMetrics`est l'objet de niveau supérieur destiné à représenter les MetricData objets.

```
// A collection of InstrumentationLibraryMetrics from a Resource.
message ResourceMetrics {
  // The resource for the metrics in this message.
  // If this field is not set then no resource info is known.
  opentelemetry.proto.resource.v1.Resource resource = 1;

  // A list of metrics that originate from a resource.
  repeated InstrumentationLibraryMetrics instrumentation_library_metrics = 2;
```

```
}
```

L'objet Ressource

`Resource` est un objet de paire de valeurs qui contient des informations sur la ressource qui a généré les métriques. Pour les métriques créées par AWS, la structure de données contient l'Amazon Resource Name (ARN) de la ressource liée à la métrique, telle qu'une instance EC2 ou un compartiment S3.

L'objet `Resource` contient un attribut appelé `attributes`, qui stocke une liste des paires de valeurs clés.

- `cloud.account.id` contient l'ID de compte
- `cloud.region` contient la région
- `aws.exporter.arn` contient l'ARN du flux de métriques
- `cloud.provider` est toujours `aws`.

```
// Resource information.
message Resource {
  // Set of labels that describe the resource.
  repeated opentelemetry.proto.common.v1.KeyValue attributes = 1;

  // dropped_attributes_count is the number of dropped attributes. If the value is 0,
  // no attributes were dropped.
  uint32 dropped_attributes_count = 2;
}
```

L' InstrumentationLibraryMetrics objet

Le champ instrumentation_library ne sera pas rempli. Nous ne remplirons que le champ de métriques que nous exportons.

```
// A collection of Metrics produced by an InstrumentationLibrary.
message InstrumentationLibraryMetrics {
  // The instrumentation library information for the metrics in this message.
  // If this field is not set then no library info is known.
  opentelemetry.proto.common.v1.InstrumentationLibrary instrumentation_library = 1;
  // A list of metrics that originate from an instrumentation library.
  repeated Metric metrics = 2;
```

```
}
```

L'objet Metric

L'objet de métrique contient un champ de données `DoubleSummary` qui contient une liste de `DoubleSummaryDataPoint`.

```
message Metric {
  // name of the metric, including its DNS name prefix. It must be unique.
  string name = 1;

  // description of the metric, which can be used in documentation.
  string description = 2;

  // unit in which the metric value is reported. Follows the format
  // described by http://unitsofmeasure.org/ucum.html.
  string unit = 3;

  oneof data {
    IntGauge int_gauge = 4;
    DoubleGauge double_gauge = 5;
    IntSum int_sum = 6;
    DoubleSum double_sum = 7;
    IntHistogram int_histogram = 8;
    DoubleHistogram double_histogram = 9;
    DoubleSummary double_summary = 11;
  }
}

message DoubleSummary {
  repeated DoubleSummaryDataPoint data_points = 1;
}
```

L' MetricDescriptor objet

L' MetricDescriptor objet contient des métadonnées. Pour plus d'informations, consultez metrics.proto sur. GitHub

Pour les flux métriques, le contenu MetricDescriptor est le suivant :

- `name` sera `amazonaws.com/`*`metric_namespace`*`/`*`metric_name`*
- `description` sera vide.

- `unit` sera rempli en mappant l'unité de référence de la métrique à la variante sensible à la casse du code unifié pour les unités de mesure. Pour de plus amples informations, veuillez consulter Traductions au OpenTelemetry format 0.7.0 and le Code unifié des unités de mesure.
- `type` sera `SUMMARY`.

L' DoubleSummaryDataPoint objet

L' DoubleSummaryDataPoint objet contient la valeur d'un point de données unique dans une série chronologique dans une DoubleSummary métrique.

```
// DoubleSummaryDataPoint is a single data point in a timeseries that describes the
// time-varying values of a Summary metric.
message DoubleSummaryDataPoint {
  // The set of labels that uniquely identify this timeseries.
  repeated opentelemetry.proto.common.v1.StringKeyValue labels = 1;

  // start_time_unix_nano is the last time when the aggregation value was reset
  // to "zero". For some metric types this is ignored, see data types for more
  // details.
  //
  // The aggregation value is over the time interval (start_time_unix_nano,
  // time_unix_nano].
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  //
  // Value of 0 indicates that the timestamp is unspecified. In that case the
  // timestamp may be decided by the backend.
  fixed64 start_time_unix_nano = 2;

  // time_unix_nano is the moment when this aggregation value was reported.
  //
  // Value is UNIX Epoch time in nanoseconds since 00:00:00 UTC on 1 January
  // 1970.
  fixed64 time_unix_nano = 3;

  // count is the number of values in the population. Must be non-negative.
  fixed64 count = 4;

  // sum of the values in the population. If count is zero then this field
  // must be zero.
  double sum = 5;
```

```
  // Represents the value at a given quantile of a distribution.
  //
  // To record Min and Max values following conventions are used:
  // - The 1.0 quantile is equivalent to the maximum value observed.
  // - The 0.0 quantile is equivalent to the minimum value observed.
  message ValueAtQuantile {
    // The quantile of a distribution. Must be in the interval
    // [0.0, 1.0].
    double quantile = 1;

    // The value at the given quantile of a distribution.
    double value = 2;
  }

  // (Optional) list of values at different quantiles of the distribution calculated
  // from the current snapshot. The quantiles must be strictly increasing.
  repeated ValueAtQuantile quantile_values = 6;
}
```

Pour plus d'informations, consultez Traductions au OpenTelemetry format 0.7.0.

Traductions au OpenTelemetry format 0.7.0

CloudWatch effectue certaines transformations pour mettre CloudWatch les données en OpenTelemetry format.

Traduction de l'espace de noms, du nom de la métrique et des dimensions

Ces attributs sont des paires de valeurs clés codées dans le mappage.

- Une paire contient l'espace de noms de la métrique
- Une paire contient le nom de la métrique
- Pour chaque dimension, CloudWatch stocke la paire suivante : `metricDatum.Dimensions[i].`*`Name`*`, metricDatum.Dimensions[i].`*`Value`*

Translation de la moyenne, de la somme SampleCount, du minimum et du maximum

Le point de données Summary permet d' CloudWatch exporter toutes ces statistiques en utilisant un seul point de données.

- `startTimeUnixNano`contient le CloudWatch `startTime`

- `timeUnixNano`contient le CloudWatch `endTime`
- `sum` contient la statistique Sum (Somme).
- `count`contient la SampleCount statistique.
- `quantile_values` contient deux objets `valueAtQuantile.value` :
  - `valueAtQuantile.quantile = 0.0` avec `valueAtQuantile.value = Min value`
  - `valueAtQuantile.quantile = 0.99` avec `valueAtQuantile.value = p99 value`
  - `valueAtQuantile.quantile = 0.999` avec `valueAtQuantile.value = p99.9 value`
  - `valueAtQuantile.quantile = 1.0` avec `valueAtQuantile.value = Max value`

Les ressources qui consomment le flux métrique peuvent calculer la statistique moyenne sous la forme SampleCountSomme/.

Traduction d'unités

CloudWatch les unités sont mappées selon la variante distinguant majuscules et minuscules du code unifié pour les unités de mesure, comme indiqué dans le tableau suivant. Pour de plus amples informations, veuillez consulter le Code unifié des unités de mesure.

CloudWatch	OpenTelemetry
Seconde	s
Seconde ou secondes	s
Microsecondes	us
Millisecondes	ms
Octets	Bit
Kilooctets	Ko
Mégaoctets	Mo
Gigaoctets	Go
Teraoctets	To

CloudWatch	OpenTelemetry
Bits	bit
Kilobits	Kb
Megabits	Mbps
Gigabits	GBit
Terabits	Tb
Pourcentage	%
Nombre	{Count}
Aucun	1

Les unités combinées avec une barre oblique sont mappées en appliquant la OpenTelemetry conversion des deux unités. Par exemple, Bytes/seconde est mappé à By/s.

Comment analyser les messages de la version OpenTelemetry 0.7.0

Cette section fournit des informations pour vous aider à démarrer avec l'analyse de la version OpenTelemetry 0.7.0.

Tout d'abord, vous devez obtenir des liaisons spécifiques à la langue, qui vous permettent d'analyser les messages de la OpenTelemetry version 0.7.0 dans votre langue préférée.

Pour obtenir des liaisons spécifiques à une langue

- Les étapes à suivre dépendent de votre langue préférée.
  - Pour utiliser Java, ajoutez la dépendance Maven suivante à votre projet Java : OpenTelemetry Java >> 0.14.1.
  - Pour utiliser une autre langue, procédez comme suit :

    a. Vérifiez que votre langue est prise en charge en consultant la liste à l'adresseGénérer vos classes.

    b. Installez le compilateur Protobuf en suivant les étapes indiquées dans Télécharger les tampons de protocole.

c. Téléchargez les ProtoBuf définitions OpenTelemetry 0.7.0 dans la version 0.7.0.

d. Vérifiez que vous vous trouvez dans le dossier racine des ProtoBuf définitions OpenTelemetry 0.7.0 téléchargées. Créez ensuite un dossier `src` puis exécutez la commande pour générer des liaisons spécifiques à la langue. Pour de plus amples informations, veuillez consulter Générer vos classes.

   Voici un exemple qui montre comment générer des liaisons Javascript.

```
protoc --proto_path=./ --js_out=import_style=commonjs,binary:src \
opentelemetry/proto/common/v1/common.proto \
opentelemetry/proto/resource/v1/resource.proto \
opentelemetry/proto/metrics/v1/metrics.proto \
opentelemetry/proto/collector/metrics/v1/metrics_service.proto
```

La section suivante présente des exemples d'utilisation des liaisons spécifiques à la langue que vous pouvez créer à l'aide des instructions précédentes.

Java

```
package com.example;

import io.opentelemetry.proto.collector.metrics.v1.ExportMetricsServiceRequest;

import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

public class MyOpenTelemetryParser {

    public List<ExportMetricsServiceRequest> parse(InputStream inputStream) throws
 IOException {
        List<ExportMetricsServiceRequest> result = new ArrayList<>();

        ExportMetricsServiceRequest request;
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
```

```
         */
        while ((request =
 ExportMetricsServiceRequest.parseDelimitedFrom(inputStream)) != null) {
            // Do whatever we want with the parsed message
            result.add(request);
        }

        return result;
    }
}
```

JavaScript

Cet exemple suppose que le dossier racine avec les liaisons générées est `./`

L'argument de données de la fonction `parseRecord` peut avoir l'un des types suivants :

- `Uint8Array` c'est optimal
- `Buffer` optimal sous le nœud
- `Array.number` entier 8 bits

```
const pb = require('google-protobuf')
const pbMetrics =
    require('./opentelemetry/proto/collector/metrics/v1/metrics_service_pb')

function parseRecord(data) {
    const result = []

    // Loop until we've read all the data from the buffer
    while (data.length) {
        /* A Kinesis record can contain multiple `ExportMetricsServiceRequest`
           records, each of them starting with a header with an
           UnsignedVarInt32 indicating the record length in bytes:
            ------ -------------------------- ------ ----------------------
           |UINT32|ExportMetricsServiceRequest|UINT32|ExportMetricsService...
            ------ -------------------------- ------ ----------------------
         */
        const reader = new pb.BinaryReader(data)
        const messageLength = reader.decoder_.readUnsignedVarint32()
        const messageFrom = reader.decoder_.cursor_
        const messageTo = messageFrom + messageLength
```

```
        // Extract the current `ExportMetricsServiceRequest` message to parse
        const message = data.subarray(messageFrom, messageTo)

        // Parse the current message using the ProtoBuf library
        const parsed =
            pbMetrics.ExportMetricsServiceRequest.deserializeBinary(message)

        // Do whatever we want with the parsed message
        result.push(parsed.toObject())

        // Shrink the remaining buffer, removing the already parsed data
        data = data.subarray(messageTo)
    }

    return result
}
```

Python

Vous devez lire les délimiteurs `var-int` vous-même ou utilisez les méthodes internes `_VarintBytes(size)` et `_DecodeVarint32(buffer, position)`. Ceux-ci retournent la position dans le tampon juste après les octets de taille. Le côté lecture construit un nouveau tampon qui est limité à la lecture uniquement des octets du message.

```
size = my_metric.ByteSize()
f.write(_VarintBytes(size))
f.write(my_metric.SerializeToString())
msg_len, new_pos = _DecodeVarint32(buf, 0)
msg_buf = buf[new_pos:new_pos+msg_len]
request = metrics_service_pb.ExportMetricsServiceRequest()
request.ParseFromString(msg_buf)
```

Go

Utilisez `Buffer.DecodeMessage()`.

C#

Utilisez `CodedInputStream`. Cette classe peut lire des messages délimités par la taille.

C++

Les fonctions décrites dans `google/protobuf/util/delimited_message_util.h` peuvent lire des messages délimités par la taille.

Autres langages

Pour d'autres langues, consultez Télécharger les tampons de protocole.

Lors de l'implémentation de l'analyseur, considérez qu'un registre Kinesis peut contenir plusieurs messages des tampons de protocole `ExportMetricsServiceRequest`, chacun d'entre eux commençant par un en-tête avec un objet `UnsignedVarInt32` indiquant la longueur de l'enregistrement en octets.

Résolution des problèmes

Si vous ne voyez pas les données de métriques à votre destination finale, vérifiez les points suivants :

- Vérifiez que le flux de métriques est en cours d'exécution. Pour savoir comment utiliser la CloudWatch console à cette fin, consultezExploitation et entretien des flux métriques.
- Les statistiques publiées il y a plus de deux jours ne sont pas diffusées en continu. Pour déterminer si une métrique particulière sera diffusée, tracez le graphique de la métrique dans la CloudWatch console et vérifiez l'âge du dernier point de données visible. Si cela fait plus de deux jours, il ne sera pas détecté par les flux métriques.
- Vérifiez les métriques émises par le flux de métriques. Dans la CloudWatch console, sous Metrics, examinez l'espace de MetricStreams noms CloudWatchAWS//pour les métriques MetricUpdateTotalMetricUpdate, et PublishErrorRate.
- Si la PublishErrorRatemétrique est élevée, vérifiez que la destination utilisée par le flux de diffusion Firehose existe et que le rôle IAM spécifié dans la configuration du flux métrique accorde au `CloudWatch` service principal les autorisations d'écriture dans ce flux. Pour plus d’informations, consultez Confiance entre Firehose CloudWatch et Firehose.
- Vérifiez que le flux de diffusion Firehose est autorisé à écrire vers la destination finale.
- Dans la console Firehose, consultez le flux de diffusion Firehose utilisé pour le flux métrique et vérifiez dans l'onglet Monitoring si le flux de diffusion Firehose reçoit des données.
- Vérifiez que vous avez configuré votre flux de diffusion Firehose avec les informations correctes.
- Vérifiez tous les journaux ou métriques disponibles pour la destination finale vers laquelle le flux de diffusion Firehose écrit.

- Pour obtenir des informations plus détaillées, activez la journalisation CloudWatch des erreurs dans les journaux sur le flux de diffusion Firehose. Pour plus d'informations, consultez la section Surveillance d'Amazon Data Firehose à l'aide CloudWatch de journaux.

Affichage des métriques disponibles

Les métriques sont d'abord regroupées par espace de noms, puis par les différentes combinaisons de dimension au sein de chaque espace de noms. Par exemple, vous pouvez afficher toutes les métriques EC2, des métriques EC2 regroupées par instance ou des métriques EC2 regroupées par groupe Auto Scaling.

Seuls les AWS services que vous utilisez envoient des statistiques à Amazon CloudWatch.

Pour obtenir la liste des AWS services auxquels des métriques sont envoyées CloudWatch, consultezAWS services qui publient CloudWatch des statistiques. À partir de cette page, vous pouvez également consulter les métriques et les dimensions publiées par chacun de ces services.

> **Note**
>
> Les métriques qui n'ont pas eu de nouveaux points de données au cours des deux dernières semaines n'apparaissent pas dans la console. Ils n'apparaissent pas non plus lorsque vous tapez leur nom de métrique ou leur nom de dimension dans la zone de recherche de l'onglet All metrics (Toutes les métriques) de la console, et ils ne sont pas renvoyés dans les résultats d'une commande list-metrics . La meilleure façon de récupérer ces métriques est d'utiliser les get-metric-statisticscommandes get-metric-dataor du AWS CLI.
> Si l'ancienne métrique que vous souhaitez afficher possède une métrique actuelle avec des dimensions similaires, vous pouvez afficher cette métrique actuelle similaire, choisir l'onglet Source et remplacer le nom de la métrique et les champs de dimension par ceux de votre choix. Vous pouvez également modifier la plage temporelle en une période où la métrique a été signalée.

Les étapes suivantes vous aident à parcourir les espaces de noms de métriques pour rechercher et afficher des métriques. Vous pouvez également rechercher des mesures à l'aide de termes de recherche ciblés. Pour plus d'informations, consultez Rechercher des métriques disponibles.

Si vous naviguez sur un compte configuré en tant que compte de surveillance dans le cadre de l'observabilité CloudWatch entre comptes, vous pouvez consulter les statistiques des comptes

sources associés à ce compte de surveillance. Lorsque les métriques des comptes sources sont affichées, l'ID ou l'étiquette du compte dont elles proviennent est également affiché. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

Afficher des métriques disponibles par espace de noms et dimension à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Sélectionnez un espace de noms de métrique (par exemple, EC2 ou Lambda).
4. Sélectionnez une dimension de métrique (par exemple, Per-Instance Metrics (Métriques par instance) ou By Function Name (Par nom de fonction)).
5. L'onglet Browse (Parcourir) affiche toutes les métriques pour cette dimension dans l'espace de noms. À côté de chaque nom de métrique se trouve un bouton d'information que vous pouvez choisir pour afficher une fenêtre contextuelle avec la définition de la métrique.

   S'il s'agit d'un compte de surveillance dans CloudWatch le domaine de l'observabilité entre comptes, vous pouvez également consulter les statistiques des comptes sources liés à ce compte de surveillance. Les colonnes Account label (Étiquette du compte) et Account id (ID du compte) de la table indiquent de quel compte provient chaque métrique.

   Vous pouvez effectuer les actions suivantes :

   a. Pour trier le tableau, utilisez l'en-tête de colonne.
   b. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d'en-tête du tableau.
   c. Pour filtrer par compte, choisissez l'étiquette du compte ou l'ID du compte, puis Add to search (Ajouter à la recherche).
   d. Pour filtrer par ressource, sélectionnez l'ID de ressource, puis Add to search.
   e. Pour filtrer par métrique, choisissez le nom de la métrique, puis Add to search (Ajouter à la recherche).
6. (Facultatif) Pour ajouter ce graphique à un CloudWatch tableau de bord, choisissez Actions, puis Ajouter au tableau de bord.

Pour afficher les métriques disponibles par espace de noms, dimension ou métrique du compte à l'aide du AWS CLI

Utilisez la commande list-metrics pour CloudWatch répertorier les métriques. Pour obtenir la liste des espaces de noms, des métriques et des dimensions de tous les services qui publient des métriques, consultez AWS services qui publient CloudWatch des statistiques.

L'exemple de commande suivant répertorie toutes les métriques pour Amazon EC2.

```
aws cloudwatch list-metrics --namespace AWS/EC2
```

Voici un exemple de sortie.

```
{
  "Metrics" : [
    ...
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkOut"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "CPUUtilization"
    },
    {
        "Namespace": "AWS/EC2",
        "Dimensions": [
            {
                "Name": "InstanceId",
                "Value": "i-1234567890abcdef0"
            }
        ],
        "MetricName": "NetworkIn"
    },
```

```
    ...
  ]
}
```

Répertorier toutes les métriques disponibles pour une ressource spécifiée

L'exemple suivant spécifie l'espace de noms `AWS/EC2` et la dimension `InstanceId` pour afficher les résultats uniquement pour l'instance spécifiée.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --dimensions
 Name=InstanceId,Value=i-1234567890abcdef0
```

Répertorier une métrique pour toutes les ressources

L'exemple suivant spécifie l'espace de noms `AWS/EC2` et un nom de métrique pour afficher les résultats uniquement pour la métrique spécifiée.

```
aws cloudwatch list-metrics --namespace AWS/EC2 --metric-name CPUUtilization
```

Pour récupérer des métriques à partir de comptes sources liés dans le cadre de l' CloudWatch observabilité entre comptes

L'exemple suivant est exécuté dans un compte de surveillance pour récupérer les métriques à la fois du compte de surveillance et de tous les comptes sources liés. Si vous n'ajoutez pas `--include-linked-accounts`, la commande renvoie uniquement les métriques du compte de surveillance.

```
aws cloudwatch list-metrics --include-linked-accounts
```

Pour récupérer les métriques d'un compte source dans le cadre de l' CloudWatch observabilité entre comptes

L'exemple suivant est exécuté dans un compte de surveillance pour récupérer les métriques du compte source avec l'ID 111122223333.

```
aws cloudwatch list-metrics --include-linked-accounts --owning-account "111122223333"
```

Rechercher des métriques disponibles

Vous pouvez effectuer des recherches dans toutes les métriques de votre compte à l'aide de termes de recherche ciblés. Les métriques renvoyées ont des résultats correspondants au sein de leur espace de noms, nom de métrique ou dimensions.

S'il s'agit d'un compte de surveillance dans CloudWatch le domaine de l'observabilité entre comptes, vous recherchez également des métriques provenant des comptes sources liés à ce compte de surveillance.

> **Note**
>
> Les métriques qui n'ont pas eu de nouveaux points de données au cours des deux dernières semaines n'apparaissent pas dans la console. Ils n'apparaissent pas non plus lorsque vous tapez leur nom de métrique ou leur nom de dimension dans la zone de recherche de l'onglet All metrics (Toutes les métriques) de la console, et ils ne sont pas renvoyés dans les résultats d'une commande list-metrics . La meilleure façon de récupérer ces métriques est d'utiliser les get-metric-statisticscommandes get-metric-dataor du AWS CLI.

Pour rechercher les métriques disponibles dans CloudWatch

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 sur l'onglet All metrics (Toutes les métriques), saisissez un terme de recherche, tel qu'un nom de métrique, un espace de noms, un ID de compte, un label de compte, un nom ou une valeur de dimension, ou un nom de ressource. Il affiche tous les espaces de noms avec des métriques pour ce terme à rechercher.

   Par exemple, si vous explorez **`volume`**, il affiche les espaces de noms qui contiennent les métriques incluant ce terme dans leur nom.

   Pour plus d'informations sur la recherche, consultez Utiliser des expressions de recherche dans les graphiques.
4. Pour représenter graphiquement tous les résultats de recherche, choisissez Graph search (Recherche de graphique)

   or

Sélectionnez un espace de noms pour afficher les métriques à partir de cet espace de noms. Vous pouvez alors effectuer ce qui suit :

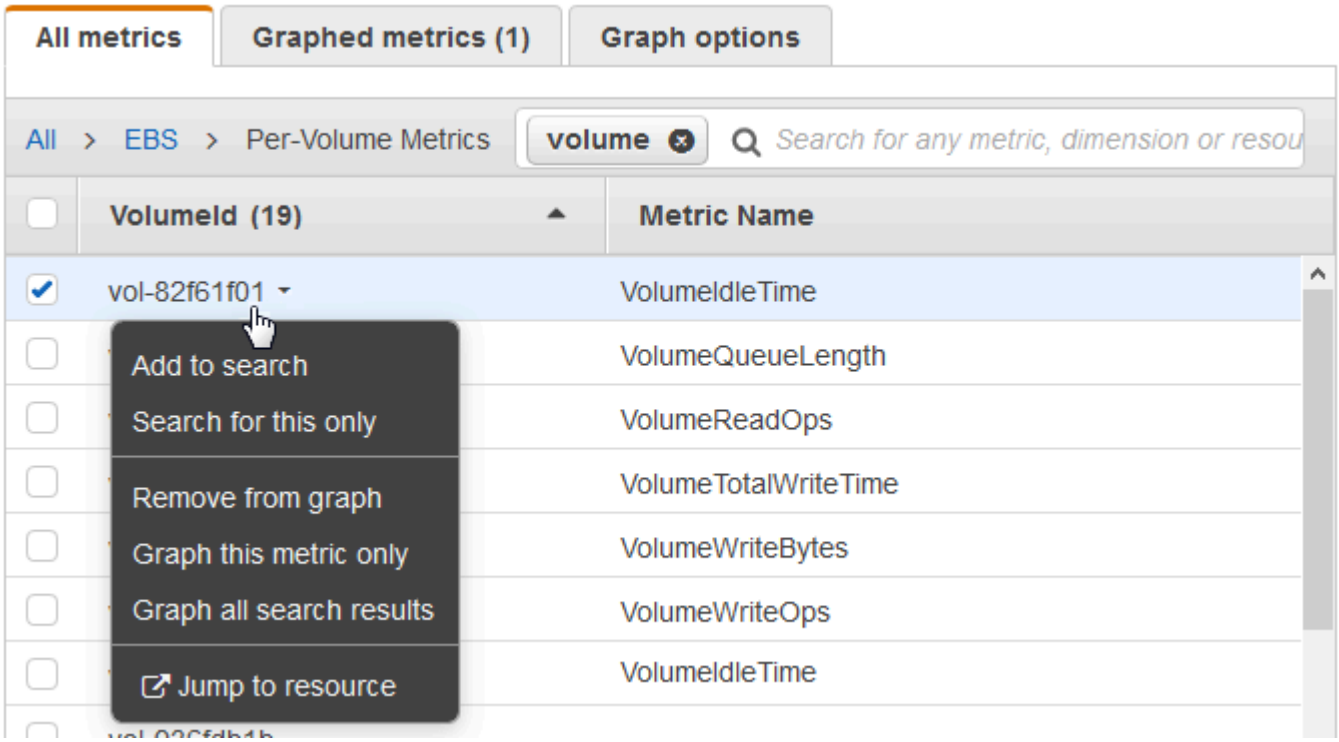
a. Pour représenter graphiquement une ou plusieurs métriques, cochez la case en regard de chaque métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d'en-tête du tableau.

b. Pour affiner votre recherche, passez la souris sur un nom de métrique et Add to search (Ajouter à la recherche) ou Search for this only (Rechercher uniquement ceci).

c. Pour afficher l'une des ressources dans sa console, sélectionnez l'ID de ressource, puis Jump to resource (Aller à la ressource).

d. Pour afficher l'aide relative à une métrique, sélectionnez le nom de la métrique, puis What is this? (De quoi s'agit-il ?).

Les métriques sélectionnées apparaissent sur le graphique.



5. (Facultatif) Sélectionnez l'un des boutons dans la barre de recherche pour modifier cette partie du terme recherché.

Graphique des métriques

Utilisez la CloudWatch console pour représenter graphiquement les données métriques générées par d'autres AWS services. Il est ainsi plus efficace de voir l'activité des métriques sur vos services. Les procédures suivantes décrivent comment représenter graphiquement les métriques dans CloudWatch.

Table des matières

Représenter graphiquement une métrique

Vous pouvez sélectionner des métriques et créer des graphiques des données métriques à l'aide de la CloudWatch console.

CloudWatch prend en charge les statistiques suivantes sur les métriques : `Average`,`Minimum`,`Maximum`,`Sum`, et`SampleCount`. Pour plus d'informations, consultez Statistiques.

Vous pouvez afficher vos données dans différents niveaux de détails. Par exemple, vous pouvez choisir une vue d'une minute, ce qui peut être utile lors d'un dépannage. Vous pouvez également choisir une vue moins détaillée, d'une heure par exemple. Cela peut être utile lors de l'affichage d'une plage de temps plus large (par exemple, 3 jours), afin de voir les tendances au fil du temps. Pour plus d'informations, consultez Périodes.

Si vous utilisez un compte configuré en tant que compte de surveillance dans le cadre de l'observabilité CloudWatch entre comptes, vous pouvez représenter graphiquement les métriques des comptes sources liés à ce compte de surveillance. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

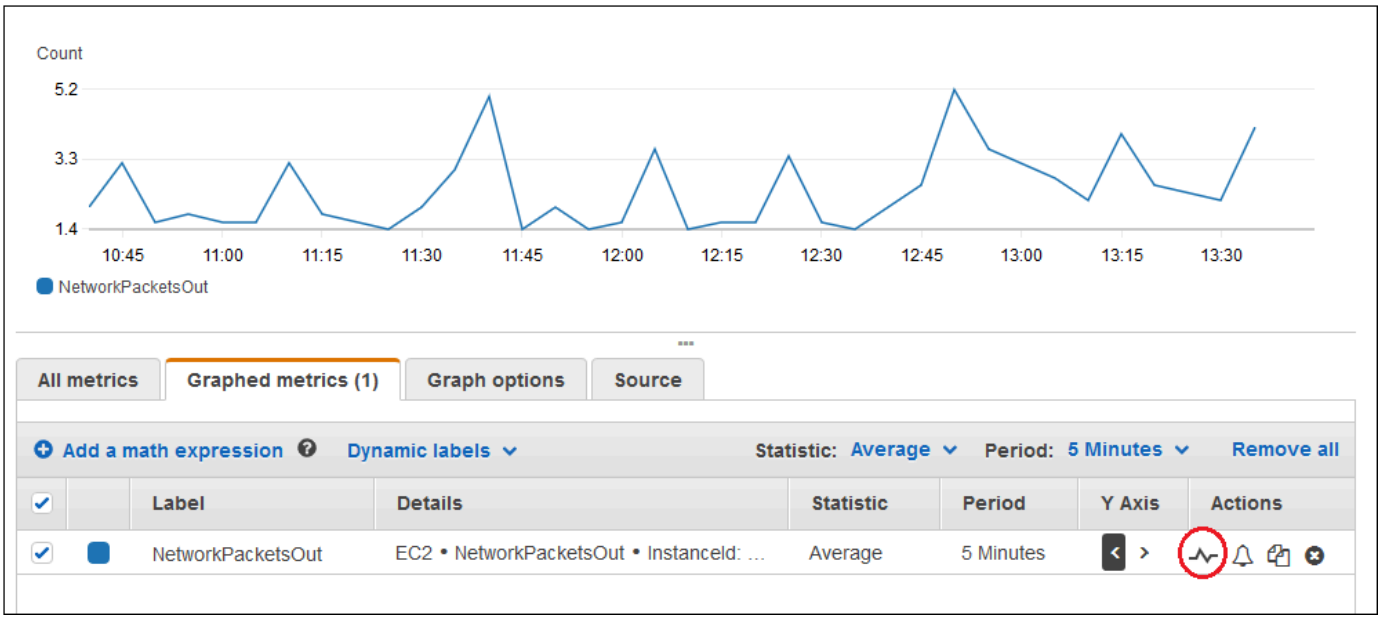
Création d'un graphique

Pour représenter graphiquement une métrique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Dans l'onglet Parcourir, saisissez un terme de recherche dans le champ de recherche, tel qu'un nom de métrique, un ID de compte ou un nom de ressource.

   Par exemple, si vous effectuez une recherche pour la métrique `CPUUtilization`, les espaces de noms et les dimensions associés à cette métrique s'affichent.
4. Sélectionnez un des résultats de votre recherche pour afficher les métriques.
5. Pour représenter graphiquement une ou plusieurs métriques, cochez la case en regard de chaque métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d'en-tête du tableau.
6. (Facultatif) Pour modifier le type de graphique, sélectionnez l'onglet Options. Vous pouvez ensuite choisir entre un graphique linéaire, un graphique à aires empilées, un nombre, une jauge, un graphique à barres ou un graphique à secteurs.
7. Sélectionnez l'onglet Graphed metrics (Graphiques des métriques).
8. (Facultatif) Pour modifier la statistique utilisée dans le graphique, choisissez la nouvelle statistique dans la zone Statistic (Statistique) en regard du nom de la métrique.

   Pour plus d'informations sur CloudWatch les statistiques, consultezCloudWatch définitions des statistiques. Pour de plus amples informations sur les statistiques de centile pxx, consultez Centiles.
9. (Facultatif) Pour ajouter une bande de détection d'anomalies qui indique les valeurs attendues pour la métrique, choisissez l'icône de détection d'anomalies sous Actions en regard de la métrique.



CloudWatch utilise jusqu'à deux semaines de données historiques récentes de la métrique pour calculer un modèle pour les valeurs attendues. Il affiche ensuite cette plage de valeurs attendues sous forme de bande sur le graphique. CloudWatch ajoute une nouvelle ligne sous la métrique pour afficher l'expression mathématique du canal de détection des anomalies, intitulée ANOMALY_DETECTION_BAND. Si des données d'historique récentes existent, vous voyez immédiatement un aperçu de la bande de détection d'anomalies, qui est une approximation de la bande de détection d'anomalies générée par le modèle. L'affichage de la bande de détection d'anomalies réelle peut prendre jusqu'à 15 minutes.

Par défaut, CloudWatch crée les limites supérieure et inférieure de la bande de valeurs attendues avec une valeur par défaut de 2 pour le seuil de bande. Pour modifier ce nombre, modifiez la valeur à la fin de la formule sous Details (Détails) pour le groupe.

- (Facultatif) Choisissez Edit model (Modifier un modèle) pour modifier la manière dont le modèle de détection d'anomalie est calculé. Vous pouvez exclure l'utilisation des périodes passées et futures de l'apprentissage pour le calcul du modèle. Il est essentiel d'exclure des données d'apprentissage les événements inhabituels tels que les pannes du système, les déploiements et les périodes de congés. Vous pouvez également spécifier le fuseau horaire à utiliser dans le modèle pour les modifications relatives à l'heure d'été.

Pour plus d'informations, consultez Utilisation de la détection des CloudWatch anomalies.

Pour masquer le modèle du graphique, décochez la ligne avec la fonction `ANOMALY_DETECTION_BAND` ou choisissez l'icône `X`. Pour supprimer entièrement le modèle, choisissez Edit model (Modifier le modèle), puis Delete model (Supprimer le modèle).

10. (Facultatif) Lorsque vous choisissez des métriques à représenter par graphique, spécifiez une étiquette dynamique à faire apparaître sur la légende du graphique pour chaque métrique. Les étiquettes dynamiques affichent une statistique sur la métrique et se mettent automatiquement à jour lorsque le tableau de bord ou le graphique est actualisé. Pour ajouter une étiquette dynamique, choisissez Graphique des métriques, puis Ajouter une étiquette dynamique.

    Par défaut, les valeurs dynamiques que vous ajoutez à l'étiquette apparaissent au début de l'étiquette. Vous pouvez ensuite sélectionner la valeur Label (Étiquette) de la métrique pour modifier l'étiquette. Pour plus d'informations, consultez Utiliser des étiquettes dynamiques.
11. Pour voir plus d'informations sur la métrique représentée graphiquement, passez le curseur sur la légende.
12. Les annotations horizontales peuvent aider les utilisateurs de graphique à voir de manière plus efficace le moment où une métrique présente des pics à un certain niveau, ou si la métrique se trouve dans une plage prédéfinie. Pour ajouter une annotation horizontale, choisissez l'onglet Options, puis sélectionnez Ajouter des annotations horizontales :

    a. Pour Label (Étiquette), saisissez une étiquette pour l'annotation.

    b. Pour Value (Valeur), saisissez la valeur de métrique où apparaît l'annotation horizontale.

    c. Pour Fill (Remplissage), spécifiez s'il faut utiliser un ombrage de remplissage avec cette annotation. Par exemple, choisissez `Above` ou `Below` pour que la zone correspondante soit remplie. Si vous spécifiez `Between`, un autre champ `Value` apparaît et la zone du graphique entre les deux valeurs est remplie.

    d. Pour Axis (Axe), spécifiez si les nombres dans `Value` font référence à la métrique associée à l'axe des Y de gauche ou à l'axe des Y de droite si le graphique comprend plusieurs métriques.

    Vous pouvez modifier la couleur de remplissage d'une annotation en choisissant le carré de couleur dans la colonne de gauche de cette annotation.

    Répétez ces étapes pour ajouter plusieurs annotations horizontales au même graphique.

    Pour masquer une annotation, décochez la case dans la colonne de gauche de cette annotation.

Pour supprimer une annotation, choisissez x dans la colonne Actions.

13. Pour obtenir une URL pour votre graphique, choisissez Actions, puis Share (Partager). Copiez l'URL à enregistrer ou à partager.
14. Pour ajouter votre graphique à un tableau de bord, choisissez Actions, Add to dashboard (Ajouter au tableau de bord).

Création d'un graphique de mesures à partir d'une autre source de données

Vous pouvez créer un graphique qui affiche les ressources provenant de sources de données autres que CloudWatch. Pour plus d'informations sur la création de connexions avec ces autres sources de données, veuillez consulter Interrogation de métriques d'autres sources de données.

Pour représenter graphiquement une métrique provenant d'une autre source de données

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Choisissez l'onglet Requête multisource.
4. Pour Sources de données, choisissez le nom de la source de données à utiliser.

   Si vous n'avez pas encore créé de connexion à la source de données de votre choix, sélectionnez Créer et gérer des sources de données, puis choisissez Créer et gérer des sources de données. Pour plus d'informations sur le reste de ce processus de création de source de données, veuillez consulter la rubrique Connexion à une source de données prédéfinie à l'aide d'un assistant.
5. L'assistant ou l'éditeur de requêtes vous invite à saisir les informations nécessaires à la requête. Le flux de travail est différent pour chaque source de données et est adapté à la source de données. Par exemple, pour les sources de données Amazon Managed Service for Prometheus et Prometheus, un éditeur de requêtes PromQL avec un assistant de requête apparaît.
6. Lorsque vous avez terminé de créer la requête, choisissez Requête graphique.

   Le graphique est renseigné avec les métriques issues de la requête.
7. (Facultatif) Les annotations horizontales peuvent aider les utilisateurs de graphique à voir de manière plus efficace le moment où une métrique présente des pics à un certain niveau, ou si la métrique se trouve dans une plage prédéfinie. Pour ajouter une annotation horizontale, choisissez l'onglet Options, puis sélectionnez Ajouter des annotations horizontales :

a. Pour Label (Étiquette), saisissez une étiquette pour l'annotation.

b. Pour Value (Valeur), saisissez la valeur de métrique où apparaît l'annotation horizontale.

c. Pour Fill (Remplissage), spécifiez s'il faut utiliser un ombrage de remplissage avec cette annotation. Par exemple, choisissez `Above` ou `Below` pour que la zone correspondante soit remplie. Si vous spécifiez `Between`, un autre champ `Value` apparaît et la zone du graphique entre les deux valeurs est remplie.

d. Pour Axis (Axe), spécifiez si les nombres dans `Value` font référence à la métrique associée à l'axe des Y de gauche ou à l'axe des Y de droite si le graphique comprend plusieurs métriques.

Vous pouvez modifier la couleur de remplissage d'une annotation en choisissant le carré de couleur dans la colonne de gauche de cette annotation.

Répétez ces étapes pour ajouter plusieurs annotations horizontales au même graphique.

Pour masquer une annotation, décochez la case dans la colonne de gauche de cette annotation.

Pour supprimer une annotation, choisissez x dans la colonne Actions.

8. (Facultatif) Pour ajouter ce graphique à un tableau de bord CloudWatch, choisissez Actions, puis Ajouter au tableau de bord.

Mise à jour d'un graphique

Mettre à jour votre graphique

1. Pour modifier le nom du graphique, choisissez l'icône représentant un crayon.
2. Pour modifier la plage de temps, sélectionnez l'une des valeurs prédéfinies ou choisissez custom (personnalisé). Pour plus d’informations, consultez Modifier la plage horaire ou le format du fuseau horaire d'un graphique.
3. Pour modifier les statistiques, choisissez l'onglet Graphed metrics (Graphique des métriques). Choisissez l'en-tête de colonne ou une valeur individuelle, puis choisissez l'une des statistiques ou des centiles prédéfinis, ou bien spécifiez un centile personnalisé (par exemple, **`p95.45`**).
4. Pour modifier la période, choisissez l'onglet Graphed metrics (Graphique des métriques). Sélectionnez l'en-tête de colonne ou une valeur individuelle et choisissez une autre valeur.

5. Pour ajouter une annotation horizontale, choisissez Graph options (Options de graphique), puis Add horizontal annotation (Ajouter une annotation horizontale) :

   a. Pour Label (Étiquette), saisissez une étiquette pour l'annotation.

   b. Pour Value (Valeur), saisissez la valeur de métrique où apparaît l'annotation horizontale.

   c. Pour Fill (Remplissage), spécifiez s'il faut utiliser un ombrage de remplissage avec cette annotation. Par exemple, choisissez `Above` ou `Below` pour que la zone correspondante soit remplie. Si vous spécifiez `Between`, un autre champ `Value` apparaît et la zone du graphique entre les deux valeurs est remplie.

   d. Pour Axis (Axe), spécifiez si les nombres dans `Value` font référence à la métrique associée à l'axe des Y de gauche ou à l'axe des Y de droite si le graphique comprend plusieurs métriques.

      Vous pouvez modifier la couleur de remplissage d'une annotation en choisissant le carré de couleur dans la colonne de gauche de cette annotation.

   Répétez ces étapes pour ajouter plusieurs annotations horizontales au même graphique.

   Pour masquer une annotation, décochez la case dans la colonne de gauche de cette annotation.

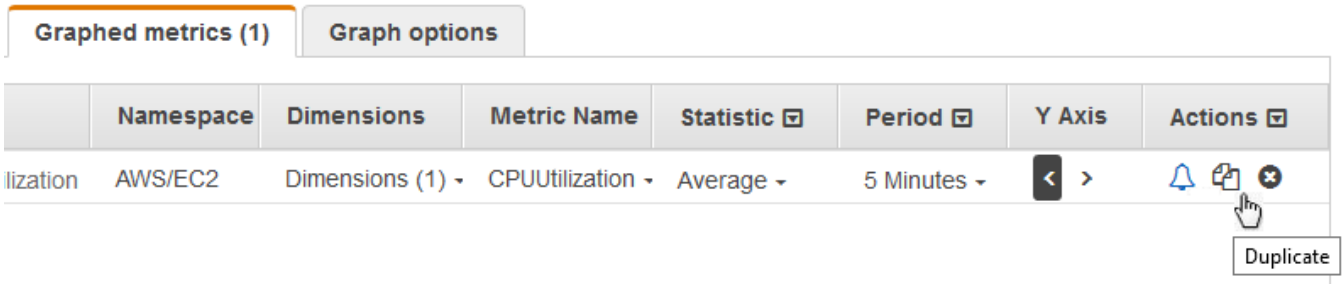
   Pour supprimer une annotation, choisissez x dans la colonne Actions.

6. Pour modifier l'intervalle d'actualisation, choisissez Refresh options (Actualiser les options), puis Auto refresh (Actualisation automatique) ou choisissez 1 Minute, 2 Minutes, 5 Minutes, ou 15 Minutes.

Duplication d'une métrique

Dupliquer une métrique

1. Sélectionnez l'onglet Graphed metrics (Graphique des métriques).
2. Pour Actions, choisissez l'icône Duplicate (Dupliquer).



3. Mettez à jour la métrique dupliquée si nécessaire.

Fusionner deux graphiques en un

Vous pouvez fusionner deux graphiques en un seul. Le graphique obtenu affiche alors les deux métriques. Cela peut être utile si différentes métriques sont déjà affichées dans différents graphiques et que vous souhaitez les combiner, ou si vous souhaitez créer facilement un seul graphique avec des métriques provenant de différentes régions.

Pour fusionner deux graphiques, vous devez utiliser l'URL ou la source JSON du graphique dans lequel vous souhaitez les fusionner.

Pour fusionner deux graphiques en un

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Ouvrez le graphique que vous souhaitez fusionner dans un autre graphique. Pour ce faire, vous pouvez sélectionner Métriques, Toutes les métriques, puis choisir une métrique à représenter graphiquement. Vous pouvez également ouvrir un tableau de bord, puis ouvrir l'un des graphiques du tableau de bord en le sélectionnant et en choisissant Ouvrir dans les métriques dans le menu en haut à droite du graphique.
3. Une fois que le graphique est ouvert, effectuez l'une des actions suivantes :
   - Copiez l'URL depuis la barre du navigateur.
   - Sélectionnez l'onglet Source, puis choisissez Copier.
4. Ouvrez le graphique dans lequel vous souhaitez fusionner le précédent graphique.
5. Lorsque le deuxième graphique est ouvert dans la vue Métriques, choisissez Actions, Graphique de fusion.
6. Saisissez l'URL ou le JSON que vous avez précédemment copié, puis choisissez Fusionner.
7. Les graphiques fusionnés apparaissent. L'axe Y de gauche correspond au graphique d'origine, tandis que l'axe Y de droite correspond au graphique que vous y avez fusionné.

> **Note**
>
> Si le graphique dans lequel vous avez fusionné utilise la fonction METRICS(), les métriques du graphique fusionné ne sont pas incluses dans le calcul METRICS() du graphique fusionné.

8. Pour enregistrer le graphique fusionné à un tableau de bord, choisissez Actions, Ajouter au tableau de bord.

Utiliser des étiquettes dynamiques

Vous pouvez utiliser des étiquettes dynamiques avec vos graphiques. Les étiquettes dynamiques ajoutent une valeur mise à jour de manière dynamique à l'étiquette pour la métrique sélectionnée. Vous pouvez ajouter une large gamme de valeurs aux étiquettes, comme indiqué dans les tableaux suivants.

La valeur dynamique indiquée dans l'étiquette est dérivée de l'intervalle de temps actuellement affiché sur le graphique. Cette partie dynamique de l'étiquette se met à jour automatiquement lorsque le tableau de bord ou le graphique est actualisé.

Si vous utilisez une étiquette dynamique avec une expression de recherche, l'étiquette dynamique s'applique à toutes les métriques renvoyées par la recherche.

Vous pouvez utiliser la CloudWatch console pour ajouter une valeur dynamique à une étiquette, modifier l'étiquette, modifier la position de la valeur dynamique dans la colonne d'étiquette et effectuer d'autres personnalisations.

Étiquettes dynamiques

Dans une étiquette dynamique, vous pouvez utiliser les valeurs suivantes relatives aux propriétés de la métrique :

Valeur dynamique de l'étiquette	Description
${AVG}	Moyenne des valeurs dans l'intervalle de temps actuellement affiché sur le graphique.

Valeur dynamique de l'étiquette	Description
${DATAPOINT_COUNT}	Nombre de points de données dans l'intervalle de temps actuellement affiché sur le graphique.
${FIRST}	La plus ancienne des valeurs de métrique dans la plage de temps actuellement affichée dans le graphique.
${FIRST_LAST_RANGE}	Différence entre les valeurs de métrique des points de données les plus anciens et les plus récents qui sont actuellement affichés dans le graphique.
${FIRST_LAST_TIME_RANGE}	La plage de temps absolue entre les points de données les plus anciens et les plus récents actuellement affichés dans le graphique.
${FIRST_TIME}	Horodatage du plus ancien point de données dans l'intervalle de temps actuellement affiché sur le graphique.
${FIRST_TIME_RELATIVE}	Différence de temps absolue entre l'heure actuelle et l'horodatage du point de données le plus ancien de la plage de temps qui est actuellement affiché dans le graphique.
${LABEL}	Représentation de l'étiquette par défaut pour une métrique.
${LAST}	Valeurs de métrique les plus récentes dans l'intervalle de temps actuellement affiché sur le graphique.
${LAST_TIME}	Horodatage du plus récent point de données dans l'intervalle de temps actuellement affiché sur le graphique.
${LAST_TIME_RELATIVE}	Différence de temps absolue entre l'heure actuelle et l'horodatage du point de données le plus récent de la plage de temps qui est actuellement affiché dans le graphique.
${MAX}	Valeur maximum dans l'intervalle de temps actuellement affiché sur le graphique.

Valeur dynamique de l'étiquette	Description
${MAX_TIME}	Horodatage du point de données ayant la valeur de métrique la plus élevée, des points de données qui sont actuellement affichés dans le graphique.
${MAX_TIME_RELATIVE}	Différence de temps absolue entre l'heure actuelle et l'horodatage du point de données avec la valeur la plus élevée, de ces points de données qui sont actuellement affichés dans le graphique.
${MIN}	Valeur minimum dans l'intervalle de temps actuellement affiché sur le graphique.
${MIN_MAX_RANGE}	Différence des valeurs de métrique entre les points de données avec les valeurs de métrique les plus élevées et les plus faibles, parmi les points de données actuellement affichés dans le graphique.
${MIN_MAX_TIME_RANGE}	Plage de temps absolue entre les points de données avec les valeurs métriques les plus élevées et les plus faibles, parmi les points de données actuellement affichés dans le graphique.
${MIN_TIME}	Horodatage du point de données ayant la valeur de métrique la plus faible, des points de données qui sont actuellement affichés dans le graphique.
${MIN_TIME_RELATIVE}	Différence de temps absolue entre l'heure actuelle et l'horodatage du point de données avec la valeur la plus faible, de ces points de données qui sont actuellement affichés dans le graphique.
$ {PROP ('AccountId')}	L'ID de AWS compte de la métrique.
$ {PROP ('AccountLabel')}	L'étiquette spécifiée pour le compte source propriétaire de cette métrique, dans l'observabilité CloudWatch entre comptes.

Valeur dynamique de l'étiquette	Description
${PROP('Dim.*dimension_name*')}	Valeur de la dimension spécifiée. Remplacez *dimension_name* par le nom de votre dimension en respectant la casse.
$ {PROP ('MetricName')}	Le nom de la métrique.
${PROP('Namespace')}	Espace de noms de la métrique.
${PROP('Period')}	Période de la métrique en secondes.
${PROP('Region')}	AWS Région dans laquelle la métrique est publiée.
${PROP('Stat')}	Statistique métrique en cours de création graphique.
${SUM}	Somme des valeurs dans l'intervalle de temps actuellement affiché sur le graphique.

Par exemple, supposons que vous ayez une expression de recherche **`SEARCH(' {AWS/Lambda, FunctionName} Errors ', 'Sum')`**, qui recherche les éléments `Errors` pour chacune de vos fonctions Lambda. Si vous définissez l'étiquette sur `[max: ${MAX} Errors for Function Name ${LABEL}]`, l'étiquette de chaque métrique est [max : ***nombre*** Errors for Function Name [Erreurs pour le nom de la fonction] ***Nom***].

Vous pouvez ajouter jusqu'à six valeurs dynamiques à une étiquette. Vous ne pouvez utiliser l'espace réservé `${LABEL}` qu'une seule fois dans chaque étiquette.

Modifier la plage horaire ou le format du fuseau horaire d'un graphique

Cette section décrit comment modifier le format de la date, de l'heure et du fuseau horaire sur un graphique de CloudWatch mesures. Elle décrit également comment zoomer sur un graphique pour appliquer une plage horaire spécifique. Pour de plus amples informations sur la création d'un graphique, consultez Représenter graphiquement une métrique.

> **Note**
>
> Si la plage de temps d'un tableau de bord est inférieure à la période utilisée pour un graphique sur le tableau de bord, les événements suivants se produisent :

- Le graphique est modifié pour afficher la quantité de données correspondant à une période complète pour ce widget, même si cette période est plus longue que celle du tableau de bord. Cela garantit la présence d'au moins un point de données sur le graphique.
- L'heure de début de la période pour ce point de données est ajustée à rebours pour s'assurer qu'au moins un point de données puisse être affiché.

Définir une plage horaire relative

New interface

Spécifier une plage de temps relative pour un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques). Dans le coin supérieur droit de l'écran, vous pouvez sélectionner une des plages de temps prédéfinies, qui s'étendent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j ou 1 sem). Vous pouvez également choisir Custom (Personnalisée) pour définir votre propre plage horaire.
3. Choisissez Custom (Personnalisée), puis sélectionnez l'onglet Relative dans le coin supérieur gauche de la boîte de dialogue. Vous pouvez spécifier une plage de temps en Minutes, Hours (Heures), Days (Jours), Weeks (Semaines), Months (Mois).
4. Une fois que vous avez spécifié une plage de temps, choisissez Apply (Appliquer).

Original interface

Spécifier une plage de temps relative pour un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques). Dans le coin supérieur droit de l'écran, vous pouvez sélectionner une des plages de temps prédéfinies, qui s'étendent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j ou 1 sem). Vous pouvez également choisir custom (personnalisée) pour définir votre propre plage horaire.

3. Choisissez custom (personnalisé), puis choisissez Relative dans le coin supérieur gauche de la boîte. Vous pouvez spécifier une plage de temps en Minutes, Hours (Heures), Days (Jours), Weeks (Semaines) ou Months (Mois).

Définir une plage horaire absolue

New interface

Spécifier une plage de temps absolue pour un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques). Dans le coin supérieur droit de l'écran, vous pouvez sélectionner une des plages de temps prédéfinies, qui s'étendent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j ou 1 sem). Vous pouvez également choisir Custom (Personnalisée) pour définir votre propre plage horaire.
3. Choisissez Custom (Personnalisée), puis sélectionnez l'onglet Absolute (Absolue) dans le coin supérieur gauche de la boîte de dialogue. Utilisez le sélecteur de calendrier ou les champs de texte pour spécifier la plage de temps.
4. Une fois que vous avez spécifié une plage de temps, choisissez Apply (Appliquer).

Original interface

Spécifier une plage de temps absolue pour un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques). Dans le coin supérieur droit de l'écran, vous pouvez sélectionner une des plages de temps prédéfinies, qui s'étendent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j ou 1 sem). Vous pouvez également choisir custom (personnalisée) pour définir votre propre plage horaire.
3. Choisissez custom (personnalisée), puis choisissez Absolute (Absolue) dans le coin supérieur gauche de la boîte de dialogue. Utilisez le sélecteur de calendrier ou les champs de texte pour spécifier la plage de temps.
4. Une fois que vous avez spécifié une plage de temps, choisissez Apply (Appliquer).

Définir le format du fuseau horaire

New interface

Spécifier un fuseau horaire pour un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques). Dans le coin supérieur droit de l'écran, vous pouvez sélectionner une des plages de temps prédéfinies, qui s'étendent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j ou 1 sem). Vous pouvez également choisir Custom (Personnalisée) pour définir votre propre plage horaire.
3. Choisissez Custom (Personnalisée), puis choisissez le menu déroulant dans le coin supérieur droit de la boîte de dialogue. Vous pouvez changer le fuseau horaire sur UTC ou Local time zone (Fuseau horaire local).
4. Après avoir effectué vos modifications, choisissez Apply (Appliquer).

Original interface

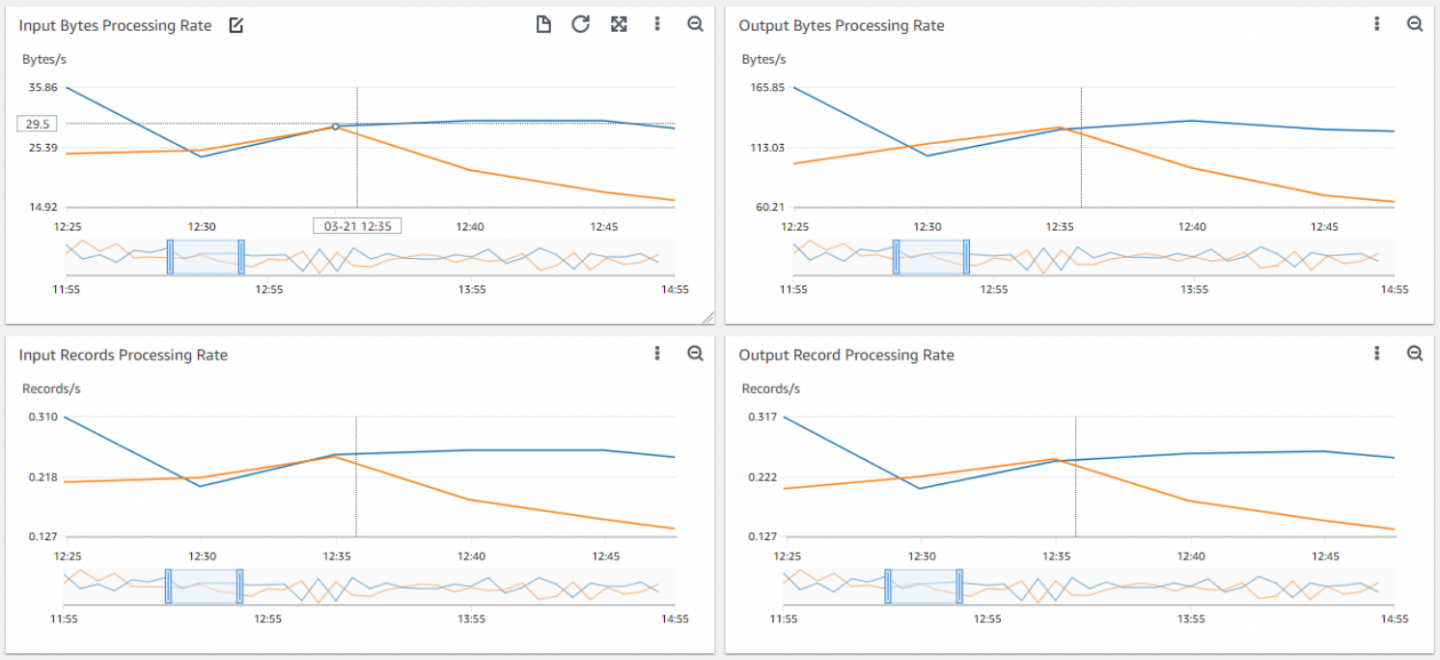
Spécifier un fuseau horaire pour un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques). Dans le coin supérieur droit de l'écran, vous pouvez sélectionner une des plages de temps prédéfinies, qui s'étendent de 1 heure à 1 semaine (1h, 3h, 12h, 1j, 3j ou 1 sem). Vous pouvez également choisir custom (personnalisée) pour définir votre propre plage horaire.
3. Choisissez custom (personnalisée), puis choisissez le menu déroulant dans le coin supérieur droit de la boîte de dialogue. Vous pouvez changer le fuseau horaire sur UTC ou Local timezone (Fuseau horaire local).

Zoomer avant sur un graphique linéaire ou un graphique à aires empilées

Dans la CloudWatch console, vous pouvez utiliser la fonction de zoom de la mini-carte pour vous concentrer sur des sections de graphiques linéaires et de graphiques à aires empilées sans passer d'une vue agrandie à une vue agrandie. Par exemple, vous pouvez utiliser la fonction de zoom de

la mini-carte pour mettre l'accent sur un pic dans un graphique linéaire, de sorte que vous puissiez comparer le pic à d'autres mesures de votre tableau de bord à partir de la même chronologie. Les procédures de cette section expliquent comment utiliser la fonction de zoom.



Dans l'image précédente, la fonction de zoom met l'accent sur un pic dans un graphique linéaire lié au taux de traitement des octets en entrée, et affiche d'autres graphiques linéaires du tableau de bord qui mettent l'accent sur des sections de la même chronologie.

New interface

Zoomer en avant sur un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques).
3. Choisissez Browse (Parcourir), puis sélectionnez une ou plusieurs métriques à représenter graphiquement.
4. Choisissez Options, puis sélectionnez Line (Ligne) sous Widget type (Type de widget).
5. Choisissez et faites glisser la zone du graphique sur laquelle vous souhaitez mettre l'accent, puis déposez.
6. Pour réinitialiser le zoom, choisissez l'icône Reset zoom (Réinitialiser le zoom), qui ressemble à une loupe avec un symbole moins (-) à l'intérieur.

Original interface

Zoomer en avant sur un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques).
3. Choisissez All metrics (Toutes les métriques), puis sélectionnez une métrique à représenter graphiquement.
4. Choisissez Graph options (Options de graphique). Sous Widget type (Type de widget), sélectionnez Line (Ligne).
5. Choisissez et faites glisser la zone du graphique sur laquelle vous souhaitez mettre l'accent, puis déposez.
6. Pour réinitialiser le zoom, choisissez l'icône Reset zoom (Réinitialiser le zoom), qui ressemble à une loupe avec un symbole moins (-) à l'intérieur.

> **Tip**
>
> Si vous avez déjà créé un tableau de bord contenant un graphique linéaire ou un graphique à aires empilées, vous pouvez y accéder et commencer à utiliser la fonction de zoom.

Modifier l'axe Y d'un graphique

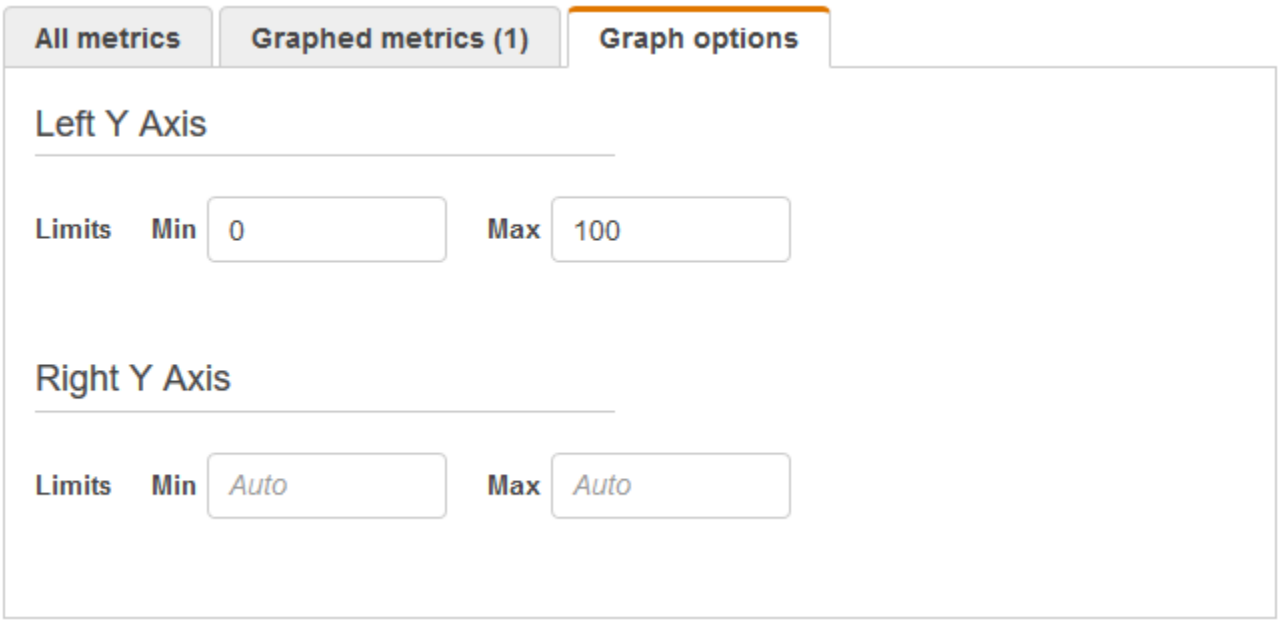
Vous pouvez définir des limites personnalisées pour l'axe des Y sur un graphique afin de mieux voir les données. Par exemple, vous pouvez modifier les limites sur un graphique `CPUUtilization` à 100 % afin de facilement voir si l'UC est faible (la courbe est proche du bas du graphique) ou élevée (la courbe est proche du haut du graphique).

Vous pouvez basculer entre deux axes des Y différents pour votre graphique. Cela est utile si le graphique contient des métriques ayant des unités différentes ou qui diffèrent considérablement dans leur plage de valeurs.

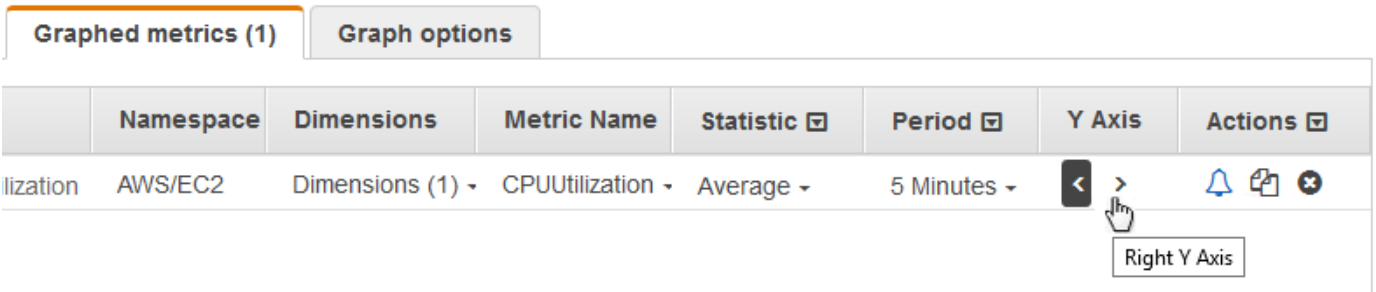
Modifier l'axe des Y sur un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.

3. Sélectionnez un espace de noms de métrique (par exemple, EC2), puis une dimension de métrique (par exemple, Per-Instance Metrics [Métriques par instance]).
4. L'onglet All metrics (Toutes les métriques) affiche toutes les métriques pour cette dimension dans cet espace de noms. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique.
5. Sur l'onglet Graph options (Options de graphique), spécifiez les valeurs Min et Max pour Left Y Axis (Axe des Y gauche). La valeur Min ne peut pas être supérieure à la valeur Max.



6. Pour créer un deuxième axe des Y, spécifiez les valeurs Min et Max pour Right Y Axis (Axe des Y droit).
7. Pour basculer entre les deux axes des Y, choisissez l'onglet Graphed metrics (Grahique des métriques). Pour Y Axis (Axe des Y), choisissez Left Y Axis (Axe des Y gauche) ou Right Y Axis (Axe des Y droite).

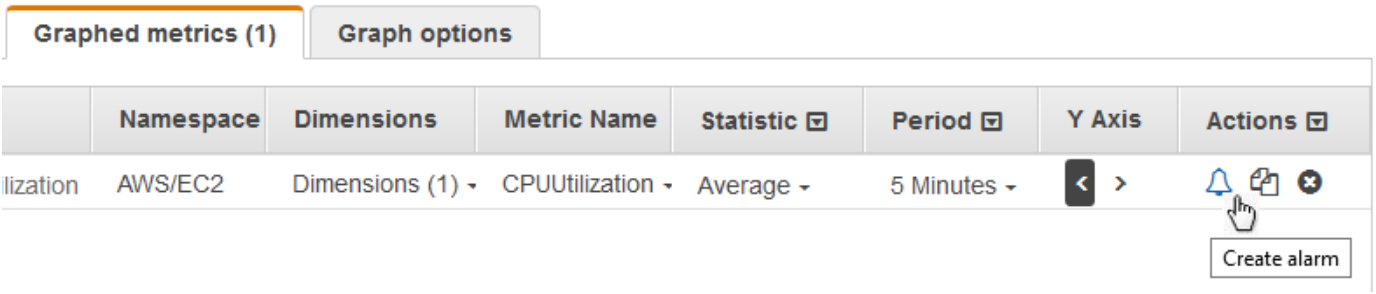


Créer une alerte à partir d'une métrique sur un graphique

Vous pouvez représenter graphiquement une métrique et créer une alerte à partir de cette métrique sur le graphique, qui présente l'avantage de remplir un grand nombre des champs de l'alerte pour vous.

Pour créer une alerte à partir d'une métrique d'un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sélectionnez un espace de noms de métrique (par exemple, EC2), puis une dimension de métrique (par exemple, Per-Instance Metrics [Métriques par instance]).
4. L'onglet All metrics (Toutes les métriques) affiche toutes les métriques pour cette dimension dans cet espace de noms. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique.
5. Pour créer une alerte pour la métrique, choisissez l'onglet Graphed metrics (Graphique des métriques). Pour Actions, sélectionnez l'icône d'alerte.



6. Sous Conditions, choisissez Static (Statique) ou Anomaly Détection (Détection d'anomalies) pour spécifier s'il faut utiliser un seuil statique ou un modèle de détection d'anomalies pour l'alerte.

   Selon votre choix, entrez le reste des données pour les conditions d'alerte.
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 pour la première valeur un nombre inférieur à celui de la seconde valeur. Pour plus d'informations, consultez . Évaluation d'une alerte.

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, consultez . Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.
9. Choisissez Next (Suivant).
10. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alerte passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Pour que l'alerte envoie plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, choisissez Add notification (Ajouter une notification).

    Pour que l'alerte n'envoie pas de notifications, choisissez Remove (Supprimer).
11. Pour que l'alerte exécute Auto Scaling ou des actions EC2, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer.
12. Lorsque vous avez terminé, choisissez Next (Suivant).
13. Saisissez un nom et une description pour l'alerte. Le nom ne doit contenir que des caractères ASCII. Sélectionnez ensuite Next (Suivant).
14. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

Utilisation de la détection des CloudWatch anomalies

Lorsque vous activez la détection des anomalies pour une métrique, CloudWatch applique des algorithmes statistiques et d'apprentissage automatique. Ces algorithmes analysent en permanence les métriques des systèmes et des applications pour déterminer les références normales et les anomalies de surface avec une intervention minimale de l'utilisateur.

Les algorithmes génèrent un modèle de détection d'anomalies. Le modèle génère une plage de valeurs attendues qui représentent le comportement normal des métriques.

Vous pouvez activer la détection des anomalies à l'aide du AWS Management Console, du AWS CLI AWS CloudFormation, ou du AWS SDK. Vous pouvez activer la détection des anomalies sur les métriques vendues par AWS et également sur les métriques personnalisées. Dans un compte configuré en tant que compte de surveillance pour l'observabilité CloudWatch entre comptes, vous pouvez créer des détecteurs d'anomalies sur les métriques des comptes sources en plus des métriques du compte de surveillance.

Vous pouvez utiliser le modèle des valeurs attendues de deux manières :

- Vous pouvez créer des alertes de détection d'anomalie basées sur la valeur attendue d'une métrique. Ces types d'alerte n'ont pas de seuil statique pour déterminer l'état de l'alerte. Au lieu de cela, ils comparent la valeur de la mesure à la valeur attendue en fonction du modèle de détection d'anomalies.

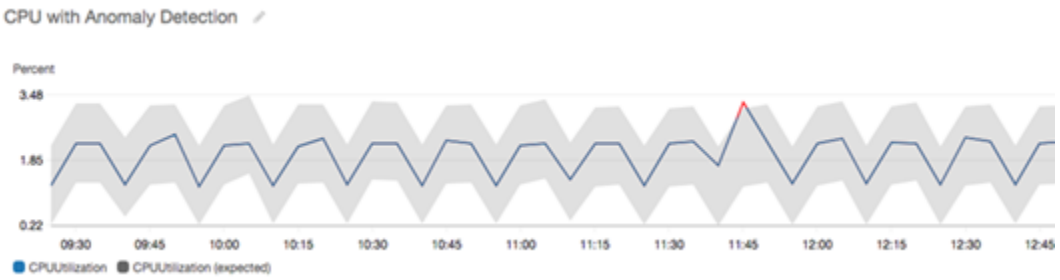
  Vous pouvez choisir si l'alerte est déclenchée lorsque la valeur de la métrique est supérieure à la bande de valeurs attendues, inférieure à la bande ou les deux.

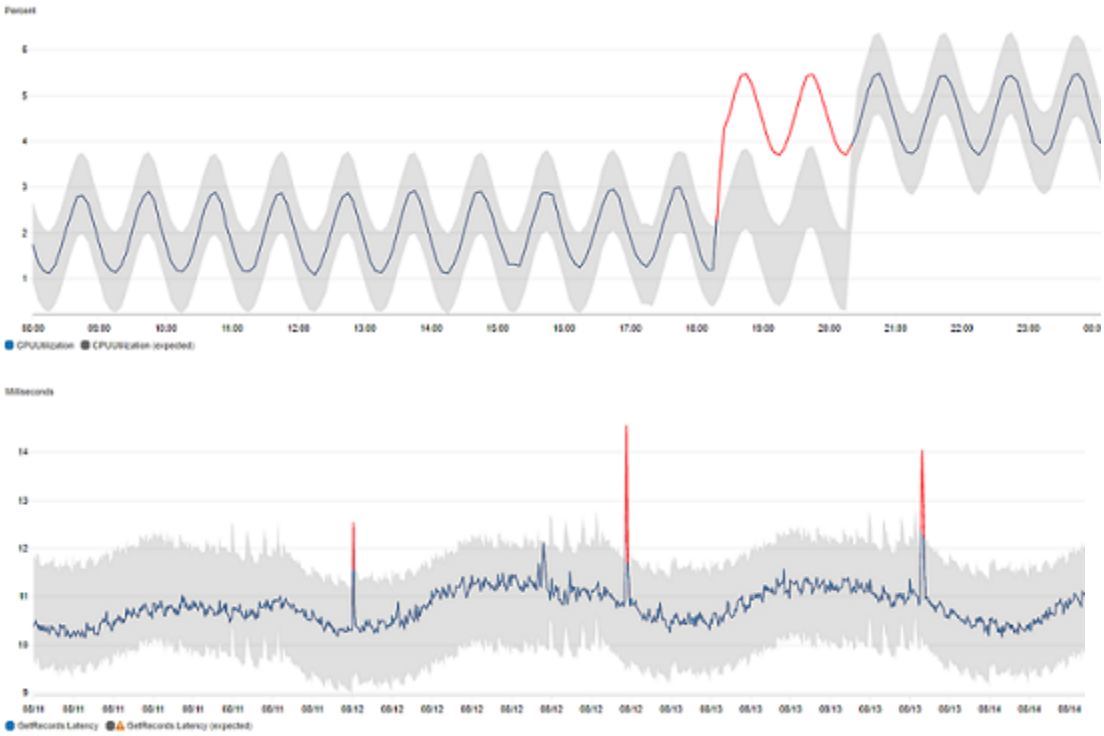
  Pour plus d'informations, consultez Création d'une CloudWatch alarme basée sur la détection d'anomalies.
- Lors de l'affichage d'un graphique des données de métrique, superposez les valeurs attendues sur le graphique sous forme de bande. Cela permet de voir clairement dans le graphique les valeurs qui sont en dehors de la plage normale. Pour plus d'informations, consultez Création d'un graphique.

  Vous pouvez également extraire les valeurs supérieures et inférieures du groupe du modèle à l'aide de la demande de l'API `GetMetricData` avec la fonction mathématique des métriques `ANOMALY_DETECTION_BAND`. Pour plus d'informations, consultez GetMetricData.

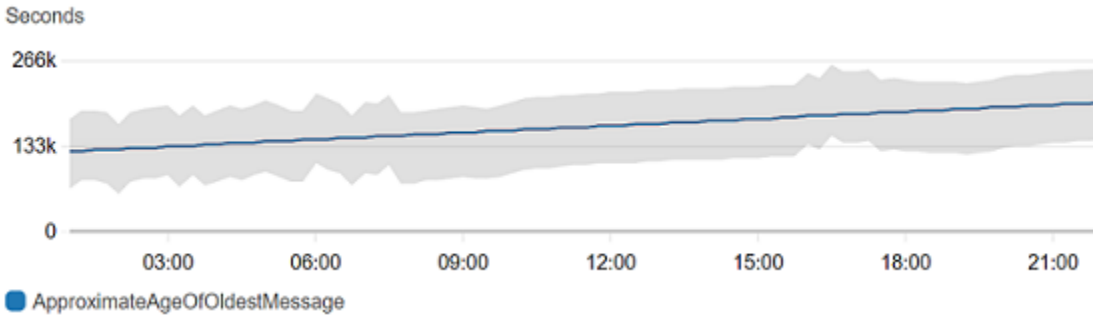
Dans un graphique avec détection d'anomalies, la plage de valeurs attendue est affichée sous la forme d'une bande grise. Si la valeur réelle de la métrique dépasse cette bande, elle est affichée en rouge pendant cette période.

Les algorithmes de détection des anomalies tiennent compte de la saisonnalité et des changements de tendance des métriques. Les modifications de saisonnalité peuvent se produire toutes les heures, tous les jours ou toutes les semaines, comme le montrent les exemples suivants.

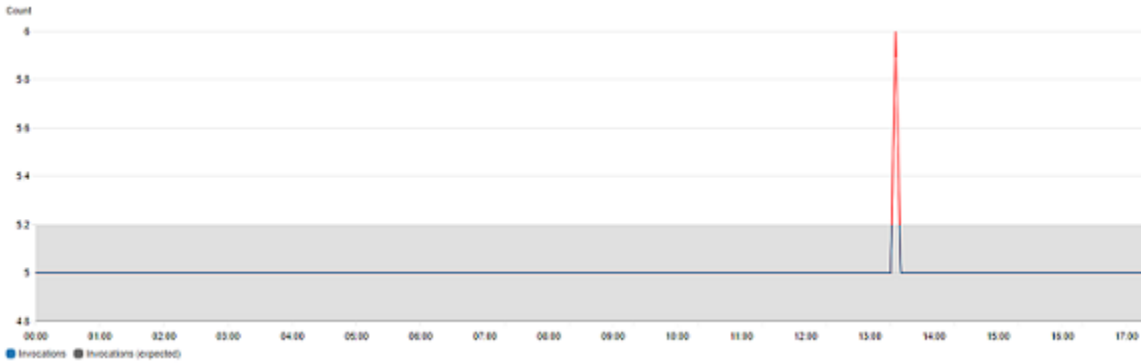




Les tendances à long terme pourraient être à la baisse ou à la hausse.



Les détections d'anomalies fonctionnent également très bien avec les métriques à modèle plat.



Comment fonctionne la détection des CloudWatch anomalies

Lorsque vous activez la détection des anomalies pour une métrique, appliquez CloudWatch des algorithmes d'apprentissage automatique aux données passées de la métrique afin de créer un modèle des valeurs attendues de la métrique. Le modèle évalue à la fois les tendances et les modèles horaires, quotidiens et hebdomadaires de la métrique. L'algorithme forme jusqu'à deux semaines de données de métriques, mais vous pouvez activer la détection d'anomalies sur une métrique, même si la métrique n'a pas deux semaines de données.

Vous spécifiez une valeur pour le seuil de détection des anomalies CloudWatch à utiliser avec le modèle pour déterminer la plage de valeurs « normale » de la métrique. Une valeur plus élevée pour le seuil de détection d'anomalies produit une bande plus épaisse de valeurs « normales ».

Le modèle de machine learning est spécifique à une métrique et à une statistique. Par exemple, si vous activez la détection d'anomalies pour une métrique à l'aide de la statistique `AVG`, le modèle est spécifique à la statistique `AVG`.

Lorsque vous CloudWatch créez un modèle pour de nombreuses métriques courantes issues AWS des services, cela garantit que la bande ne s'étend pas au-delà des valeurs logiques. Par exemple, la bande pour `MemoryUtilization` une instance EC2 restera comprise entre 0 et 100, et le suivi des bandes, qui ne peut pas être négatif CloudFront `Requests`, ne s'étendra jamais en dessous de zéro.

Après avoir créé un modèle, la détection des CloudWatch anomalies évalue le modèle en permanence et l'ajuste pour s'assurer qu'il est aussi précis que possible. Il s'agit notamment de reformer le modèle pour l'ajuster si les valeurs de métrique évoluent au fil du temps ou ont des changements brusques, et inclut également des prédicteurs pour améliorer les modèles de métriques saisonniers, pointilleux ou clairsemés.

Une fois que vous avez activé la détection d'anomalies sur une métrique, vous pouvez choisir d'exclure des périodes définies de la métrique de leur utilisation dans la formation du modèle. De cette façon, vous pouvez exclure des déploiements ou d'autres événements inhabituels de leur utilisation dans la formation du modèle, garantissant que le modèle le plus précis est créé.

L'utilisation de modèles de détection d'anomalies pour les alarmes entraîne des frais sur votre AWS compte. Pour en savoir plus, consultez Tarification Amazon CloudWatch.

Détection d'anomalies sur les mathématiques appliquées aux métriques

La détection d'anomalies sur les mathématiques métriques est une fonction que vous pouvez utiliser pour créer des alertes de détection d'anomalies sur la sortie d'expressions mathématiques

métriques. Vous pouvez utiliser ces expressions pour créer des graphiques qui visualisent les canaux de détection d'anomalies. La fonction prend en charge les fonctions arithmétiques de base, les opérateurs logiques et de comparaison, et la plupart des autres fonctions. Pour plus d'informations sur les fonctions qui ne sont pas prises en charge, consultez la section Utilisation des mathématiques métriques dans le guide de CloudWatch l'utilisateur Amazon.

Vous pouvez créer des modèles de détection d'anomalies basés sur des expressions mathématiques métriques similaires à la façon dont vous avez déjà créé des modèles de détection d'anomalies. Depuis la CloudWatch console, vous pouvez appliquer la détection des anomalies aux expressions mathématiques métriques et sélectionner la détection des anomalies comme type de seuil pour ces expressions.

> **Note**
>
> La détection des anomalies sur les mathématiques métriques ne peut être activée et modifiée que dans la dernière version de l'interface utilisateur des métriques. Lorsque vous créez des détecteurs d'anomalies basés sur des expressions mathématiques métriques dans la nouvelle version de l'interface, vous pouvez les afficher dans l'ancienne version, mais pas les modifier.

Pour plus d'informations sur la création d'alertes et de modèles pour la détection d'anomalies et les calculs de métriques, consultez les sections suivantes :

- Création d'une CloudWatch alarme basée sur la détection d'anomalies
- Création d'une CloudWatch alarme basée sur une expression mathématique métrique

Vous pouvez également créer, supprimer et découvrir des modèles de détection d'anomalies basés sur des expressions mathématiques métriques à l'aide de l' CloudWatch API avec `PutAnomalyDetector`, `DeleteAnomalyDetector` et`DescribeAnomalyDetectors`. Pour plus d'informations sur ces actions d'API, consultez les sections suivantes dans le Amazon CloudWatch API Reference.

- PutAnomalyDetector
- DeleteAnomalyDetector
- DescribeAnomalyDetectors

Pour plus d'informations sur le prix des alarmes de détection d'anomalies, consultez les CloudWatch tarifs Amazon.

Utilisation des mathématiques appliquées aux métriques

Les mathématiques métriques vous permettent d'interroger plusieurs CloudWatch métriques et d'utiliser des expressions mathématiques pour créer de nouvelles séries chronologiques basées sur ces métriques. Vous pouvez visualiser les séries chronologiques obtenues sur la CloudWatch console et les ajouter aux tableaux de bord. En utilisant AWS Lambda les métriques comme exemple, vous pouvez diviser la `Errors` métrique par la `Invocations` métrique pour obtenir un taux d'erreur. Ajoutez ensuite les séries chronologiques obtenues à un graphique sur votre CloudWatch tableau de bord.

Vous pouvez également utiliser des mathématiques appliquées aux métriques par programmation, en utilisant l'opération d'API `GetMetricData`. Pour plus d'informations, consultez GetMetricData.

Ajouter une expression mathématique à un CloudWatch graphique

Vous pouvez ajouter une expression mathématique à un graphique de votre CloudWatch tableau de bord. Chaque graphique est limité à 500 métriques et expressions. Par conséquent, vous ne pouvez ajouter une expression mathématique que si le nombre de métriques contenues dans le graphique est de 499 au maximum. Cela s'applique même si toutes les métriques ne sont pas affichées sur le graphique.

Pour ajouter une expression mathématique à un graphique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Création ou modification d'un graphique Le graphique doit contenir au moins une métrique.
3. Choisissez Graphed metrics (Graphique des métriques).
4. Choisissez Math expression (Expression mathématique), puis Start with empty expression (Commencer par une expression vide). Une nouvelle ligne s'affiche pour l'expression.
5. Dans la nouvelle ligne, sous la colonne Details (Détails), saisissez l'expression mathématique. Les tables de la rubrique Metric Math Syntax and Functions répertorient les fonctions que vous pouvez utiliser dans l'expression.

   Pour utiliser une métrique ou le résultat d'une autre expression dans le cadre de la formule pour cette expression, utilisez la valeur indiquée dans la colonne Id : par exemple, m1+m2 ou e1-MIN(e1).

Vous pouvez modifier la valeur de Id. Elle peut comprendre des chiffres, des lettres et des traits de soulignement et doit commencer par une lettre minuscule. La modification de la valeur de Id à un nom plus descriptif peut également rendre un graphique plus compréhensible : par exemple, remplacement de m1 et m2 par errors (erreurs) et requests (demandes).

> **Tip**
>
> Choisissez la flèche vers le bas en regard de Math Expression (Expression mathématique) pour afficher la liste des fonctions prises en charge, que vous pouvez utiliser lors de la création de votre expression.

6. Pour la colonne Label (Étiquette) de l'expression, saisissez un nom qui décrit ce que calcule l'expression.

   Si le résultat d'une expression est un ensemble de séries temporelles, chacune de ces séries temporelles s'affiche sur le graphique sur une ligne distincte, avec une couleur différente. La légende située directement sous le graphique correspond à chaque ligne du graphique. Pour une expression unique qui génère plusieurs séries temporelles, le format des sous-titres de la légende pour ces séries temporelles est ***`Expression-Étiquette Métrique-Étiquette`***. Par exemple, si le graphique comprend une métrique avec une étiquette Errors (Erreurs) et une expression FILL(METRICS(), 0) avec une étiquette Filled With 0: (Rempli avec 0 :), une ligne de la légende sera Filled With 0: Errors (Rempli avec 0 : Erreurs). Définissez *`Expression-Étiquette`* de manière à ce qu'elle soit vide pour que la légende affiche uniquement les étiquettes de métriques d'origine.

   Lorsqu'une expression génère un ensemble de séries temporelles sur le graphique, vous ne pouvez pas modifier les couleurs utilisées pour chacune des séries temporelles.

7. Une fois que vous avez ajouté les expressions souhaitées, vous pouvez simplifier le graphique en masquant certaines métriques d'origine. Pour masquer une métrique ou une expression, décochez la case à gauche du champ Id.

Syntaxe et fonctions des mathématiques appliquées aux métriques

Les sections suivantes expliquent les fonctions disponibles pour les mathématiques appliquées aux métriques. Toutes les fonctions doivent être entièrement écrites en majuscules (comme AVG), alors que le champ Id pour toutes les métriques et expressions mathématiques doit commencer par une lettre minuscule.

Le résultat final d'une expression mathématique doit être une seule série temporelle ou un ensemble de séries temporelles. Certaines fonctions génèrent un nombre scalaire. Vous pouvez utiliser ces fonctions dans le cadre d'une plus grande fonction qui génère une série temporelle. Par exemple, la fonction AVG d'une seule série temporelle produit un nombre scalaire qui ne peut donc pas être le résultat de l'expression finale. Vous pouvez cependant l'utiliser dans la fonction m1-AVG(m1) pour afficher une série temporelle de la différence entre chaque point de données individuel et la valeur moyenne de cette série temporelle.

Abréviations des types de données

Certaines fonctions sont valides pour certains types de données uniquement. Les abréviations de la liste suivante sont utilisées dans les tableaux des fonctions pour représenter les types de données pris en charge pour chaque fonction :

- S représente un nombre scalaire, par exemple 2, -5, ou 50,25.
- TS est une série chronologique (une série de valeurs pour une seule CloudWatch métrique au fil du temps) : par exemple, la `CPUUtilization` métrique des trois derniers jours. `i-1234567890abcdef0`
- TS[] est un ensemble de séries chronologiques, par exemple les séries chronologiques pour plusieurs métriques.
- String[] est un tableau de chaînes de caractères.

Fonction METRICS()

La fonction METRICS() renvoie toutes les métriques de la demande. Les expressions mathématiques ne sont pas incluses.

Vous pouvez utiliser la fonction METRICS() au sein d'une plus grande expression qui génère une seule série chronologique ou un ensemble de séries temporelles. Par exemple, l'expression SUM(METRICS()) renvoie une série temporelle (TS) qui est la somme des valeurs de l'ensemble des métriques du graphique. METRICS()/100 renvoie un ensemble de séries temporelles, chacun étant une série temporelle montrant chaque point de données de l'une des métriques divisée par 100.

Vous pouvez utiliser la fonction METRICS() avec une chaîne pour renvoyer uniquement les métriques du graphique qui contiennent cette chaîne dans leur champ Id. Par exemple, l'expression SUM(METRICS("errors")) renvoie une série temporelle qui est la somme des valeurs de l'ensemble des métriques du graphique comportant « errors » dans leur champ Id. Vous pouvez également utiliser SUM([METRICS(“4xx”), METRICS(“5xx”)]) pour faire correspondre plusieurs chaînes.

Fonctions arithmétiques de base

Le tableau suivant répertorie les fonctions arithmétiques de base prises en charge. Les valeurs manquantes dans la série temporelle sont traitées en tant que 0. Si la valeur d'un point de données entraîne une fonction à tenter de diviser par zéro, le point de données est abandonné.

Opération	Arguments	Exemples
Opérateurs arithmétiques : + - * / ^	S, S S, TS TS, TS S, TS[] TS, TS[]	PERIOD(m1)/60 5 * m1 m1 - m2 SUM(100/[m1, m2]) AVG(METRICS()) METRICS()*100
Soustraction unaire -	S TS TS[]	-5*m1 -m1 SUM(-[m1, m2])

Opérateurs logiques et de comparaison

Vous pouvez utiliser des opérateurs logiques et de comparaison avec une paire de séries temporelles ou une paire de valeurs scalaires simples. Lorsque vous utilisez un opérateur de comparaison avec une paire de séries temporelles, les opérateurs renvoient une série temporelle dans laquelle chaque point de données est 0 (false) ou 1 (true). Si vous utilisez un opérateur de comparaison sur une paire de valeurs scalaires, une seule valeur scalaire est renvoyée, 0 ou 1.

Lorsque des opérateurs de comparaison sont utilisés entre deux séries temporelles et qu'une seule des séries temporelles a une valeur pour un horodatage particulier, la fonction traite la valeur manquante dans l'autre série temporelle en tant que 0.

Vous pouvez utiliser des opérateurs logiques en conjonction avec des opérateurs de comparaison, pour créer des fonctions plus complexes.

Le tableau suivant répertorie les opérateurs pris en charge.

Type d'opérateur	Opérateurs pris en charge
Opérateurs de comparaison	== != <= >= < >
Opérateurs logiques	ET ou && OU ou ||

Pour voir comment ces opérateurs sont utilisés, prenons deux séries temporelles : metric1 a des valeurs de `[30, 20, 0, 0]` et metric2 des valeurs de `[20, -, 20, -]` où `-` indique qu'il n'y a pas de valeur pour cet horodatage.

Expression	Sortie
(metric1 < metric2)	0, 0, 1, 0
(metric1>=30)	1, 0, 0, 0
(metric1>15 AND metric2>15)	1, 0, 0, 0

Fonctions prises en charge pour les mathématiques appliquées aux métriques

Le tableau suivant décrit les fonctions que vous pouvez utiliser dans les expressions mathématiques. Saisissez toutes les fonctions en majuscules.

Le résultat final d'une expression mathématique doit être une seule série temporelle ou un ensemble de séries temporelles. Certaines fonctions des tableaux dans les sections suivantes génèrent un

nombre scalaire. Vous pouvez utiliser ces fonctions dans le cadre d'une plus grande fonction qui génère une série temporelle. Par exemple, la fonction AVG d'une seule série temporelle produit un nombre scalaire qui ne peut donc pas être le résultat de l'expression finale. Vous pouvez cependant l'utiliser dans la fonction m1-AVG(m1) pour afficher une série temporelle de la différence entre chaque point de données individuel et la valeur moyenne de ce point de données.

Dans le tableau suivant, chaque exemple de la colonne Exemples est une expression qui renvoie une seule série temporelle ou un ensemble de séries temporelles. Ces exemples montrent de quelle manière les fonctions qui renvoient des nombres scalaires peuvent être utilisées dans le cadre d'une expression valable qui génère une seule série temporelle.

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
ABS	TS TS[]	TS TS[]	Renvoie la valeur absolue de chaque point de données.	ABS(m1-m2) MIN(ABS([m1, m2])) ABS(METRICS())	✓
ANOMALY_DETECTION_BAND	TS TS, S	TS[]	Renvoie un groupe de détection d'anomalies pour la métrique spécifiée. Le groupe se compose de deux séries chronologiques, l'une représentant la limite supérieure de la valeur attendue « normale » de la métrique, et l'autre qui représente la limite inférieure. La fonction accepte deux arguments	ANOMALY_DETECTION_BAND (m1) ANOMALY_DETECTION_BAND (m1,4)	

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			. Le premier est l'ID de la métrique pour laquelle créer le groupe. Le deuxième argument est le nombre d'écarts-types à utiliser pour le groupe. Si vous ne spécifiez pas cet argument, la valeur par défaut de 2 est utilisée. Pour plus d'informations, consultez Utilisation de la détection des CloudWatch anomalies.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
AVG	TS TS[]	S TS	La fonction AVG d'une seule série temporell e renvoie un nombre scalaire qui représente la moyenne de tous les points de données de la métrique. La fonction AVG d'un ensemble de séries temporell es renvoie une seule série temporelle. Les valeurs manquantes sont traitées comme 0. **Note** Nous vous recommand ons de ne pas utiliser cette fonction dans les CloudWatc h alarmes si vous souhaitez qu'elle renvoie un scalaire. Par exemple, `AVG(m2)`. Chaque fois	SUM([m1,m2])/AVG(m2) AVG(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			qu'une alarme évalue s'il faut changer d'état, CloudWatch tente de récupérer un nombre de points de données supérieur au nombre spécifié comme périodes d'évaluation. Cette fonction agit différemment lorsque des données supplémentaires sont demandées. Pour utiliser cette fonction avec des alertes, en particulier des alertes comportant des actions Auto Scaling,		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			nous vous recommandons de régler l'alerte pour qu'elle utilise M sur N points de données, où M < N.		
CEIL	TS TS[]	TS TS[]	Renvoie la valeur plafond de chaque métrique. Le plafond est le plus petit entier supérieur ou égal à chaque valeur.	CEIL(m1) CEIL(METRICS()) SUM(CEIL(METRICS()))	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
DATAPOINT_COUNT	TS TS[]	S TS	Renvoie le nombre des points de données qui ont rapporté des valeurs. Ceci est utile pour calculer les moyennes des métriques clairsemées. **Note** Nous vous recommandons de ne pas utiliser cette fonction dans les CloudWatch alarmes. Chaque fois qu'une alarme évalue s'il faut changer d'état, CloudWatch tente de récupérer un nombre de points de données supérieur au nombre spécifié	SUM(m1) / DATAPOINT_COUNT(m1) DATAPOINT_COUNT(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			comme périodes d'évaluation. Cette fonction agit différemm ent lorsque des données supplémentaires sont demandées .		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
DB_PERF_INSIGHTS	String, String, String String, String, String[]	TS (avec une seule chaîne) TS[] (avec un tableau de chaînes de caractèr s)	Renvoie les métriques de compteur d'Analyse des performances pour des bases de données telles qu'Amazon Relational Database Service et Amazon DocumentDB (compatib le avec MongoDB). Cette fonction renvoie la même quantité de données que celle que vous pouvez obtenir en interrogeant directeme nt les API d'Analyse des performances. Vous pouvez utiliser ces mesures CloudWatc h pour représenter graphiquement et créer des alarmes. **Important** Lorsque vous utilisez cette fonction, vous devez spécifier l'ID	DB_PERF_INSIGHTS(‘ RDS', ‘db-ABCDE FGHIJKLMN OPQRSTUVWXY1', ‘os.cpuUtilization .user.avg') DB_PERF_INSIGHTS(‘ DOCDB, ‘db-ABCDEFGHIJKLMN OPQRSTUVWXY1', [‘os.cpuUtilization.idle.av g', ‘os.cpuUtilization .user.max'])	

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			de ressource de base de données unique de la base de données. Il est différent de l’identif iant de base de données. Pour trouver un identifiant de ressource de base de données dans la console Amazon RDS, choisisse z l’instance de base de données pour en afficher les détails. Choisissez ensuite l'onglet Configura tion. L’ID de ressource est indiqué		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			dans la section Configuration. DB_PERF_INSIGHTS introduit également la métrique DBLoad à des intervalles inférieurs à la minute. Les métriques Performance Insights récupérées à l'aide de cette fonction ne sont pas stockées dans CloudWatch. Par conséquent, certaines CloudWatch fonctionnalités telles que l'observabilité entre comptes, la détection des anomalies, les flux métriques, l'explorateur de métriques et Metric Insights ne fonctionnent pas avec les métriques Performance Insights que vous récupérez		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			avec DB_PERF_INSIGHTS. Une seule demande utilisant la fonction DB_PERF_INSIGHTS peut récupérer les nombres de points de données suivants. • 1080 points de données pour des périodes à haute résolution (1 s, 10 s, 30 s) • 1440 points de données pour des périodes de résolution standard (1 m, 5 m, 1 heure, 1 jour) La fonction DB_PERF_INSIGHTS ne prend en charge que les durées de période suivantes : • 1 seconde • 10 secondes • 30 secondes		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			• 1 minute • 5 minutes • 1 heure • 1 jour Pour obtenir plus d'informations sur les métriques de compteur d'Analyse des performances d'Amazon RDS, consultez Métrique de compteur de Performance Insights. Pour obtenir plus d'informations sur les métriques de compteur d'Analyse des performances d'Amazon DocumentDB, consultez Performance Insights for counter metrics. **Note** Les métriques haute résolution avec une		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			granularité inférieure à la minute récupérées par DB_PERF_INSIGHTS ne s'appliquent qu'à la métrique DBLoad, ou aux métriques du système d'exploitation si vous avez activé la Surveillance améliorée à une résolution supérieure. Pour de plus amples informations sur la Surveillance améliorée d'Amazon RDS, consultez Surveillance des métriques du système d'exploitation à l'aide de la		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			Surveillance améliorée. Vous pouvez créer une alerte haute résolution à l'aide de la fonction DB_PERF_INSIGHTS pour une durée maximale de trois heures. Vous pouvez utiliser la CloudWatch console pour représenter graphiquement les métriques récupérées à l'aide de la fonction DB_PERF_INSIGHTS pour n'importe quel intervalle de temps.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
DIFF	TS TS[]	TS TS[]	Renvoie la différence entre chaque valeur de la série temporelle et la valeur précédente de cette série temporelle.	DIFF(m1)	✓
DIFF_TIME	TS TS[]	TS TS[]	Renvoie la différence en secondes entre l'horodatage de chaque valeur de la série temporelle et l'horodatage de la valeur précédente de cette série temporelle.	DIFF_TIME(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
FILL	TS, [S | REPEA | LINEAR TS[], [TS | S | REPEA | LINEAR	TS TS[]	Remplit les valeurs manquantes d'une série temporelle. Il existe plusieurs options pour les valeurs à utiliser comme remplissage pour les valeurs manquantes : • Vous pouvez spécifier une valeur à utiliser comme valeur de remplissage. • Vous pouvez spécifier une métrique à utiliser comme valeur de remplissage. • Vous pouvez utiliser le plugin REPEAT pour remplir les valeurs manquantes avec la valeur réelle la plus récente de la métrique avant la valeur manquante. • Vous pouvez utiliser le plugin LINEAR pour remplir les valeurs	FILL(m1,10) FILL(METRICS(), 0) FILL(METRICS(), m1) FILL(m1, MIN(m1)) FILL(m1, REPEAT) FILL(METRICS(), LINEAR)	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			manquantes avec des valeurs qui créent une interpolation linéaire entre les valeurs au début et à la fin de l'écart. **Note** Lorsque vous utilisez cette fonction dans une alerte, vous pouvez rencontrer un problème si vos métriques sont publiées avec un léger retard et que la minute la plus récente n'a jamais de données. Dans ce cas, FILL remplace ce point de données manquant		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			par la valeur demandée. Cela fait que le dernier point de données de la métrique est toujours la valeur FILL, ce qui peut entraîner le blocage de l'alerte dans l'état OK ou alerte. Vous pouvez contourner cela en utilisant une alerte M sur N. Pour plus d'informations, consultez Évaluation d'une alerte.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
FIRST LAST	TS[]	TS	Il renvoie la première ou la dernière série temporelle à partir d'un tableau de séries temporelles. Ceci est utile en l'utilisant avec la fonction SORT (TRI). Il est également possible de l'utiliser pour obtenir les seuils élevés et bas à partir de la fonction ANOMALY_DETECTION_BAND.	IF(FIRST(SORT(METRICS(), AVG, DESC))>100, 1, 0) examine la métrique supérieure d'un tableau, qui est trié par AVG. Il renvoie ensuite un 1 ou un 0 pour chaque point de données, selon que la valeur de ce point de données est supérieure à 100. LAST(ANOMALY_DETECTION_BAND(m1)) renvoie la limite supérieure de la bande de prédiction des anomalies.	✓
FLOOR	TS TS[]	TS TS[]	Renvoie le plancher de chaque métrique. Le plancher est le plus grand nombre entier inférieur ou égal à chaque valeur.	FLOOR(m1) FLOOR(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
IF	Express n IF	TS	Utilisez IF (SI) avec un opérateur de comparaison pour filtrer les points de données d'une série temporelle ou créer une série temporelle mixte composée de plusieurs séries temporelles assemblées. Pour plus d'informations, consultez Utilisation des expressions IF.	Pour obtenir des exemples, consultez Utilisation des expressions IF.	✓
INSIGHT_RULE_METRIC	INSIGHULE_MIC(ruleNe, metricNe)	TS	Utilisez INSIGHT_RULE_METRIC pour extraire des statistiques d'une règle dans Contributor Insights. Pour plus d'informations, consultez Graphique des métriques générées par les règles.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
LAMBDA	LAMBD (Lambd nctionN e [, argume facultati f] *)	TS TS{}	Appelle une fonction Lambda pour interroge r des métriques provenant d'une source de données qui n'en est pas une. CloudWatc h Pour plus d'informa tions, consultez Comment transmettre des arguments à votre fonction Lambda.		
LOG	TS TS[]	TS TS[]	Le LOG d'une série temporelle renvoie la valeur logarithm e naturel de chaque valeur de la série temporelle.	LOG(METRICS())	✓
LOG10	TS TS[]	TS TS[]	Le LOG10 d'une série temporelle renvoie la valeur logarithme de base 10 de chaque valeur de la série temporelle.	LOG10(m1)	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
MAX	TS TS[]	S TS	La fonction MAX d'une seule série temporell e renvoie un nombre scalaire qui représent e la valeur maximale de tous les points de données de la métrique. Si vous entrez un tableau de séries chronologiques, la fonction MAX crée et renvoie une série chronologique composée de la valeur la plus élevée pour chaque point de données, parmi les séries temporelles utilisées comme entrée. **Note** Nous vous recommand ons de ne pas utiliser cette fonction dans les CloudWatc	MAX(m1)/m1 MAX(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			h alarmes si vous souhaitez qu'elle renvoie un scalaire. Par exemple, `MAX(m2)` chaque fois qu'une alarme évalue s'il faut changer d'état, CloudWatch tente de récupérer un nombre de points de données supérieur au nombre spécifié sous forme de périodes d'évaluation. Cette fonction agit différemment lorsque des données supplémentaires sont demandées.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
METRIC_COUNT	TS[]	S	Renvoie le nombre de métriques dans l'ensemble de séries temporelles.	m1/METRIC_COUNT(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
METRICS	null chaîne	TS[]	La fonction METRICS () renvoie toutes les CloudWatch métriques de la demande. Les expressions mathémati ques ne sont pas incluses. Vous pouvez utiliser la fonction METRICS() au sein d'une plus grande expression qui génère une seule série chronologique ou un ensemble de séries temporelles. Vous pouvez utiliser la fonction METRICS() avec une chaîne pour renvoyer uniquemen t les métriques du graphique qui contienne nt cette chaîne dans leur champ Id. Par exemple, l'expression SUM(METRICS("error s")) renvoie une série temporelle qui est la	AVG(METRICS()) SUM(METRICS("error s"))	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			somme des valeurs de l'ensemble des métriques du graphique comportant « errors » dans leur champ Id. Vous pouvez également utiliser SUM([METRICS(“4xx”), METRICS(“5xx”)]) pour faire correspondre plusieurs chaînes.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
MIN	TS TS[]	S TS	La fonction MIN d'une seule série temporell e renvoie un nombre scalaire qui représent e la valeur minimale de tous les points de données de la métrique. Si vous entrez un tableau de séries chronologiques, la fonction MIN crée et renvoie une série chronologique composée de la valeur la plus faible pour chaque point de données, parmi les séries temporelles utilisées comme entrée. Si vous entrez un tableau de séries chronologiques, la fonction MIN crée et renvoie une série chronologique composée de la valeur la plus élevée pour	m1-MIN(m1) MIN(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			chaque point de données, parmi les séries temporelles utilisées comme entrée. **Note** Nous vous recommandons de ne pas utiliser cette fonction dans les CloudWatch alarmes si vous souhaitez qu'elle renvoie un scalaire. Par exemple, `MIN(m2)` chaque fois qu'une alarme évalue s'il faut changer d'état, CloudWatch tente de récupérer un nombre de points de données		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			supérieur au nombre spécifié sous forme de périodes d'évaluation. Cette fonction agit différemment lorsque des données supplémentaires sont demandées.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?

MINUTE
HOUR
DAY
DATE
MONTH
YEAR
EPOCH
	TS	TS	Ces fonctions prennent la période et la plage de la série temporelle et renvoient une nouvelle série temporelle non clairsemée où chaque valeur est basée sur son horodatage.
			• MINUTE renvoie une série temporelle non clairsemée d'entiers compris entre 0 et 59 qui représentent la minute UTC de chaque horodatage de la série temporelle d'origine.
			• HOUR renvoie une série temporelle non clairsemée d'entiers compris entre 0 et 23 qui représent ent l'heure UTC de chaque horodatage de la série temporelle d'origine.
				MINUTE(m1)
				IF(DAY(m1)<6,m1) renvoie des métriques uniquement à partir des jours de semaine, du lundi au vendredi.
				IF(MONTH(m1) == 4,m1) renvoie uniquement les métriques publiées en avril.
					✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			• DAY renvoie une série temporelle non clairsemée d'entiers compris entre 1 et 7 qui représentent le jour de la semaine UTC de chaque horodatage de la série temporelle d'origine. 1 représente Lundi et 7 représente Dimanche. • DATE renvoie une série temporelle non clairsemée d'entiers compris entre 1 et 31 qui représentent le jour du mois UTC de chaque horodatage de la série temporelle d'origine. • MONTH renvoie une série temporelle non clairsemée d'entiers compris entre 1 et 12 qui représentent le mois UTC de chaque horodatage de la série temporelle d'origine.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			1 représente Janvier et 12 représente Décembre. • YEAR renvoie une série temporelle non clairsemée d'entiers qui représentent l'année UTC de chaque horodatage de la série temporelle d'origine. • EPOCH renvoie une série temporell e non clairsemée d'entiers qui représent ent l'heure UTC en secondes depuis l'époque de chaque horodatage de la série temporelle d'origine. L'EPOCH (époque) est le 1er janvier 1970.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
PERIOD	TS	S	Renvoie la période de la métrique en secondes. Les métriques sont des entrées valides, mais pas les résultats d'autres expressions.	m1/PERIOD(m1)	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
RATE	TS TS[]	TS TS[]	Renvoie le taux de modification de la métrique par seconde. Il s'agit de la différence entre la dernière valeur du point de données et la valeur du point de données précédent, divisée par la différence de temps, en secondes, entre les deux valeurs. ⚠ Important La définition d'alarmes sur des expressions qui utilisent la fonction RATE sur des métriques contenant des données éparses peut se comporter de manière imprévisible, car la plage de points	RATE(m1) RATE(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			de données récupérés lors de l'évaluat ion de l'alarme peut varier en fonction de la date de dernière publication des points de données.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
REMOVE_EMPTY	TS[]	TS[]	Supprime toutes les séries temporelles qui n'ont pas de points de données d'un tableau de séries temporelles. Le résultat est un tableau de séries temporelles où chaque série temporelle contient au moins un point de données. **Note** Nous vous recommandons de ne pas utiliser cette fonction dans les CloudWatch alarmes. Chaque fois qu'une alarme évalue s'il faut changer d'état, CloudWatch tente de récupérer un nombre de points	REMOVE_EMPTY(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			de données supérieur au nombre spécifié comme périodes d'évaluation. Cette fonction agit différemment lorsque des données supplémentaires sont demandées.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
RUNNING_SUM	TS TS[]	TS TS[]	Renvoie une série temporelle avec la somme en cours d'exécution des valeurs de la série temporelle d'origine. **Note** Nous vous recommandons de ne pas utiliser cette fonction dans les CloudWatch alarmes. Chaque fois qu'une alarme évalue s'il faut changer d'état, CloudWatch tente de récupérer un nombre de points de données supérieur au nombre spécifié comme périodes	RUNNING_SUM([m1,m2])	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			d'évaluation. Cette fonction agit différemment lorsque des données supplémentaires sont demandées.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
SEARCH	Express n de recherc	Une ou plusieur TS	Renvoie une ou plusieurs séries temporelles qui correspondent à une recherche des critères que vous spécifiez. La fonction SEARCH (RECHERCHE) vous permet d'ajouter plusieurs séries temporelles associées à un graphique avec une expression. Le graphique est dynamiquement mis à jour afin d'inclure les nouvelles métriques qui sont ajoutées ultérieurement et correspondent aux critères de recherche . Pour plus d'informations, consultez Utiliser des expressions de recherche dans les graphiques. Vous ne pouvez pas créer une alerte basée		✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			sur une expression SEARCH. En effet, les expressions de recherche renvoient plusieurs séries temporelles et une alerte basée sur une expression mathémati que ne peut regarder qu'une seule série temporelle. Si vous êtes connecté à un compte de surveilla nce dans le cadre de l'observabilité CloudWatch entre comptes, la fonction SEARCH trouve des métriques dans les comptes sources et dans le compte de surveillance.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
SERVICE_QUOTA	TS qui est une métriqu d'utilisation	TS	Renvoie le quota de service pour la métrique d'utilisation donnée. Vous pouvez l'utiliser pour visualiser la comparaison de votre utilisation actuelle avec le quota et pour définir des alertes qui vous avertissent lorsque vous approchez du quota. Pour plus d'informations, consultez AWS métriques d'utilisation.		✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
SLICE	(TS[], S, S) ou (TS[], S)	TS[] TS	Récupère une partie d'un tableau de séries temporelles. Ceci est particulièrement utile en combinaison avec SORT (TRI). Par exemple, vous pouvez exclure le résultat supérieur d'un tableau de séries temporelles. Vous pouvez utiliser deux arguments scalaires pour définir l'ensemble de séries temporelles que vous souhaitez renvoyer. Les deux scalaires définissent le début (inclus) et la fin (exclusif) du tableau à renvoyer. Le tableau est indexé sur zéro, donc la première série temporelle du tableau est la série temporelle 0. Vous pouvez également spécifier une seule valeur et CloudWatch renvoyer toutes les	SLICE(SORT (METRICS(), SUM, DESC), 0, 10) renvoie les 10 métriques du tableau de métriques de la demande qui possèdent la valeur SUM la plus élevée. SLICE(SORT (METRICS(), AVG, ASC), 5) trie le tableau des métriques par la statistique AVG, puis renvoie toutes les séries temporelles sauf les 5 qui possèdent l'AVG le plus bas.	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			séries chronologiques commençant par cette valeur.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
SORT	(TS[], FUNCT SORT_(R) (TS[], FUNCT SORT_(R, S)	TS[]	Trie un tableau de séries temporelles en fonction de la fonction que vous spécifiez. La fonction que vous utilisez peut être AVG, MIN, MAX, ou SUM. L'ordre de tri peut être ASC pour croissant (valeurs les plus basses d'abord) ou DESC pour trier les valeurs les plus élevées d'abord. Vous pouvez éventuellement spécifier un nombre après l'ordre de tri qui agit comme limite. Par exemple, la spécifica tion d'une limite de 5 renvoie uniquement les 5 premières séries temporelles du tri. Lorsque cette fonction mathématique est affichée sur un graphique, les étiquette s de chaque métrique du graphique sont	SORT(METRICS(), AVG, DESC, 10) calcule la valeur moyenne de chaque série temporelle, trie la série temporelle présentan t les valeurs les plus élevées au début du tri et renvoie uniquement les 10 séries temporell es présentant les moyennes les plus élevées. SORT(METRICS(), MAX, ASC) trie le tableau des métriques selon la statistique MAX, puis les renvoie toutes dans l'ordre croissant.	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			également triées et numérotées.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
STDDEV	TS TS[]	S TS	La fonction STDDEV d'une seule série temporelle renvoie un nombre scalaire qui représente l'écart-type de tous les points de données de la métrique. La fonction STDDEV d'un ensemble de séries temporelles renvoie une seule série temporelle. **Note** Nous vous recommandons de ne pas utiliser cette fonction dans les CloudWatch alarmes si vous souhaitez qu'elle renvoie un scalaire. Par exemple, `STDDEV(m2)` chaque fois qu'une alarme évalue s'il faut	m1/STDDEV(m1) STDDEV(METRICS())	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			changer d'état, CloudWatch tente de récupérer un nombre de points de données supérieur au nombre spécifié sous forme de périodes d'évaluation. Cette fonction agit différemment lorsque des données supplémentaires sont demandées.		

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
SUM	TS TS[]	S TS	La fonction SUM d'une seule série temporelle renvoie un nombre scalaire qui représente la somme des valeurs de tous les points de données de la métrique. La fonction SUM d'un ensemble de séries temporelles renvoie une seule série temporelle. **Note** Nous vous recommandons de ne pas utiliser cette fonction dans les CloudWatch alarmes si vous souhaitez qu'elle renvoie un scalaire. Par exemple, `SUM(m1)`. Chaque fois qu'une alarme évalue s'il faut	SUM(METRICS())/SUM(m1) SUM([m1,m2]) SUM(METRICS("errors"))/SUM(METRICS("requests"))*100	✓

Fonction	Argume	Type de renvoi*	Description	Exemples	Prise en charge pour les comptes croisés ?
			changer d'état, CloudWatch tente de récupérer un nombre de points de données supérieur au nombre spécifié comme périodes d'évaluation. Cette fonction agit différemment lorsque des données supplémentaires sont demandées.		
TIME_SERIES	S	TS	Renvoie une série temporelle non clairsemée où chaque valeur est définie sur un argument scalaire.	TIME_SERIES(MAX(m1)) TIME_SERIES(5*AVG(m1)) TIME_SERIES(10)	✓

*Vous ne pouvez pas utiliser une fonction qui renvoie uniquement un nombre scalaire, car tous les résultats finaux des expressions doivent être une seule série temporelle ou un ensemble de séries temporelles. Au lieu de cela, utilisez ces fonctions dans le cadre d'une plus grande expression qui renvoie une série temporelle.

Utilisation des expressions IF

Utilisez IF (SI) avec un opérateur de comparaison pour filtrer les points de données d'une série temporelle ou créer une série temporelle mixte composée de plusieurs séries temporelles assemblées.

IF (SI) utilise les arguments suivants :

```
IF(condition, trueValue, falseValue)
```

La condition est évaluée à FALSE (FAUX) si la valeur du point de données de condition est 0, et à TRUE (VRAI) si la valeur de la condition est une autre valeur, que cette valeur soit positive ou négative. Si la condition est une série temporelle, elle est évaluée séparément pour chaque horodatage.

La liste suivante répertorie les syntaxes valables. Pour chacune de ces syntaxes, le résultat est une seule série temporelle.

- IF(TS ***Opérateur de comparaison*** S, S | TS, S | TS)

  > **Note**
  >
  > Si la valeur `TS comparison operator S` est VRAIE mais `metric2` qu'aucun point de données ne correspond, la sortie sera égale à 0.

- IF(TS, TS, TS)
- IF(TS, S, TS)
- IF(TS, TS, S)
- IF(TS, S, S)
- IF(S, TS, TS)

Les sections suivantes fournissent davantage de détails et d'exemples pour ces syntaxes.

IF(TS ***Opérateur de comparaison*** S, scalar2 | metric2, scalar3 | metric3)

Valeur de série temporelle en sortie correspondante :

- a la valeur de scalar2 ou metric2, si TS ***Opérateur de comparaison*** S est TRUE
- a la valeur de scalar3 ou metric3, si TS ***Opérateur de comparaison*** S est FALSE
- a la valeur 0 si l'***opérateur de comparaison*** TS est VRAI et que le point de données correspondant dans metric2 n'existe pas.
- a la valeur 0 si l'***opérateur de comparaison*** TS est FAUX et que le point de données correspondant dans metric3 n'existe pas.
- est une série temporelle vide, si le point de données correspondant n'existe pas dans metric3, ou si scalar3/metric3 est omis de l'expression

IF(metric1, metric2, metric3)

Pour chaque point de données de metric1, la valeur de série temporelle en sortie correspondante :

- a la valeur de metric2, si le point de données correspondant de metric1 est TRUE.
- a la valeur de metric3, si le point de données correspondant de metric1 est FALSE.
- a la valeur 0, si le point de données correspondant de metric1 est TRUE et le point de données correspondant n'existe pas dans metric2.
- est supprimé, si le point de données correspondant de metric1 est FALSE et que le point de données correspondant n'existe pas dans metric3
- est supprimé si le point de données correspondant de metric1 est FALSE et que metric3 est omis de l'expression.
- est supprimé si le point de données correspondant de metric1 est manquant.

Le tableau suivant présente un exemple de cette syntaxe.

Métrique ou fonction	Valeurs
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
(metric3)	[0, 0, 20, -, 20]

Métrique ou fonction	Valeurs
IF(metric1, metric2, metric3)	[30, 0, 20, 0, -]

IF(metric1, scalar2, metric3)

Pour chaque point de données de metric1, la valeur de série temporelle en sortie correspondante :

- a la valeur de scalar2, si le point de données correspondant de metric1 est TRUE.
- a la valeur de metric3, si le point de données correspondant de metric1 est FALSE.
- est supprimé, si le point de données correspondant de metric1 a la valeur FALSE et que le point de données correspondant n'existe pas dans metric3, ou si metric3 est omis de l'expression.

Métrique ou fonction	Valeurs
(metric1)	[1, 1, 0, 0, -]
scalar2	5
(metric3)	[0, 0, 20, -, 20]
IF(metric1, scalar2, metric3)	[5, 5, 20, -, -]

IF(metric1, metric2, scalar3)

Pour chaque point de données de metric1, la valeur de série temporelle en sortie correspondante :

- a la valeur de metric2, si le point de données correspondant de metric1 est TRUE.
- a la valeur de scalar3, si le point de données correspondant de metric1 est FALSE.
- a la valeur 0, si le point de données correspondant de metric1 est TRUE et le point de données correspondant n'existe pas dans metric2.
- est supprimé si le point de données correspondant dans metric1 n'existe pas.

Métrique ou fonction	Valeurs
(metric1)	[1, 1, 0, 0, -]
(metric2)	[30, -, 0, 0, 30]
scalar3	5
IF(metric1, metric2, scalar3)	[30, 0, 5, 5, -]

IF(scalar1, metric2, metric3)

Valeur de séries temporelles en sortie correspondante :

- a la valeur de metric2, si scalar1 est TRUE.
- a la valeur de metric3, si scalar1 est FALSE.
- est une série temporelle vide, si metric3 est omis de l'expression.

Exemples de cas d'utilisation des expressions IF

Les exemples suivants illustrent les utilisations possibles de la fonction IF.

- Pour afficher uniquement les valeurs basses d'une métrique :

  IF(metric1<400, metric1)
- Pour changer chaque point de données d'une métrique en l'une des deux valeurs, pour afficher les hauts et les bas relatifs de la métrique d'origine :

  IF(metric1<400, 10, 2)
- Pour afficher un 1 pour chaque horodatage où la latence est supérieure au seuil et afficher un 0 pour tous les autres points de données :

  IF(latence>seuil, 1, 0)

Utiliser les mathématiques métriques avec l'opération GetMetricData d'API

Vous pouvez utiliser `GetMetricData` pour effectuer des calculs à l'aide d'expressions mathématiques, ainsi que pour récupérer de grands lots de données de métriques en un seul appel d'API. Pour plus d'informations, consultez GetMetricData.

Détection d'anomalies sur les mathématiques appliquées aux métriques

La détection d'anomalies sur les mathématiques appliquées aux métriques est une fonction que vous pouvez utiliser pour créer des alertes de détection d'anomalies sur des métriques uniques et des sorties d'expressions mathématiques de métrique. Vous pouvez utiliser ces expressions pour créer des graphiques qui visualisent les canaux de détection d'anomalies. La fonction prend en charge les fonctions arithmétiques de base, les opérateurs logiques et de comparaison, et la plupart des autres fonctions.

La détection d'anomalies sur les mathématiques appliquées aux métriques ne prend pas en charge les fonctions suivantes :

- Expressions qui contiennent plusieurs ANOMALY_DETECTION_BAND dans la même ligne.
- Expressions qui contiennent plus de 10 métriques ou expressions mathématiques.
- Expressions qui contiennent l'expression METRICS (MÉTRIQUES).
- Expressions qui contiennent la fonction SEARCH (CHERCHER).
- Expressions utilisant la fonction DP_PERF_INSIGHTS.
- Expressions qui utilisent des métriques avec des périodes différentes.
- Détecteurs d'anomalies mathématiques métriques qui utilisent des métriques à haute résolution comme entrée.

Pour plus d'informations sur cette fonctionnalité, consultez la section Utilisation de la détection des CloudWatch anomalies dans le guide de l' CloudWatch utilisateur Amazon.

Utiliser des expressions de recherche dans les graphiques

Les expressions de recherche sont un type d'expression mathématique que vous pouvez ajouter aux CloudWatch graphiques. Les expressions de recherche vous permettent d'ajouter rapidement plusieurs métriques liées entre elles à un graphique. Elles vous permettent également de créer des

graphiques dynamiques qui ajoutent automatiquement les métriques appropriées à leur affichage, même si ces métriques n'existent pas lorsque vous créez le graphique.

Par exemple, vous pouvez créer une expression de recherche qui affiche la métrique `AWS/EC2` `CPUUtilization` pour toutes les instances de la région. Si, par la suite, vous lancez une nouvelle instance, la `CPUUtilization` de la nouvelle instance est automatiquement ajoutée au graphique.

Lorsque vous utilisez une expression de recherche dans un graphique, la recherche détecte l'expression de recherche dans les noms de métriques, les espaces de noms, les noms de dimensions et les valeurs de dimension. Vous pouvez utiliser des opérateurs booléens plus complexes et puissants pour les recherches. Une expression de recherche ne peut trouver que les métriques ayant généré des données au cours des deux dernières semaines.

Vous ne pouvez pas créer d'alerte basée sur l'expression SEARCH. En effet, les expressions de recherche renvoient plusieurs séries temporelles, et une alerte basée sur une expression mathématique ne peut regarder qu'une seule série temporelle.

Si vous utilisez un compte de surveillance dans le cadre de l'observabilité CloudWatch entre comptes, vos expressions de recherche peuvent trouver des statistiques dans les comptes sources liés à ce compte de surveillance.

Rubriques

CloudWatch syntaxe des expressions de recherche

Une expression de recherche valide dispose du format suivant.

```
SEARCH(' {Namespace, DimensionName1, DimensionName2, ...} SearchTerm', 'Statistic')
```

Par exemple :

```
SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')
```

- La première partie de la requête après le mot `SEARCH`, délimitée par des accolades, est le schéma de métrique à rechercher. Le schéma de métrique contient un espace de noms de métrique et

un ou plusieurs noms de dimension. L'inclusion d'un schéma de métrique dans une requête de recherche est facultative. Si cette valeur est spécifiée, le schéma de métrique doit contenir un espace de noms et peut éventuellement contenir un ou plusieurs noms de dimension qui sont valides dans cet espace de noms.

Vous n'avez pas besoin d'utiliser des guillemets à l'intérieur du schéma de métrique, sauf si un espace de noms ou un nom de dimension comprend des espaces ou des caractères non alphanumériques. Dans ce cas, vous devez placer le nom qui contient ces caractères dans des guillemets doubles.

- Le `SearchTerm` est également facultatif, mais une recherche valide doit contenir le schéma de métrique, le `SearchTerm`, ou les deux à la fois. Le `SearchTerm` contient généralement un ou plusieurs ID de compte, noms de métriques ou valeurs de dimension. Le `SearchTerm` peut inclure plusieurs termes à rechercher, à la fois par correspondance partielle et par correspondance exacte. Il peut également contenir des opérateurs booléens.

  L'utilisation d'un identifiant de compte dans un ne `SearchTerm` fonctionne que dans les comptes configurés comme comptes de surveillance à des fins d' CloudWatch observabilité entre comptes. La syntaxe pour un ID de compte dans un `SearchTerm` est `:aws.AccountId = "`*`444455556666`*`"`. Vous pouvez également utiliser `'LOCAL'` pour spécifier le compte de surveillance lui-même : `:aws.AccountId = 'LOCAL'`

  Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

  Le `SearchTerm` peut inclure un ou plusieurs identifiants, par exemple, `MetricName=` comme dans cet exemple, mais l'utilisation d'indicateurs n'est pas obligatoire.

  Le schéma de métrique et le `SearchTerm` doivent être placés dans des guillemets simples.

- `Statistic`Il s'agit du nom de toute CloudWatch statistique valide. Il doit être entouré de guillemets simples. Pour plus d'informations, consultez Statistiques.

L'exemple précédent effectue une recherche dans l'espace de noms `AWS/EC2` pour les métriques qui disposent de `InstanceId` comme nom de dimension. Elle renvoie toutes les métriques `CPUUtilization` trouvées, avec le graphique illustrant la statistique `Average`.

Une expression de recherche ne peut trouver que les métriques ayant généré des données au cours des deux dernières semaines.

Limites d'expression de recherche

La taille maximale de la requête d'expression de recherche est de 1 024 caractères. Vous pouvez avoir jusqu'à 100 expressions de recherche dans un même graphique. Un graphique peut afficher jusqu'à 500 séries temporelles.

CloudWatch expressions de recherche : Tokenisation

Lorsque vous spécifiez un`SearchTerm`, la fonction de recherche recherche des jetons, qui sont des sous-chaînes générées CloudWatch automatiquement à partir de noms de métriques complets, de noms de dimensions, de valeurs de dimension et d'espaces de noms. CloudWatch génère des jetons qui se distinguent par la majuscule en forme de chameau dans la chaîne d'origine. Les caractères numériques servent également comme le début de nouveaux jetons et les caractères non alphanumériques servent de délimiteurs, créant des jetons avant et après les caractères non alphanumériques.

Une chaîne continue du même type de caractère jeton délimiteur entraîne un jeton.

Tous les jetons sont générés en minuscules. Le tableau suivant montre des exemples de jetons générés.

Chaîne d'origine	Jetons générés
CustomCount1	`customcount1` , `custom`, `count`, `1`
SDBFailure	`sdbfailure` , `sdb`, `failure`
Project2-trial333	`project2trial333` , `project`, `2`, `trial`, `333`

CloudWatch expressions de recherche : Correspondances partielles

Lorsque vous spécifiez un`SearchTerm`, le terme de recherche est également tokenisé. CloudWatchtrouve des métriques sur la base de correspondances partielles, c'est-à-dire des correspondances entre un jeton unique généré à partir du terme de recherche et un jeton unique généré à partir d'un nom de métrique, d'un espace de noms, d'un nom de dimension ou d'une valeur de dimension.

Les recherches de correspondances partielles pour correspondre à un seul jeton ne sont pas sensibles à la casse. Par exemple, l'utilisation de n'importe quelle expression de recherche peut renvoyer la métrique `CustomCount1` :

- `count`
- `Count`
- `COUNT`

Cependant, l'utilisation de `couNT` en tant qu'expression de recherche ne trouvera pas `CustomCount1`, car la casse dans le terme recherché `couNT` provoque la création de jetons dans `cou` et `NT`.

Les recherches peuvent également correspondre à des jetons composites, qui sont plusieurs jetons qui apparaissent consécutivement dans le nom d'origine. La recherche est sensible à la casse pour faire correspondre un jeton composite. Par exemple, si le terme d'origine est `CustomCount1`, les recherches de `CustomCount` ou de `Count1` réussissent, mais les recherches de `customcount` ou de `count1` échouent.

CloudWatch expressions de recherche : correspondances exactes

Vous pouvez définir une recherche pour trouver uniquement des correspondances exactes de votre terme de recherche en utilisant des guillemets autour du terme recherché qui nécessite une correspondance exacte. Ces guillemets doubles sont entourés de guillemets simples utilisés par l'ensemble du terme recherché. Par exemple, **`SEARCH(' {MyNamespace}, "CustomCount1" ', 'Maximum')`** trouve la chaîne exacte `CustomCount1` si elle existe en tant que nom de dimension, nom de métrique ou valeur de dimension dans l'espace de noms nommé `MyNamespace`. Toutefois, la recherche **`SEARCH(' {MyNamespace}, "customcount1" ', 'Maximum')`** ou **`SEARCH(' {MyNamespace}, "Custom" ', 'Maximum')`** ne trouve pas cette chaîne.

Vous pouvez combiner des conditions de correspondance partielles et des conditions de correspondance exactes dans une seule expression de recherche. Par exemple,**`SEARCH(' {AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')`** renvoie la métrique Elastic Load Balancing nommée `ConsumedLCUs`, ainsi que toutes les métriques ou dimensions Elastic Load Balancing qui contiennent le jeton `flow`.

L'utilisation de la correspondance exacte est également un bon moyen pour trouver des noms dotés de caractères spéciaux, tels que les caractères non alphanumériques ou des espaces, comme dans l'exemple suivant.

```
SEARCH(' {"My Namespace", "Dimension@Name"}, "Custom:Name[Special_Characters" ',
 'Maximum')
```

CloudWatch expressions de recherche : exclusion d'un schéma métrique

Tous les exemples présentés jusqu'à présent incluent un schéma de métrique, en accolades. Les recherches qui omettent un schéma de métrique sont également valides.

Par exemple, **`SEARCH(' "CPUUtilization" ', 'Average')`** renvoie tous les noms de métriques, les noms de dimensions, les valeurs de dimension et les espaces de noms qui correspondent exactement à la chaîne `CPUUtilization`. Dans les espaces de noms des AWS métriques, cela peut inclure des métriques provenant de plusieurs services, notamment Amazon EC2, Amazon ECS SageMaker, etc.

Pour limiter cette recherche à un seul AWS service, la meilleure pratique consiste à spécifier l'espace de noms et toutes les dimensions nécessaires dans le schéma métrique, comme dans l'exemple suivant. Bien que cela réduise la recherche pour l'espace de noms `AWS/EC2`, il renvoie toujours des résultats d'autres métriques si vous avez défini `CPUUtilization` comme valeur de dimension pour ces métriques.

```
SEARCH(' {AWS/EC2, InstanceType} "CPUUtilization" ', 'Average')
```

Sinon, vous pouvez ajouter l'espace de noms dans le `SearchTerm`, comme illustré dans l'exemple suivant. Mais la recherche ne correspond à aucune chaîne `AWS/EC2` dans cet exemple, même si elle était un nom ou une valeur de dimension personnalisée.

```
SEARCH(' "AWS/EC2" MetricName="CPUUtilization" ', 'Average')
```

CloudWatch expressions de recherche : Spécification des noms de propriétés dans la recherche

La correspondance exacte de recherche suivante pour `"CustomCount1"` renvoie toutes les métriques dotées exactement de ce nom.

```
SEARCH(' "CustomCount1" ', 'Maximum')
```

Mais elle renvoie également des métriques avec des noms de dimension, des valeurs de dimension ou des espaces de noms de `CustomCount1`. Pour mieux structurer votre recherche, vous pouvez spécifier le nom de la propriété du type d'objet que vous souhaitez rechercher dans vos recherches. L'exemple suivant recherche tous les espaces de noms et renvoie les métriques nommées `CustomCount1`.

```
SEARCH(' MetricName="CustomCount1" ', 'Maximum')
```

Vous pouvez également utiliser des espaces de noms et des paires nom/valeur de dimension en tant que noms des propriétés, comme dans les exemples suivants. Le premier exemple illustre également que vous pouvez utiliser les noms de propriété avec des correspondances partielles de recherche.

```
SEARCH(' InstanceType=micro ', 'Average')
```

```
SEARCH(' InstanceType="t2.micro" Namespace="AWS/EC2" ', 'Average')
```

CloudWatch expressions de recherche : caractères non alphanumériques

Les caractères non alphanumériques servent de délimiteurs et indiquent où les noms des métriques, dimensions, espaces de noms et termes de recherche doivent être séparés en jetons. Lorsque les termes provoquent la création de jetons, les caractères non alphanumériques sont enlevés et n'apparaissent pas dans les jetons. Par exemple, `Network-Errors_2` génère les jetons `network`, `errors` et 2.

Votre terme de recherche peut inclure des caractères non alphanumériques. Si ces caractères apparaissent dans vos termes de recherche, ils peuvent spécifier des jetons composites dans une correspondance partielle. Par exemple, l'ensemble de recherches suivant trouverait les métriques nommées `Network-Errors-2` ou `NetworkErrors2`.

```
network/errors
network+errors
network-errors
Network_Errors
```

Lorsque vous faites une recherche de valeur exacte, les caractères non alphanumériques utilisés dans la recherche exacte doivent être les bons caractères qui apparaissent dans la chaîne recherchée. Par exemple, si vous voulez trouver `Network-Errors-2`, la recherche de `"Network-Errors-2"` aboutit, mais une recherche de `"Network_Errors_2"` échoue.

Lorsque vous effectuez une recherche de correspondance exacte, les caractères suivants doivent être placés dans une séquence d'échappement avec une barre oblique inverse.

```
" \ ( )
```

Par exemple, pour trouver le nom de la métrique `Europe\France Traffic(Network)` en correspondance exacte, utilisez le terme de recherche **`"Europe\\France Traffic\(Network\)"`**

CloudWatch expressions de recherche : opérateurs booléens

La recherche prend en charge l'utilisation des opérateurs booléens `AND`, `OR` et `NOT` dans le `SearchTerm`. Les opérateurs booléens sont entourés de guillemets simples que vous utilisez pour insérer l'ensemble du terme recherché. Les opérateurs booléens sont sensibles à la casse. Par conséquent, `and`, `or` et `not` ne sont pas valides comme opérateurs booléens.

Vous pouvez utiliser `AND` explicitement dans votre recherche, par exemple **`SEARCH('{AWS/EC2,InstanceId} network AND packets', 'Average')`**. Si vous n'utilisez aucun opérateur booléen entre les termes recherchés, cela implique une recherche de ceux-ci comme s'il y avait un opérateur `AND`, ce qui génère les mêmes résultats de recherche qu'avec **`SEARCH(' {AWS/EC2,InstanceId} network packets ', 'Average')`**.

Utilisez `NOT` pour exclure des résultats les sous-ensembles de données. Par exemple, **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT i-1234567890123456 ', 'Average')`** renvoie `CPUUtilization` pour toutes vos instances, à l'exception de l'instance `i-1234567890123456`. Vous pouvez également utiliser une clause `NOT` comme seul terme de recherche. Par exemple, **`SEARCH( 'NOT Namespace=AWS ', 'Maximum')`** génère toutes vos métriques personnalisées (métriques avec des espaces de noms qui n'incluent pas `AWS`).

Vous pouvez utiliser plusieurs expressions `NOT` dans une requête. Par exemple, **`SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" NOT "ProjectA" NOT "ProjectB" ', 'Average')`** renvoie l'élément `CPUUtilization` de toutes les instances de la région, à l'exception de celles ayant des valeurs de dimension `ProjectA` ou `ProjectB`.

Vous pouvez combiner des opérateurs booléens pour des recherches plus puissantes et détaillées, comme dans les exemples suivants. Utilisez des parenthèses pour regrouper les opérateurs.

Les deux exemples suivants renvoient tous les noms de métriques contenant `ReadOps` et provenant à la fois des espaces de noms EC2 et EBS.

```
SEARCH(' (EC2 OR EBS) AND MetricName=ReadOps ', 'Maximum')
```

```
SEARCH(' (EC2 OR EBS) MetricName=ReadOps ', 'Maximum')
```

L'exemple suivant restreint la recherche précédente pour trouver uniquement les résultats qui comprennent `ProjectA`, qui pourrait être la valeur d'une dimension.

```
SEARCH(' (EC2 OR EBS) AND ReadOps AND ProjectA ', 'Maximum')
```

L'exemple suivant utilise des groupes imbriqués. Elle renvoie les métriques Lambda pour les `Errors` de toutes les fonctions et les `Invocations` de fonctions avec des noms qui incluent les chaînes `ProjectA` ou `ProjectB`.

```
SEARCH(' {AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

CloudWatch expressions de recherche : utilisation d'expressions mathématiques

Vous pouvez utiliser une expression de recherche au sein d'une expression mathématique dans un graphique.

Par exemple, **`SUM(SEARCH(' {AWS/Lambda, FunctionName} MetricName="Errors" ', 'Sum'))`** renvoie la somme de la métrique `Errors` de toutes vos fonctions Lambda.

L'utilisation de lignes distinctes pour votre expression de recherche et votre expression mathématique peut générer des résultats plus utiles. Par exemple, supposons que vous utilisez les deux expressions suivantes dans un graphique. La première ligne affiche les lignes `Errors` distinctes pour chacune de vos fonctions Lambda. L'ID de cette expression est `e1`. La deuxième ligne ajoute une autre ligne illustrant la somme des erreurs provenant de toutes les fonctions.

```
SEARCH(' {AWS/Lambda, FunctionName}, MetricName="Errors" ', 'Sum')
SUM(e1)
```

CloudWatch exemples d'expressions de recherche

Les exemples suivants illustrent plus d'utilisations et syntaxes d'expressions de recherche. Commençons par une recherche `CPUUtilization` dans toutes les instances de la région, puis examinons les variations.

Cet exemple affiche une ligne pour chaque instance de la région, affichant la métrique `CPUUtilization` provenant de l'espace de noms `AWS/EC2`.

```
SEARCH(' {AWS/EC2,InstanceId} MetricName="CPUUtilization" ', 'Average')
```

Le changement de `InstanceId` à `InstanceType` modifie le graphique pour que celui-ci affiche une ligne pour chaque type d'instance utilisé dans la région. Les données de toutes les instances de chaque type sont regroupées sur une ligne pour ce type d'instance.

```
SEARCH(' {AWS/EC2,InstanceType} MetricName="CPUUtilization" ', 'Average')
```

L'exemple suivant regroupe les `CPUUtilization` par type d'instance et affiche une ligne pour chaque type d'instance qui inclut la chaîne `micro`.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType=micro MetricName="CPUUtilization" ',
 'Average')
```

Cet exemple rétrécit l'exemple précédent, modifiant ainsi `InstanceType` à une recherche exacte pour les instances t2.micro.

```
SEARCH('{AWS/EC2,InstanceType} InstanceType="t2.micro" MetricName="CPUUtilization" ',
 'Average')
```

La recherche suivant supprime la partie `{metric schema}` de la requête et, par conséquent, la métrique `CPUUtilization` de tous les espaces de noms s'affichant dans le graphique. Cela peut renvoyer un certain nombre de résultats car le graphique inclut plusieurs lignes pour la `CPUUtilization` métrique de chaque AWS service, agrégées selon différentes dimensions.

```
SEARCH('MetricName="CPUUtilization" ', 'Average')
```

Pour affiner davantage ces résultats, vous pouvez spécifier deux espaces de noms de métriques spécifiques.

```
SEARCH('MetricName="CPUUtilization" AND ("AWS/ECS" OR "AWS/ES") ', 'Average')
```

L'exemple précédent est le seul moyen d'effectuer une recherche de plusieurs espaces de noms avec une seule requête de recherche, comme vous ne pouvez spécifier qu'un seul schéma de métrique dans chaque requête. Toutefois, pour ajouter plus de structures, vous pouvez utiliser deux requêtes dans le graphique, comme dans l'exemple suivant. Cet exemple ajoute également plus de structure en spécifiant une dimension à utiliser pour regrouper les données Amazon ECS.

```
SEARCH('{AWS/ECS ClusterName}, MetricName="CPUUtilization" ', 'Average')
```

```
SEARCH(' {AWS/EBS} MetricName="CPUUtilization" ', 'Average')
```

Par exemple,`ConsumedLCUs` renvoie la métrique Elastic Load Balancing nommée , ainsi que toutes les métriques ou dimensions Elastic Load Balancing qui contiennent le jeton `flow`.

```
SEARCH('{AWS/NetworkELB, LoadBalancer} "ConsumedLCUs" OR flow ', 'Maximum')
```

L'exemple suivant utilise des groupes imbriqués. Elle renvoie les métriques Lambda pour les `Errors` de toutes les fonctions et les `Invocations` de fonctions avec des noms qui incluent les chaînes `ProjectA` ou `ProjectB`.

```
SEARCH('{AWS/Lambda,FunctionName} MetricName="Errors" OR (MetricName="Invocations" AND
 (ProjectA OR ProjectB)) ', 'Average')
```

L'exemple suivant affiche l'ensemble de vos métriques personnalisées, à l'exception des métriques générées par les services AWS .

```
SEARCH('NOT Namespace=AWS ', 'Average')
```

L'exemple suivant affiche les métriques avec les noms de métriques, les espaces de noms, les dimensions des noms et les valeurs de dimension qui contiennent la chaîne `Errors` dans leur nom.

```
SEARCH('Errors', 'Average')
```

L'exemple suivant restreint la recherche aux correspondances exactes. Par exemple, cette recherche détecte le nom de la métrique `Errors`, mais pas les métriques nommées `ConnectionErrors` ou `errors`.

```
SEARCH(' "Errors" ', 'Average')
```

L'exemple suivant montre comment spécifier les noms contenant des espaces ou des caractères spéciaux dans le schéma de métrique, dans le cadre du terme recherché.

```
SEARCH('{"Custom-Namespace", "Dimension Name With Spaces"}, ErrorCount ', 'Maximum')
```

CloudWatch exemples d'expressions de recherche d'observabilité entre comptes

CloudWatch exemples d'observabilité entre comptes

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 utiliser la fonction SEARCH pour renvoyer des métriques provenant de comptes sources spécifiés. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

L'exemple suivant récupère toutes les métriques Lambda du compte avec l'ID de compte 111122223333.

```
SEARCH(' AWS/Lambda :aws.AccountId = "111122223333" ', 'Average')
```

L'exemple suivant extrait toutes les métriques AWS/EC2 de deux comptes : 111122223333 et 777788889999.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR "777788889999") ', 'Average')
```

L'exemple suivant récupère toutes les métriques AWS/EC2 du compte source 111122223333 et du compte de surveillance lui-même.

```
SEARCH(' AWS/EC2 :aws.AccountId = ("111122223333" OR 'LOCAL') ', 'Average')
```

L'exemple suivant extrait la SUM de la métrique MetaDataToken du compte 444455556666 contenant la dimension InstanceId.

```
SEARCH('{AWS/EC2,InstanceId} :aws.AccountId=444455556666 MetricName=\"MetadataNoToken
\"','Sum')
```

Création d'un CloudWatch graphique avec une expression de recherche

Sur la CloudWatch console, vous pouvez accéder à la fonctionnalité de recherche lorsque vous ajoutez un graphique à un tableau de bord ou en utilisant la vue Metrics.

Vous ne pouvez pas créer une alerte basée sur une expression SEARCH. En effet, les expressions de recherche renvoient plusieurs séries temporelles et une alerte basée sur une expression mathématique ne peut regarder qu'une seule série temporelle.

Pour ajouter un graphique avec une expression de recherche à un tableau de bord existant

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), puis sélectionnez un tableau de bord.
3. Sélectionnez Add widget (Ajouter un widget).
4. Choisissez Line (Ligne) ou Stacked area (Zone empilée), puis Configure (Configurer).
5. Dans l'onglet Graphed metrics (Graphique des métriques), choisissez Add a math expression (Ajouter une expression mathématique).
6. Pour Details (Détails), saisissez l'expression de recherche que vous souhaitez. Par exemple, **`SEARCH('{AWS/EC2,InstanceId} MetricName="CPUUtilization"', 'Average')`**
7. (Facultatif) Pour ajouter une autre expression de recherche ou expression mathématique au graphique, choisissez Add a math expression (Ajouter une expression mathématique)
8. (Facultatif) Une fois que vous avez ajouté une expression de recherche, vous pouvez spécifier une étiquette dynamique à faire apparaître sur la légende du graphique pour chaque métrique. Les étiquettes dynamiques affichent une statistique sur la métrique et se mettent automatiquement à jour lorsque le tableau de bord ou le graphique est actualisé. Pour ajouter une étiquette dynamique, choisissez Graphed metrics (Graphique des métriques), puis Dynamic labels (Étiquettes dynamiques).

   Par défaut, les valeurs dynamiques que vous ajoutez à l'étiquette apparaissent au début de l'étiquette. Vous pouvez ensuite cliquer sur la valeur Label (Étiquette) de la métrique pour modifier l'étiquette. Pour plus d’informations, consultez Utiliser des étiquettes dynamiques.
9. (Facultatif) Pour ajouter une seule métrique au graphique, choisissez l'onglet All metrics (Toutes les métriques) et explorez jusqu'à trouver la métrique voulue.
10. (Facultatif) Pour modifier l'intervalle de temps indiqué sur le graphique, choisissez custom (personnalisé) en haut du graphique, ou choisissez l'une des périodes situées à gauche de custom (personnalisé).
11. (Facultatif) Les annotations horizontales permettent aux utilisateurs du tableau de bord de voir rapidement quand une métrique a des pics à un certain niveau, ou si la métrique se trouve dans une plage prédéfinie. Pour ajouter une annotation horizontale, choisissez Graph options (Options de graphique), puis Add horizontal annotation (Ajouter une annotation horizontale) :

    a. Pour Label (Étiquette), saisissez une étiquette pour l'annotation.
    b. Pour Value (Valeur), saisissez la valeur de métrique où apparaît l'annotation horizontale.
    c. Pour Fill (Remplissage), spécifiez s'il faut utiliser un ombrage de remplissage avec cette annotation. Par exemple, choisissez `Above` ou `Below` pour que la zone correspondante

soit remplie. Si vous spécifiez `Between`, un autre champ `Value` apparaît et la zone du graphique entre les deux valeurs est remplie.

d. Pour Axis (Axe), spécifiez si les nombres dans `Value` font référence à la métrique associée à l'axe des Y de gauche ou à l'axe des Y de droite si le graphique comprend plusieurs métriques.

Vous pouvez modifier la couleur de remplissage d'une annotation en choisissant le carré de couleur dans la colonne de gauche de cette annotation.

Répétez ces étapes pour ajouter plusieurs annotations horizontales au même graphique.

Pour masquer une annotation, décochez la case dans la colonne de gauche de cette annotation.

Pour supprimer une annotation, choisissez x dans la colonne Actions.

12. (Facultatif) Les annotations verticales vous aident à indiquer les étapes importantes dans un graphique, par exemple les événements opérationnels ou le début et la fin d'un déploiement. Pour ajouter une annotation verticale, choisissez Graph options (Options de graphique), puis Add vertical annotation (Ajouter des annotations verticales) :

a. Pour Label (Étiquette), saisissez une étiquette pour l'annotation. Pour afficher seulement la date et l'heure sur l'annotation, ne renseignez pas le champ Label (Étiquette).

b. Pour Date, spécifiez la date et l'heure de l'affichage de l'annotation verticale.

c. Pour Fill (Remplir), indiquez si l'ombrage de remplissage doit être utilisé avant ou après une annotation verticale, ou entre deux annotations verticales. Par exemple, choisissez `Before` ou `After` pour que la zone correspondante soit remplie. Si vous spécifiez `Between`, un autre champ `Date` apparaît et la zone du graphique entre les deux valeurs est remplie.

Répétez ces étapes pour ajouter plusieurs annotations verticales au même graphique.

Pour masquer une annotation, décochez la case dans la colonne de gauche de cette annotation.

Pour supprimer une annotation, choisissez x dans la colonne Actions.

13. Choisissez Create widget (Créer un widget).
14. Choisissez Save dashboard (Enregistrer le tableau de bord).

Pour utiliser l'affichage des métriques afin de tracer les métriques recherchées

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, saisissez les jetons à rechercher : par exemple, **`cpuutilization t2.small`**.

   Les résultats correspondant à votre recherche s'affichent.
4. Pour représenter graphiquement toutes les métriques correspondant à votre recherche, choisissez Graph search (Tracer la recherche).

   or

   Pour affiner votre recherche, choisissez l'un des espaces de noms qui apparaît dans vos résultats de recherche.
5. Si vous avez sélectionné un espace de noms pour affiner vos résultats, vous pouvez effectuer les opérations suivantes :

   a. Pour représenter graphiquement une ou plusieurs métriques, cochez la case en regard de chaque métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d'en-tête du tableau.

   b. Pour affiner votre recherche, passez la souris sur un nom de métrique et Add to search (Ajouter à la recherche) ou Search for this only (Rechercher uniquement ceci).

   c. Pour afficher l'aide relative à une métrique, sélectionnez le nom de la métrique, puis What is this? (De quoi s'agit-il ?).

   Les métriques sélectionnées apparaissent sur le graphique.
6. (Facultatif) Sélectionnez l'un des boutons dans la barre de recherche pour modifier cette partie du terme recherché.
7. (Facultatif) Pour ajouter le graphique à un tableau de bord, choisissez Actions, puis Add to dashboard (Ajouter au tableau de bord).

Obtention des statistiques d’une métrique

CloudWatch définitions des statistiques

Les statistiques sont des regroupements de données de métrique sur une période donnée. Lorsque vous tracez ou récupérez les statistiques d'une métrique, vous spécifiez le paramètre Period (Période), par exemple cinq minutes, à utiliser pour calculer chaque valeur statistique. Par exemple, si le champ Period (Période) est cinq minutes, le champSum (Somme) est la somme de toutes les valeurs de l'échantillon recueillies au cours de la période de cinq minutes, tandis que le paramètre Minimum est la valeur la plus faible recueillie au cours de la période de cinq minutes.

CloudWatch prend en charge les statistiques suivantes pour les métriques.

- SampleCountest le nombre de points de données au cours de la période.
- Sum (Somme) est la somme des valeurs de tous les points de données collectés au cours de la période.
- Average (Moyenne) est la valeur de `Sum/SampleCount` pendant la période spécifiée.
- Minimum est la valeur la plus basse observée pendant la période spécifiée.
- Maximum est la valeur la plus haute observée pendant la période spécifiée.
- Percentile (p) (Centile [p]) indique la position relative d'une valeur dans un jeu de données. Par exemple, p95 est le 95e centile et signifie que 95 % les données de cette période sont inférieures à cette valeur et que 5 % des données lui sont supérieures. Les centiles vous permettent de mieux comprendre la distribution des données de vos métriques.
- Trimmed mean (TM) (Moyenne ajustée [TM]) est la moyenne de toutes les valeurs situées entre deux limites spécifiées. Les valeurs en dehors des limites sont ignorées lorsque la moyenne est calculée. Vous définissez les limites comme un ou deux nombres compris entre 0 et 100, jusqu'à 10 décimales. Les nombres peuvent être des valeurs absolues ou des pourcentages. Par exemple, tm90 calcule la moyenne après avoir supprimé les 10 % des points de données avec les valeurs les plus élevées. TM(2 %:98 %) calcule la moyenne après avoir supprimé les 2 % de points de données les plus bas et les 2 % de points de données les plus élevés. TM(150:1000) calcule la moyenne après avoir supprimé tous les points de données inférieurs ou égaux à 150, ou supérieurs à 1 000.
- Interquartile mean (IQM) (Moyenne interquartile [IQM]) est la moyenne ajustée de la plage interquartile, ou le milieu 50 % des valeurs. Elle est équivalente à TM(25 %:75 %).
- Winsorized mean (WM) (Moyenne winsorisée [WM]) est similaire à la moyenne ajustée. Cependant, avec la moyenne winsorisée, les valeurs qui sont en dehors de la limite ne sont pas

ignorées, mais sont considérées comme égales à la valeur au bord de la limite appropriée. Après cette normalisation, la moyenne est calculée. Vous définissez les limites comme un ou deux nombres compris entre 0 et 100, jusqu'à 10 décimales. Par exemple, wm98 calcule la moyenne tout en traitant les 2 % des valeurs les plus élevées pour être égal à la valeur au 98e percentile. WM(10 %:90 %) calcule la moyenne tout en traitant les 10 % les plus élevés des points de données comme étant la valeur de la limite de 90 %, et en traitant les 10 % les plus bas des points de données comme étant la valeur de la limite de 10 %.

- Percentile rank (PR) (Rang centile [PR]) est le pourcentage de valeurs qui atteignent un seuil fixe. Par exemple, PR(:300) renvoie le pourcentage de points de données dont la valeur est inférieure ou égale à 300. PR(100:2000) renvoie le pourcentage de points de données dont la valeur est comprise entre 100 et 2 000.

  Le classement par percentile est exclusif dans la borne inférieure et inclusif dans la limite supérieure.

- Trimmed count (Nombre ajusté [TC]) est le nombre de points de données dans la plage choisie pour une statistique moyenne ajustée. Par exemple, tc90 renvoie le nombre de points de données, sans compter les points de données qui se situent dans les 10 % les plus élevés des valeurs. TC(0,005:0,030) renvoie le nombre de points de données avec des valeurs comprises entre 0,005 (exclusif) et 0,030 (inclus).

- Trimmed sum (Somme ajustée [TS]) est la somme des valeurs des points de données dans la plage choisie pour une statistique moyenne ajustée. Elle est équivalente à (Moyenne ajustée) * (Nombre ajusté). Par exemple, ts90 renvoie la somme des points de données, sans compter les points de données qui se situent dans les 10 % les plus élevés des valeurs. TS(80 %:) renvoie la somme des valeurs de point de données, sans compter les points de données dont les valeurs se situent dans les 80 % les plus bas de la plage de valeurs.

> **Note**
>
> Pour la moyenne ajustée, le nombre ajusté, la somme ajustée et la moyenne winsorisée, si vous définissez deux limites en tant que valeurs fixes au lieu de pourcentages, le calcul inclut des valeurs égales à la limite supérieure, mais n'inclut pas les valeurs égales à la limite inférieure.

Syntaxe

Pour la moyenne ajustée, le nombre ajusté, la somme ajustée et la moyenne winsorisée, les règles de syntaxe suivantes s'appliquent :

- L'utilisation de parenthèses avec un ou deux nombres sans signe de pourcentage définit les limites à utiliser comme valeurs dans le jeu de données qui se trouvent entre les deux centiles que vous spécifiez. Par exemple, TM(10 %:90 %) utilise uniquement les valeurs comprises entre le 10e et le 90e centiles. TM(:95 %) utilise les valeurs de l'extrémité la plus basse des données définies jusqu'au 95e centile, en ignorant les 5 % des points de données avec les valeurs les plus élevées.
- L'utilisation de parenthèses avec un ou deux nombres sans signe de pourcentage définit les limites à utiliser comme valeurs dans le jeu de données qui se trouvent entre les valeurs explicites que vous spécifiez. Par exemple, TC(80:500) utilise uniquement les valeurs comprises entre 80 (exclusive) et 500 (inclusive). TC(:0,5) utilise uniquement les valeurs qui sont égales ou inférieures à 0,5.
- L'utilisation d'un nombre sans parenthèses calcule à l'aide de pourcentages, en ignorant les points de données supérieurs au centile spécifié. Par exemple, tm99 calcule la moyenne tout en ignorant 1 % des points de données avec la valeur la plus élevée. Il s'agit de la même valeur que TM(:99 %).
- La moyenne ajustée, le nombre ajusté, la somme ajustée et la moyenne winsorisée peuvent toutes être abrégées à l'aide de lettres majuscules lors de la spécification d'une plage, telle que TM(5 %:95 %), TM(100:200) ou TM(:95 %). Ils ne peuvent être abrégés qu'en minuscules lorsque vous spécifiez un seul nombre, tel que tm99.

Cas d'utilisation des statistiques

- Trimmed mean (Moyenne ajustée) est particulièrement utile pour les métriques ayant une taille d'échantillon importante, comme la latence de page Web. Par exemple, tm99 ne tient pas compte des valeurs aberrantes extrêmement élevées qui pourraient résulter de problèmes de réseau ou d'erreurs humaines, afin de donner un nombre plus précis pour la latence moyenne des requêtes typiques. De même, TM(10 %:)ne tient pas compte des 10 % les plus faibles des valeurs de latence, telles que celles résultant des accès au cache. Et TM(10 %:99 %) exclut ces deux types de valeurs aberrantes. Nous vous recommandons d'utiliser une moyenne ajustée pour surveiller la latence.

- Il est conseillé de surveiller le nombre de valeurs ajustées chaque fois que vous utilisez la moyenne ajustée, pour vous assurer que le nombre de valeurs utilisées dans vos calculs de moyenne ajustée est suffisant pour être statistiquement significatif.
- Le rang de centile vous permet de placer des valeurs dans des « casiers » de plages, et vous pouvez l'utiliser pour créer manuellement un histogramme. Pour ce faire, décomposez vos valeurs en plusieurs casiers, tels que PR(:1), PR(1:5), PR(5:10) et PR(10:). Placez chacun de ces casiers dans une visualisation sous forme de graphiques à barres, et vous obtenez un histogramme.

  Le classement par percentile est exclusif dans la borne inférieure et inclusif dans la limite supérieure.

Centiles par rapport à la moyenne ajustée

Un centile, tel que p99, et une moyenne ajustée, telle que tm99, mesurent des valeurs similaires, mais pas identiques. p99 et tm99 ignorent 1 % des points de données avec les valeurs les plus élevées, qui sont considérés comme des valeurs aberrantes. Puis, p99 est la valeur maximale des 99 % restants, tandis que tm99 est la moyenne des 99 % restants. Si vous regardez la latence des requêtes Web, p99 vous indique la pire expérience client, en ignorant les valeurs aberrantes, tandis que tm99 vous indique l'expérience client moyenne, en ignorant les valeurs aberrantes.

La moyenne ajustée est une bonne statistique de latence à surveiller si vous cherchez à optimiser votre expérience client.

Exigences relatives à l'utilisation de centiles, de la moyenne ajustée et d'autres statistiques

CloudWatch a besoin de points de données bruts pour calculer les statistiques suivantes :

- Centiles
- Moyenne ajustée
- Moyenne interquartile
- Moyenne winsorisée
- Somme ajustée
- Nombre ajustée
- Rang de centile

Si vous publiez des données pour des statistiques personnalisées à l'aide d'un jeu de statistiques au lieu de données brutes, vous pouvez récupérer ces types de statistiques pour ces données uniquement si l'une des conditions suivantes est vraie :

- La SampleCount valeur de l'ensemble de statistiques est 1 et Min, Max et Sum sont tous égaux.
- Les valeurs Min et Max sont égales, et Sum est égale à Min multiplié par SampleCount.

Les AWS services suivants incluent des mesures qui prennent en charge ces types de statistiques.

- API Gateway
- Application Load Balancer
- Amazon EC2
- Elastic Load Balancing
- Kinesis
- Amazon RDS

De plus, ces types de statistiques sur les centiles ne sont pas disponibles pour les métriques lorsque l'une des valeurs des métriques est un nombre négatif.

Les exemples suivants vous montrent comment obtenir des statistiques pour les CloudWatch métriques relatives à vos ressources, telles que vos instances EC2.

Exemples

- Obtenir des statistiques pour une ressource spécifique
- Regrouper des statistiques sur des ressources
- Regroupement de statistiques par groupe Auto Scaling
- Regrouper des statistiques par Amazon Machine Image (AMI)

Obtenir des statistiques pour une ressource spécifique

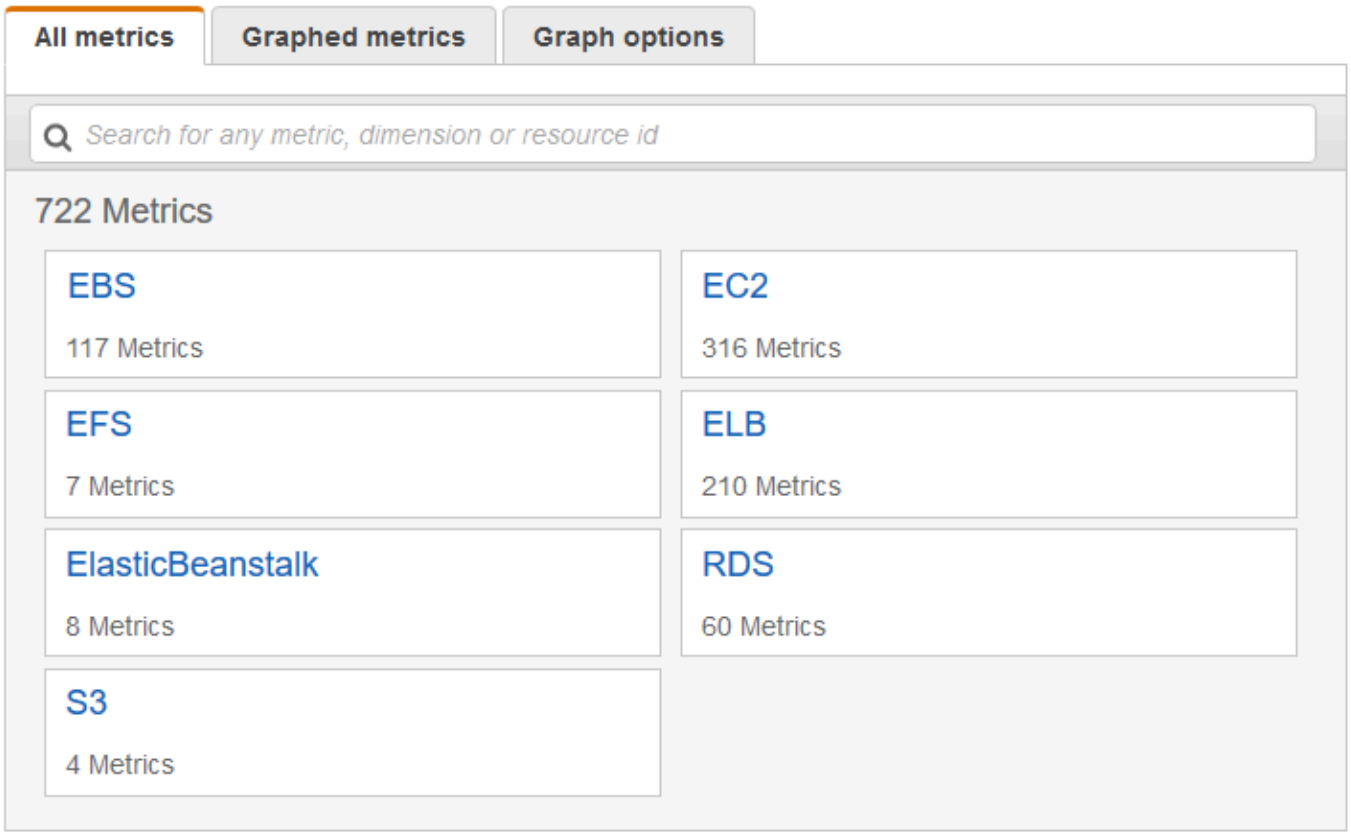
L'exemple suivant vous montre comment déterminer l'utilisation maximale de l'UC d'une instance EC2 spécifique.

Prérequis

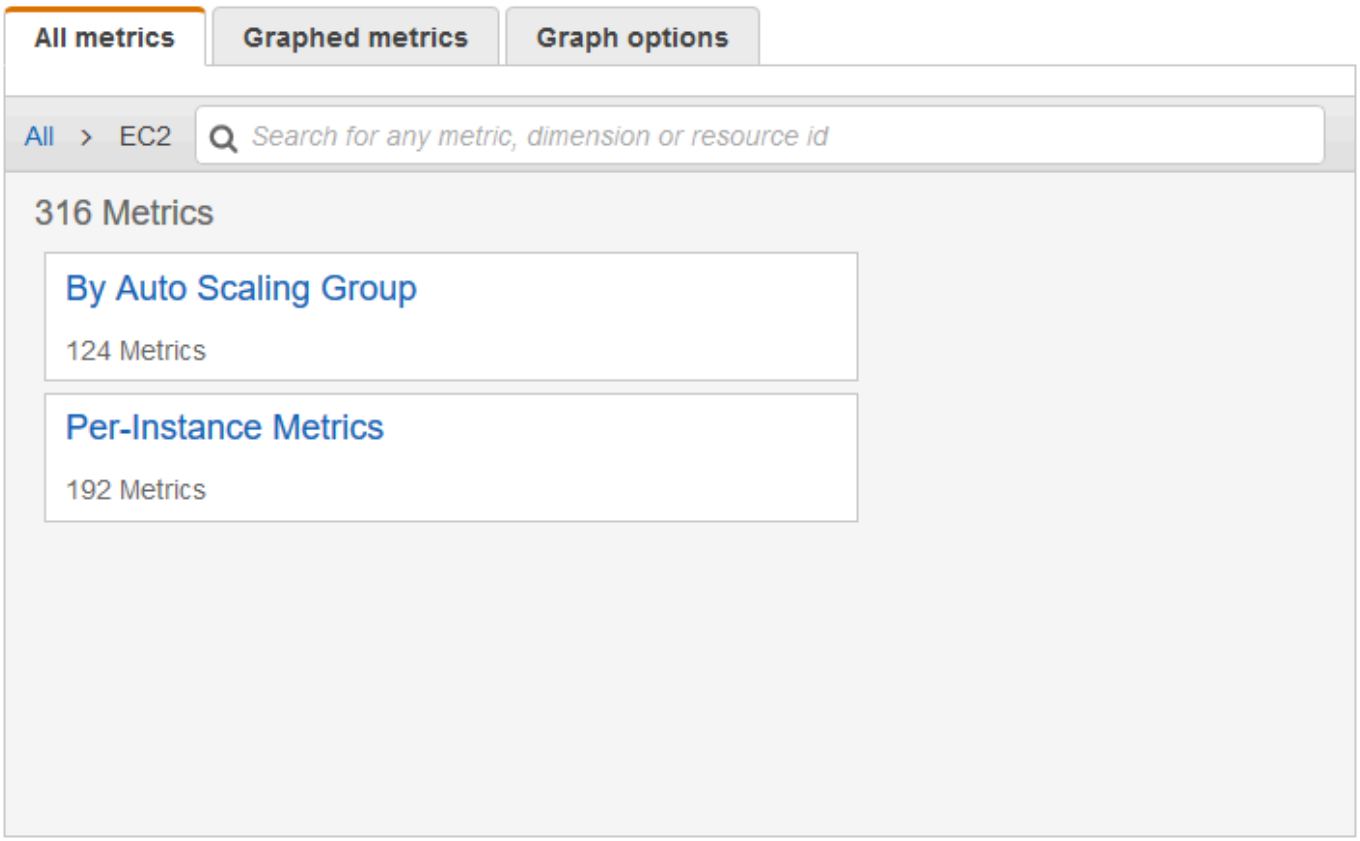
- Vous devez avoir l'ID de l'instance. Vous pouvez obtenir l'ID d'instance à l'aide de la console Amazon EC2 ou de la commande describe-instances.
- Par défaut, la surveillance basique est activée, mais vous pouvez activer la surveillance détaillée. Pour plus d'informations, consultez Enable or Disable Detailed Monitoring for Your Instances (Activer ou désactiver la surveillance détaillée pour vos instances) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Afficher l'utilisation moyenne de l'UC pour une instance spécifique à l'aide de la console

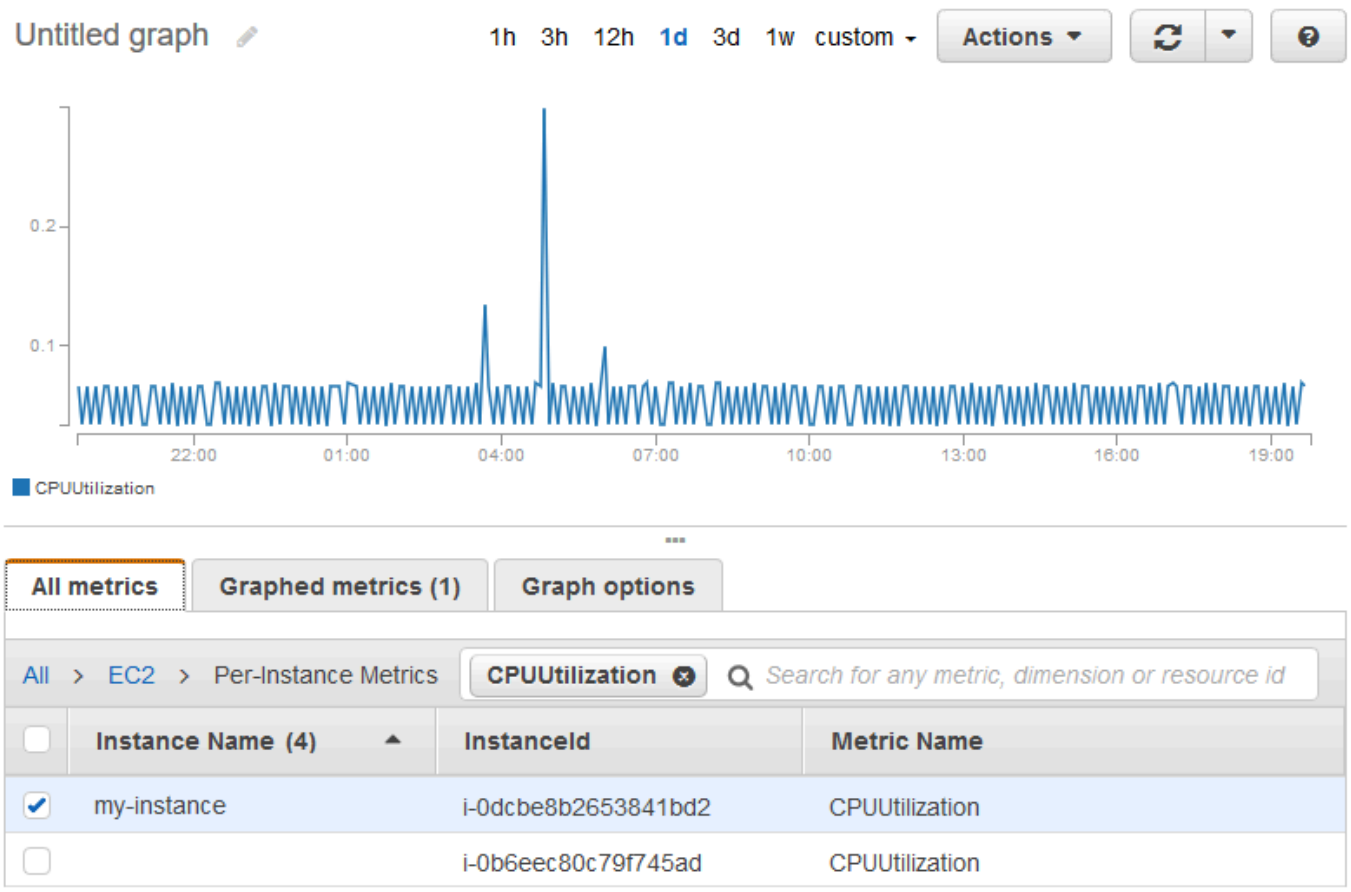
1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques).
3. Sélectionnez l'espace de noms de métrique EC2.



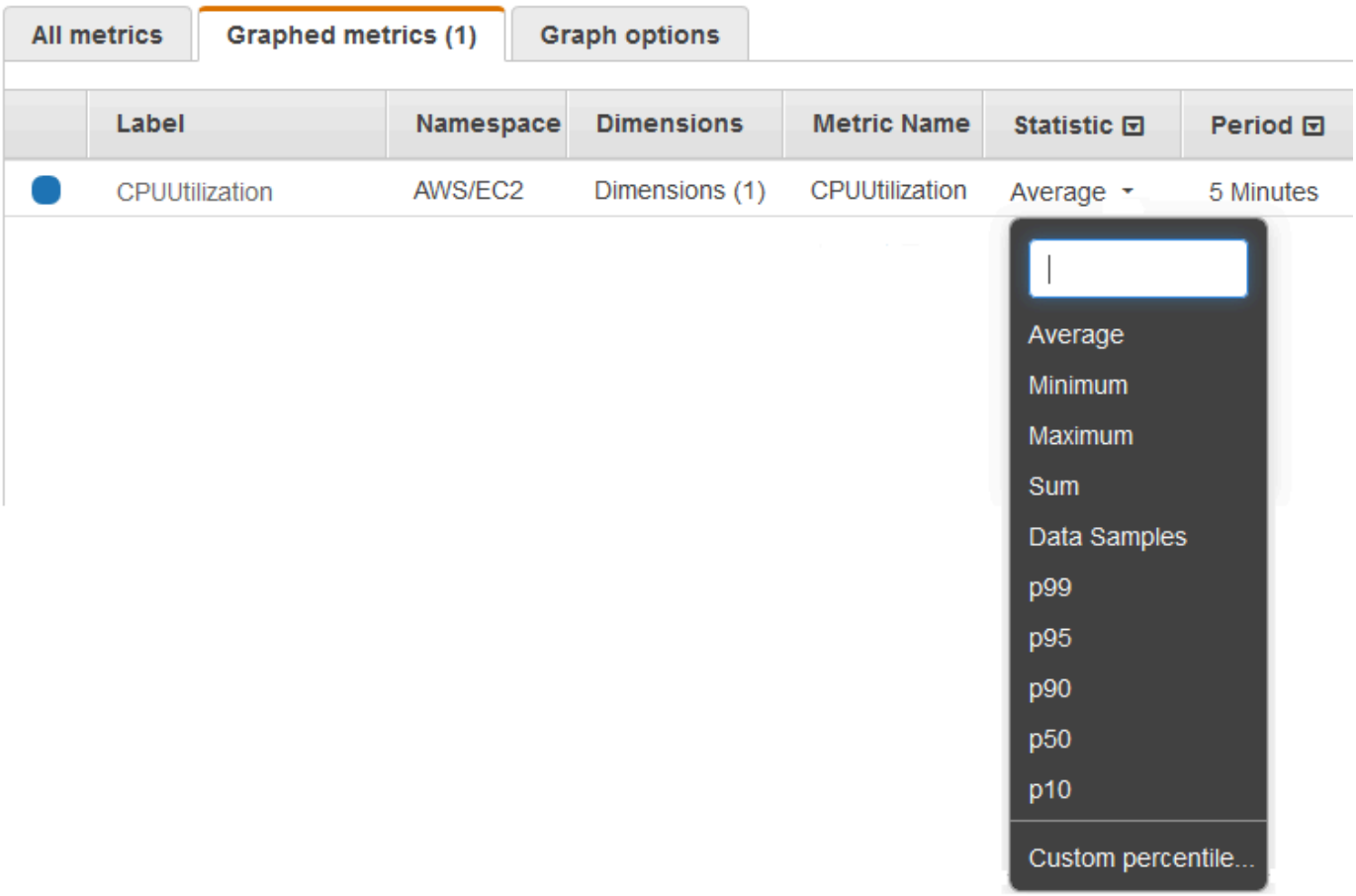
4. Sélectionnez la dimension Per-Instance Metrics (Métriques par instance).



5. Dans le champ de recherche, entrez **`CPUUtilization`**, puis appuyez sur Entrée. Sélectionnez la ligne de l'instance spécifique qui affiche un graphique pour la métrique `CPUUtilization` pour l'instance. Pour modifier le nom du graphique, choisissez l'icône représentant un crayon. Pour modifier la plage de temps, sélectionnez l'une des valeurs prédéfinies ou choisissez custom (personnalisé).



6. Pour modifier les statistiques, choisissez l'onglet Graphed metrics (Graphique des métriques). Choisissez l'en-tête de colonne ou une valeur individuelle, puis choisissez l'une des statistiques ou des centiles prédéfinis, ou bien spécifiez un centile personnalisé (par exemple, **`p99.999`**).



7. Pour modifier la période, choisissez l'onglet Graphed metrics (Graphique des métriques). Sélectionnez l'en-tête de colonne ou une valeur individuelle et choisissez une autre valeur.

Pour obtenir l'utilisation du processeur par instance EC2 à l'aide du AWS CLI

Utilisez la get-metric-statisticscommande comme suit pour obtenir la `CPUUtilization` métrique pour l'instance spécifiée.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=InstanceId,Value=i-1234567890abcdef0 --statistics Maximum \
--start-time 2016-10-18T23:18:00 --end-time 2016-10-19T23:18:00 --period 360
```

Les statistiques renvoyées sont des valeurs de six minutes pour l'intervalle de 24 heures demandé. Chaque valeur représente le pourcentage maximal d'utilisation de l'UC pour l'instance spécifiée au cours d'une période de six minutes donnée. Les points de données ne sont pas renvoyés dans l'ordre chronologique. Voici une illustration du début de l'exemple de sortie (la sortie complète comprend

des points de données correspondant à tous les intervalles de six minutes au cours de la période de 24 heures).

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-19T00:18:00Z",
            "Maximum": 0.33000000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T03:18:00Z",
            "Maximum": 99.670000000000002,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-19T07:18:00Z",
            "Maximum": 0.34000000000000002,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Regrouper des statistiques sur des ressources

Vous pouvez agréger les mesures relatives AWS aux ressources de plusieurs ressources. Les métriques sont complètement séparées entre les régions, mais vous pouvez utiliser les calculs de métriques pour agréger des métriques similaires entre les régions. Pour plus d'informations, consultez Utilisation des mathématiques appliquées aux métriques.

Par exemple, vous pouvez regrouper des statistiques pour vos instances EC2 qui ont la surveillance détaillée activée. Les instances qui utilisent une surveillance basique ne sont pas incluses. Par conséquent, vous devez activer la surveillance détaillée (à un coût supplémentaire), qui fournit des données par périodes de 1 minute. Pour plus d'informations, consultez Enable or Disable Detailed Monitoring for Your Instances (Activer ou désactiver la surveillance détaillée pour vos instances) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

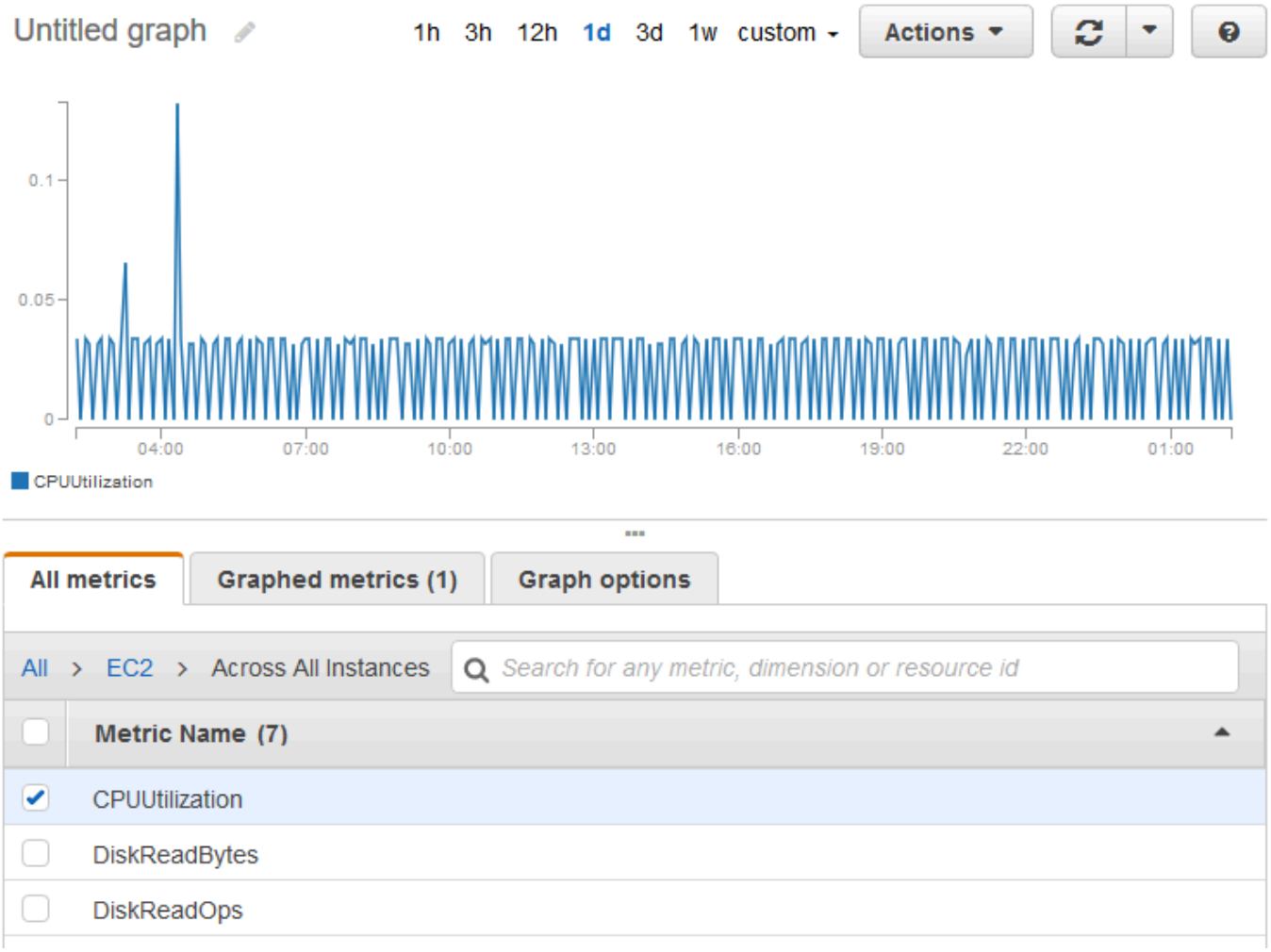
Cet exemple vous montre comment obtenir l'utilisation moyenne de l'UC pour vos instances EC2. Aucune dimension n'étant spécifiée, CloudWatch renvoie des statistiques pour toutes les dimensions

de l'espace de `AWS/EC2` noms. Pour obtenir les statistiques d'autres métriques, consultez la page AWS services qui publient CloudWatch des statistiques.

> **Important**
>
> Cette technique de récupération de toutes les dimensions d'un espace de AWS noms ne fonctionne pas pour les espaces de noms personnalisés dans lesquels vous publiez. CloudWatch Avec les espaces de noms personnalisés, vous devez spécifier l'ensemble complet des dimensions associées à un point de données particulier pour pouvoir extraire les statistiques qui incluent le point de données.

Afficher l'utilisation moyenne de l'UC pour vos instances EC2

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Choisissez l'espace de noms EC2, puis Across All Instances (Sur toutes les instances).
4. Sélectionnez la ligne contenant `CPUUtilization` qui affiche un graphique pour la métrique pour toutes vos instances EC2. Pour modifier le nom du graphique, choisissez l'icône représentant un crayon. Pour modifier la plage de temps, sélectionnez l'une des valeurs prédéfinies ou choisissez custom.



5. Pour modifier les statistiques, choisissez l'onglet Graphed metrics (Graphique des métriques). Choisissez l'en-tête de colonne ou une valeur individuelle, puis choisissez l'une des statistiques ou des centiles prédéfinis, ou bien spécifiez un centile personnalisé (par exemple, **`p95.45`**).
6. Pour modifier la période, choisissez l'onglet Graphed metrics (Graphique des métriques). Sélectionnez l'en-tête de colonne ou une valeur individuelle et choisissez une autre valeur.

Pour obtenir l'utilisation moyenne du processeur sur vos instances EC2 à l'aide du AWS CLI

Utilisez la commande get-metric-statistics comme suit :

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization
 --statistics "Average" "SampleCount" \
--start-time 2016-10-11T23:18:00 --end-time 2016-10-12T23:18:00 --period 3600
```

Voici un exemple de sortie :

```
{
    "Datapoints": [
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-12T07:18:00Z",
            "Average": 0.038235294117647062,
            "Unit": "Percent"
        },
        {
            "SampleCount": 240.0,
            "Timestamp": "2016-10-12T09:18:00Z",
            "Average": 0.16670833333333332,
            "Unit": "Percent"
        },
        {
            "SampleCount": 238.0,
            "Timestamp": "2016-10-11T23:18:00Z",
            "Average": 0.041596638655462197,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```


Regroupement de statistiques par groupe Auto Scaling

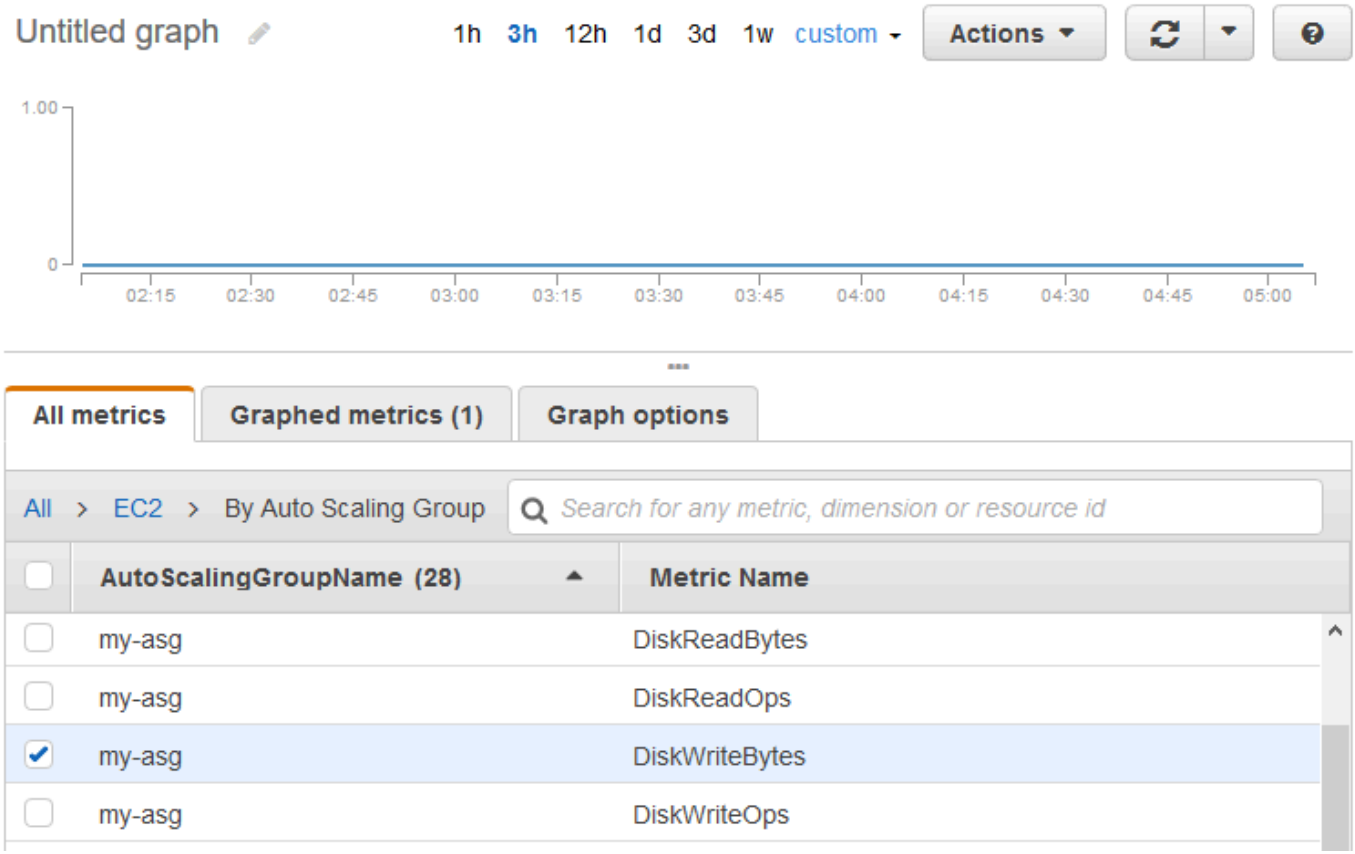
Vous pouvez regrouper des statistiques pour les instances EC2 dans un groupe Auto Scaling. Les métriques sont complètement distinctes entre les régions, mais vous pouvez utiliser les mathématiques des CloudWatch métriques pour agréger et transformer les métriques de plusieurs régions. Vous pouvez également utiliser le tableau de bord multi-comptes pour effectuer des calculs sur les métriques provenant de différents comptes.

Cet exemple vous montre comment obtenir le nombre total d'octets écrit sur disque pour un groupe Auto Scaling. Le total est calculé par durée d'une minute sur une période de 24 heures pour toutes les instances EC2 dans le groupe Auto Scaling spécifié.

DiskWriteBytes Pour afficher les instances d'un groupe Auto Scaling à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques).

3. Choisissez l’espace de noms EC2, puis choisissez By Auto Scaling Group (Par groupe Auto Scaling).
4. Sélectionnez la ligne pour la DiskWriteBytesmétrique et le groupe Auto Scaling spécifique, qui affiche un graphique pour la métrique pour les instances du groupe Auto Scaling. Pour modifier le nom du graphique, choisissez l'icône représentant un crayon. Pour modifier la plage de temps, sélectionnez l'une des valeurs prédéfinies ou choisissez custom (personnalisé).



5. Pour modifier les statistiques, choisissez l'onglet Graphed metrics (Graphique des métriques). Choisissez l'en-tête de colonne ou une valeur individuelle, puis choisissez l'une des statistiques ou des centiles prédéfinis, ou bien spécifiez un centile personnalisé (par exemple, **`p95.45`**).
6. Pour modifier la période, choisissez l'onglet Graphed metrics (Graphique des métriques). Sélectionnez l'en-tête de colonne ou une valeur individuelle et choisissez une autre valeur.

DiskWriteBytes Pour obtenir les instances d'un groupe Auto Scaling à l'aide du AWS CLI

Utilisez la commande get-metric-statistics comme suit.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name DiskWriteBytes
--dimensions Name=AutoScalingGroupName,Value=my-asg --statistics "Sum" "SampleCount" \
```

```
--start-time 2016-10-16T23:18:00 --end-time 2016-10-18T23:18:00 --period 360
```

Voici un exemple de sortie.

```
{
    "Datapoints": [
        {
            "SampleCount": 18.0,
            "Timestamp": "2016-10-19T21:36:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        },
        {
            "SampleCount": 5.0,
            "Timestamp": "2016-10-19T21:42:00Z",
            "Sum": 0.0,
            "Unit": "Bytes"
        }
    ],
    "Label": "DiskWriteBytes"
}
```

Regrouper des statistiques par Amazon Machine Image (AMI)

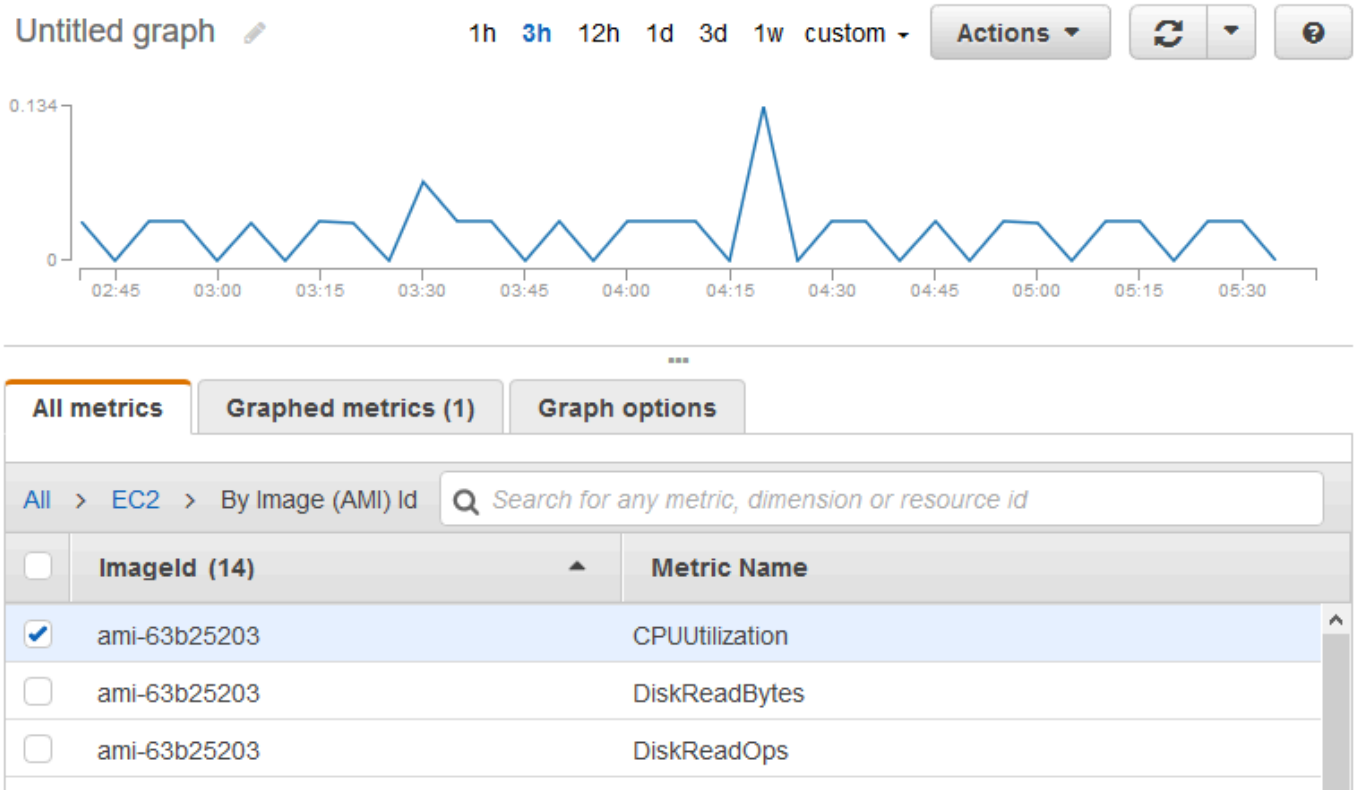
Vous pouvez regrouper des statistiques pour les instances EC2 dont la surveillance détaillée est activée. Les instances qui utilisent une surveillance basique ne sont pas incluses. Pour plus d'informations, consultez Enable or Disable Detailed Monitoring for Your Instances (Activer ou désactiver la surveillance détaillée pour vos instances) dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Cet exemple vous montre comment déterminer l'utilisation moyenne de l'UC pour toutes les instances qui utilisent l'AMI spécifiée. La moyenne est calculée par intervalles de 60 secondes pour une période d'un jour.

Afficher l'utilisation moyenne de l'UC par AMI à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Metrics (Métriques).
3. Choisissez l'espace de noms EC2, puis choisissez By Image (AMI) Id (Par ID d'image [AMI]).

4. Sélectionnez la ligne de la métrique `CPUUtilization` et l'AMI spécifique, qui affiche un graphique pour la métrique pour l'AMI spécifiée. Pour modifier le nom du graphique, choisissez l'icône représentant un crayon. Pour modifier la plage de temps, sélectionnez l'une des valeurs prédéfinies ou choisissez custom (personnalisé).



5. Pour modifier les statistiques, choisissez l'onglet Graphed metrics (Graphique des métriques). Choisissez l'en-tête de colonne ou une valeur individuelle, puis choisissez l'une des statistiques ou des centiles prédéfinis, ou bien spécifiez un centile personnalisé (par exemple, **`p95.45`**).
6. Pour modifier la période, choisissez l'onglet Graphed metrics (Graphique des métriques). Sélectionnez l'en-tête de colonne ou une valeur individuelle et choisissez une autre valeur.

Pour obtenir l'utilisation moyenne du processeur par AMI à l'aide du AWS CLI

Utilisez la commande get-metric-statistics comme suit.

```
aws cloudwatch get-metric-statistics --namespace AWS/EC2 --metric-name CPUUtilization \
--dimensions Name=ImageId,Value=ami-3c47a355 --statistics Average \
--start-time 2016-10-10T00:00:00 --end-time 2016-10-11T00:00:00 --period 3600
```

L'opération retourne les statistiques qui sont des valeurs d'une heure pour un intervalle d'un jour. Chaque valeur représente le pourcentage d'utilisation moyenne de l'UC pour les instances EC2 exécutant l'AMI spécifiée. Voici un exemple de sortie.

```
{
    "Datapoints": [
        {
            "Timestamp": "2016-10-10T07:00:00Z",
            "Average": 0.041000000000000009,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T14:00:00Z",
            "Average": 0.079579831932773085,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2016-10-10T06:00:00Z",
            "Average": 0.036000000000000011,
            "Unit": "Percent"
        },
        ...
    ],
    "Label": "CPUUtilization"
}
```

Publier des métriques personnalisées

Vous pouvez publier vos propres statistiques à CloudWatch l'aide de l'API AWS CLI ou d'une API. Vous pouvez consulter les graphiques statistiques de vos statistiques publiées à l'aide du AWS Management Console.

CloudWatch stocke les données relatives à une métrique sous la forme d'une série de points de données. Chaque point de données comporte un horodatage associé. Vous pouvez même publier un ensemble regroupé de points de données appelé un ensemble de statistiques.

Rubriques

- Métriques haute résolution
- Utiliser les dimensions
- Publier des points de données uniques

- Publier des ensembles de statistiques
- Publier la valeur zéro
- Arrêter la publication des métriques

Métriques haute résolution

Chaque métrique appartient à l'une des catégories suivantes :

- Résolution standard, avec des données dont la granularité est d'une minute
- Haute résolution, avec des données dont la granularité est d'une seconde

Les métriques produites par les AWS services ont une résolution standard par défaut. Lorsque vous publiez une métrique personnalisée, vous pouvez la définir en tant que résolution standard ou haute résolution. Lorsque vous publiez une métrique haute résolution, que vous la CloudWatch stockez avec une résolution de 1 seconde, et vous pouvez la lire et la récupérer sur une période de 1 seconde, 5 secondes, 10 secondes, 30 secondes ou un multiple de 60 secondes.

Les métriques haute résolution peuvent vous donner des informations immédiates sur l'activité de votre application sur une période inférieure à une minute. Gardez à l'esprit que chaque appel `PutMetricData` pour des métriques personnalisées est facturé, donc des appels `PutMetricData` plus fréquents sur une métrique haute résolution peut entraîner des frais plus élevés. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Si vous définissez une alerte sur une métrique haute résolution, vous pouvez spécifier une alerte haute résolution avec une période de 10 secondes ou de 30 secondes ou vous pouvez définir une alerte régulière avec une période correspondant à n'importe quel multiple de 60 secondes. Les frais sont plus élevés pour les alertes haute résolution dont la période est de 10 ou 30 secondes.

Utiliser les dimensions

Dans les métriques personnalisées, le paramètre `--dimensions` est commun. Une dimension explicite davantage la nature de la métrique et les données qu'elle stocke. Vous pouvez avoir jusqu'à 30 dimensions affectées à une métrique, et chaque dimension est définie par une paire nom et valeur.

La manière de préciser une dimension varie selon les commandes que vous utilisez. Avec put-metric-data, vous spécifiez chaque dimension comme *MyName*= *MyValue*, et avec get-metric-statisticsou put-metric-alarmvous utilisez le format `Name=` *MyName*, `Value=` *MyValue*. Par exemple,

la commande suivante publie une métrique `Buffers` comptant deux dimensions nommées `InstanceId` et `InstanceType`.

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace MyNameSpace --unit
 Bytes --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small
```

Cette commande récupère des statistiques concernant cette même métrique. Vous devez utiliser des virgules pour séparer les parties Nom et Valeur d'une dimension, mais vous utilisez un espace entre les dimensions, s'il y en a plusieurs.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --namespace MyNameSpace --
dimensions Name=InstanceId,Value=1-23456789 Name=InstanceType,Value=m1.small --start-
time 2016-10-15T04:00:00Z --end-time 2016-10-19T07:00:00Z --statistics Average --period
 60
```

Si une seule métrique inclut plusieurs dimensions, vous devez spécifier une valeur pour chaque dimension définie lorsque vous l'utilisez get-metric-statistics. Par exemple, la métrique Amazon S3 `BucketSizeBytes` inclut les dimensions `BucketName` et `StorageType` vous devez donc spécifier les deux dimensions avec get-metric-statistics.

```
aws cloudwatch get-metric-statistics --metric-name BucketSizeBytes --start-time
 2017-01-23T14:23:00Z --end-time 2017-01-26T19:30:00Z --period 3600 --namespace
 AWS/S3 --statistics Maximum --dimensions Name=BucketName,Value=MyBucketName
 Name=StorageType,Value=StandardStorage --output table
```

Vous pouvez voir les dimensions définies pour une métrique grâce à la commande list-metrics.

Publier des points de données uniques

Pour publier un point de données unique pour une métrique nouvelle ou existante, utilisez la put-metric-datacommande avec une seule valeur et un horodatage. Par exemple, les actions suivantes publient chacune un point de données.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 2 --timestamp 2016-10-20T12:00:00.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 4 --timestamp 2016-10-20T12:00:01.000Z
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService --
value 5 --timestamp 2016-10-20T12:00:02.000Z
```

Si vous appelez cette commande avec un nouveau nom de métrique, CloudWatch crée une métrique pour vous. Dans le cas contraire, CloudWatch associe vos données à la métrique existante que vous avez spécifiée.

> **Note**
>
> Lorsque vous créez une métrique, cela peut prendre jusqu'à 2 minutes avant que vous puissiez récupérer les statistiques de la nouvelle métrique à l'aide de la get-metric-statisticscommande. Cependant, l'affichage de la nouvelle métrique dans la liste des métriques récupérées à l'aide de la commande list-metrics peut prendre jusqu'à 15 minutes.

Bien que vous puissiez publier des points de données avec des horodatages aussi précis qu'un millième de seconde, il agrège les données à une CloudWatch granularité minimale d'une seconde. CloudWatch enregistre la moyenne (somme de tous les éléments divisée par le nombre d'éléments) des valeurs reçues pour chaque période, ainsi que le nombre d'échantillons, la valeur maximale et la valeur minimale pour la même période. Par exemple, la métrique `PageViewCount` des exemples précédents contient trois points de données avec des horodatages à quelques secondes les uns des autres. Si vos règles sont définies sur 1 minute, CloudWatch agrège les trois points de données car ils sont tous horodatés sur une période d'une minute.

Vous pouvez utiliser la commande get-metric-statistics pour extraire des statistiques basées sur les points de données que vous avez publiés.

```
aws cloudwatch get-metric-statistics --namespace MyService --metric-name PageViewCount
 \
--statistics "Sum" "Maximum" "Minimum" "Average" "SampleCount" \
--start-time 2016-10-20T12:00:00.000Z --end-time 2016-10-20T12:05:00.000Z --period 60
```

Voici un exemple de sortie.

```
{
    "Datapoints": [
        {
            "SampleCount": 3.0,
            "Timestamp": "2016-10-20T12:00:00Z",
            "Average": 3.6666666666666665,
            "Maximum": 5.0,
            "Minimum": 2.0,
```

```
            "Sum": 11.0,
            "Unit": "None"
        }
    ],
    "Label": "PageViewCount"
}
```

Publier des ensembles de statistiques

Vous pouvez agréger vos données avant de les publier sur CloudWatch. Lorsque vous avez plusieurs points de données par minute, les regroupements permettent de réduire le nombre d'appels à put-metric-data. Par exemple, au lieu d'appeler put-metric-data plusieurs fois pour trois points de données espacés de trois secondes maximum, vous pouvez regrouper les données dans un ensemble de statistiques que vous publiez en un seul appel, en utilisant le paramètre `--statistic-values`.

```
aws cloudwatch put-metric-data --metric-name PageViewCount --namespace MyService
 --statistic-values Sum=11,Minimum=2,Maximum=5,SampleCount=3 --
timestamp 2016-10-14T12:00:00.000Z
```

CloudWatch a besoin de points de données bruts pour calculer les percentiles. Si, au lieu de cela, vous publiez des données avec un ensemble de statistiques, vous ne pouvez pas récupérer de statistiques relatives aux percentiles pour ces données si aucune des conditions suivantes n'est réunie :

- La valeur `SampleCount` de l'ensemble de statistiques est égale à 1.
- Les valeurs `Minimum` et `Maximum` de l'ensemble de statistiques sont égales.

Publier la valeur zéro

Lorsque vos données sont plus sporadiques et que vous avez des périodes sans aucune donnée, vous pouvez choisir de publier la valeur zéro (`0`) pour cette période ou aucune valeur du tout. Si vous utilisez des appels réguliers à `PutMetricData` pour surveiller l'état de votre application, vous souhaiterez peut-être publier le chiffre zéro au lieu d'une absence de valeur. Par exemple, vous pouvez définir une CloudWatch alarme pour vous avertir si votre application ne publie pas de statistiques toutes les cinq minutes. Cette application doit alors publier des zéros pour les périodes sans donnée.

Vous pouvez également publier des zéros si vous souhaitez effectuer le suivi du nombre total de points de données ou si vous souhaitez que des statistiques de type minimum ou moyenne incluent des points de données avec la valeur 0.

Arrêter la publication des métriques

Pour arrêter de publier des métriques personnalisées sur CloudWatch, modifiez le code de votre application ou de votre service pour arrêter de l'utiliser PutMetricData. CloudWatch n'extrait pas les métriques des applications, il ne reçoit que ce qui lui est envoyé. Pour arrêter de publier vos métriques, vous devez donc les arrêter à la source.

Utilisation des CloudWatch alarmes Amazon

Vous pouvez créer des alarmes métriques et composites dans Amazon CloudWatch.

- Une alarme métrique surveille une seule CloudWatch métrique ou le résultat d'une expression mathématique basée sur CloudWatch des métriques. L'alarme réalise une ou plusieurs actions en fonction de la valeur de la métrique ou de l'expression par rapport à un seuil sur un certain nombre de périodes. L'action peut consister à envoyer une notification à une rubrique Amazon SNS, à exécuter une action Amazon EC2 ou Amazon EC2 Auto Scaling, ou à créer un OpsItem incident ou dans Systems Manager.
- Une alerte composite contient une expression de règle qui prend en compte les états d'alerte des autres alertes que vous avez créées. L'alerte composite passe à l'état ALARM uniquement si toutes les conditions de la règle sont remplies. Les alertes spécifiées dans l'expression de règle d'alerte composite peuvent inclure des alertes de métrique et d'autres alertes composites.

  L'utilisation d'alertes composites peut réduire le bruit d'alerte. Vous pouvez créer plusieurs alertes de métrique, mais aussi créer une alerte composite et configurer des alertes uniquement pour l'alerte composite. Par exemple, un composite peut passer à l'état ALARM uniquement lorsque toutes les alertes de métrique sous-jacentes sont à l'état ALARM.

  Les alarmes composites peuvent envoyer des notifications Amazon SNS lorsqu'elles changent d'état, et peuvent créer des Systems Manager OpsItems ou des incidents lorsqu'elles passent en état ALARM, mais ne peuvent pas effectuer d'actions EC2 ou d'actions Auto Scaling.

> **Note**
>
> Vous pouvez créer autant d'alarmes que vous le souhaitez dans votre AWS compte.

Vous pouvez ajouter des alarmes aux tableaux de bord afin de surveiller et de recevoir des alertes concernant vos AWS ressources et applications dans plusieurs régions. Une fois que vous avez ajouté une alerte à un tableau de bord, elle devient grise lorsque son état est `INSUFFICIENT_DATA`, et elle devient rouge quand son état est `ALARM`. L'alerte n'a pas de couleur lorsqu'elle se trouve à l'état `OK`.

Vous pouvez également ajouter aux favoris les alarmes récemment visitées à l'aide de l'option Favoris et récents du volet de navigation de la CloudWatch console. L'option Favorites and recents

(Favoris et récents) se compose de deux colonnes pour vos alertes favorites et les alertes que vous avez récemment consultées.

Une alerte appelle des actions uniquement lorsque l'état de l'alerte change. L'exception concerne les alertes avec des actions Auto Scaling. Dans le cas d'actions Auto Scaling, l'alerte continue d'appeler l'action une fois par minute pendant laquelle elle reste dans le nouvel état.

Une alerte peut surveiller une métrique dans le même compte. Si vous avez activé la fonctionnalité multi-comptes sur votre CloudWatch console, vous pouvez également créer des alarmes qui surveillent les statistiques d'autres AWS comptes. La création d'alertes composites entre comptes n'est pas prise en charge. La création d'alertes croisées qui utilisent des expressions mathématiques est prise en charge, sauf que les fonctions `ANOMALY_DETECTION_BAND`, `INSIGHT_RULE`, et `SERVICE_QUOTA` ne sont pas prises en charge pour les alertes de compte croisé.

> **Note**
>
> CloudWatch ne teste ni ne valide les actions que vous spécifiez, et ne détecte aucune erreur Amazon EC2 Auto Scaling ou Amazon SNS résultant d'une tentative d'appel d'actions inexistantes. Vérifiez que vos actions d'alerte existent.

États d'alerte de métrique

Une alerte de métrique peut avoir les états suivants :

- `OK` – La métrique ou l'expression se trouve dans le seuil défini.
- `ALARM` – La métrique ou l'expression se trouve à l'extérieur du seuil défini.
- `INSUFFICIENT_DATA` – L'alerte vient de commencer, la métrique n'est pas disponible, ou la quantité de données n'est pas suffisante pour permettre à la métrique de déterminer le statut de l'alerte.

Évaluation d'une alerte

Lorsque vous créez une alarme, vous spécifiez trois paramètres à activer CloudWatch pour évaluer quand modifier l'état de l'alarme :

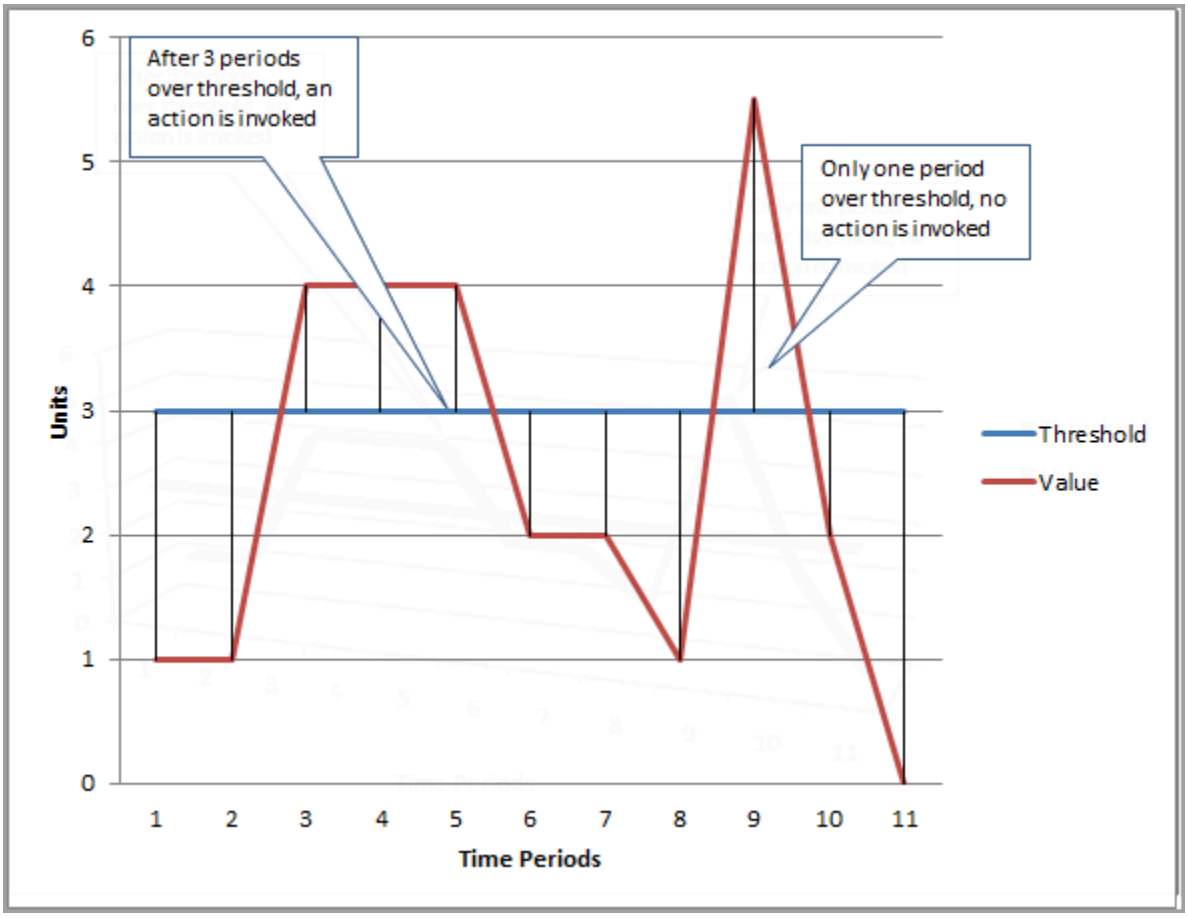
- La Période est la durée nécessaire pour évaluer la métrique ou l'expression afin de créer chaque point de données pour une alerte. Elle est exprimée en secondes.

- Evaluation Periods (Périodes d'évaluation) est le nombre de périodes, ou de points de données, les plus récents à évaluer pour déterminer l'état de l'alerte.
- Datapoints to Alarm (Points de données avant l'alerte) est le nombre de points de données pendant les périodes d'évaluation qui doit être dépassé pour que l'alerte passe à l'état ALARM. Les points de données au-delà du seuil n'ont pas besoin d'être consécutifs, mais ils doivent simplement tous correspondre au dernier nombre de points de données correspondant à la valeur Evaluation Period (Période d'évaluation).

Pour toute période d'une minute ou plus, une alerte est évaluée toutes les minutes et l'évaluation est basée sur la fenêtre de temps définie par la Période et les Périodes d'évaluation. Par exemple, si la Période est de 5 minutes (300 secondes) et que les Périodes d'évaluation sont de 1, alors à la fin de la cinquième minute, l'alerte est évaluée en fonction des données des minutes 1 à 5. Ensuite, à la fin de la minute 6, l'alerte est évaluée en fonction des données des minutes 2 à 6.

Si la durée de l'alerte est de 10 secondes ou 30 secondes, l'alerte est évaluée toutes les 10 secondes.

Dans la figure suivante, le seuil d'alerte d'une alerte de métrique est défini sur trois unités. Evaluation Period (Période d'évaluation) et Datapoints to Alarm (Points de données à l'alerte)sont définis sur 3. Cela signifie que lorsque les trois points de données des trois périodes consécutives les plus récentes sont au-dessus du seuil, l'alerte passe à l'état ALARM. Dans le schéma, cela se produit entre la troisième et la cinquième période. À la sixième période, la valeur repasse sous le seuil. L'une des périodes évaluées n'est donc pas en dépassement et l'état de l'alerte revient à l'état OK. Au cours de la neuvième période, le seuil est dépassé à nouveau, mais pendant une seule période. Par conséquent, le statut de l'alerte reste OK.



Lorsque vous configurez différentes valeurs pour Evaluation Periods (Périodes d'évaluation) et Datapoints to Alarm (Points de données avant l'alerte), vous définissez une alerte « M sur N ». Datapoints à Alarm (Points de données avant l'alerte) est (« M ») et Evaluation Periods (Périodes d'évaluation) est (« N »). L'intervalle d'évaluation correspond au nombre de périodes d'évaluation multiplié par la durée de la période. Par exemple, si vous configurez 4 points de données sur 5 avec une période de 1 minute, l'intervalle d'évaluation est de 5 minutes. Si vous configurez 3 points de données sur 3 avec une période de 10 minutes, l'intervalle d'évaluation est de 30 minutes.

> **Note**
>
> Si des points de données sont manquants peu après la création d'une alarme et que la métrique a été signalée CloudWatch avant que vous ne créiez l'alarme, CloudWatch récupère les points de données les plus récents avant la création de l'alarme lors de l'évaluation de l'alarme.

Actions d'alerte

Vous pouvez spécifier les actions d'une alerte lorsqu'elle change d'état entre les états OK, ALARM et INSUFFICIENT_DATA.

La plupart des actions peuvent être définies pour la transition vers chacun des trois états. À l'exception des actions Auto Scaling, elles se produisent uniquement lors des transitions d'état et ne sont pas exécutées à nouveau si la condition persiste pendant plusieurs heures ou jours. Vous pouvez utiliser le fait que plusieurs actions sont autorisées pour qu'une alerte envoie un e-mail lorsqu'un seuil est dépassé, puis un autre lorsque la condition de dépassement prend fin. Cela vous permet de vérifier que vos actions de mise à l'échelle ou de récupération sont déclenchées au moment prévu et fonctionnent comme vous le souhaitez.

Les actions d’alarme suivantes sont prises en charge.

- Notifier un ou plusieurs abonnés à l’aide d’une rubrique Amazon Simple Notification Service. Les abonnés peuvent aussi bien être des applications que des personnes. Pour plus d'informations sur Amazon SNS, consultez Qu'est-ce qu'Amazon SNS ?.
- Invoquer une fonction Lambda. C’est le moyen le plus simple d’automatiser des actions personnalisées en cas de modification de l’état des alarmes.
- Les alarmes basées sur des métriques EC2 peuvent également effectuer des actions EC2, tels que l’arrêt, la résiliation, le redémarrage ou la récupération d’une instance EC2. Pour plus d’informations, consultez Création d'alarmes qui arrêtent, mettent hors service, redémarrent ou récupèrent une instance EC2.
- Les alarmes peuvent également effectuer des actions pour mettre à l’échelle un groupe Auto Scaling. Pour plus d'informations, consultez Procédures et stratégies de mise à l'échelle simples pour Amazon EC2 Auto Scaling.
- Les alarmes peuvent être OpsItems créées dans le Systems Manager Ops Center ou créer des incidents dans AWS Systems Manager Incident Manager. Ces actions ne sont exécutées que lorsque l'alerte passe à l'état ALARM. Pour plus d'informations, consultez Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

Actions d’alarme Lambda

CloudWatch alarm garantit un appel asynchrone de la fonction Lambda pour un changement d'état donné, sauf dans les cas suivants :

- Lorsque la fonction n'existe pas.
- When n' CloudWatch est pas autorisé à invoquer la fonction Lambda.

Si CloudWatch vous ne parvenez pas à joindre le service Lambda ou si le message est rejeté pour une autre raison, CloudWatch réessayez jusqu'à ce que l'appel aboutisse. Lambda met le message en file d'attente et gère les nouvelles tentatives d'exécution. Pour plus d'informations sur ce modèle d'exécution, notamment sur la manière dont Lambda gère les erreurs, consultez la section Invocation asynchrone dans le guide du développeur. AWS Lambda

Vous pouvez appeler une fonction Lambda dans le même compte ou dans d'autres AWS comptes.

Lorsque vous spécifiez une alarme pour invoquer une fonction Lambda en tant qu'action d'alarme, vous pouvez choisir de spécifier le nom de la fonction, son alias ou une version spécifique d'une fonction.

Lorsque vous spécifiez une fonction Lambda comme action d'alarme, vous devez créer une politique de ressources pour la fonction afin de permettre au principal du CloudWatch service d'invoquer la fonction.

Pour ce faire, vous pouvez utiliser le AWS CLI, comme dans l'exemple suivant :

```
aws lambda add-permission \
--function-name my-function-name \
--statement-id AlarmAction \
--action 'lambda:InvokeFunction' \
--principal lambda.alarms.cloudwatch.amazonaws.com \
--source-account 111122223333 \
--source-arn arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name
```

Vous pouvez également créer une politique similaire à l'un des exemples suivants, puis l'attribuer à la fonction.

L'exemple suivant spécifie sur quel compte se trouve l'alarme, de sorte que seules les alarmes de ce compte (111122223333) peuvent invoquer la fonction.

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

```
        "Sid": "AlarmAction",
        "Effect": "Allow",
        "Principal": {
            "Service": "lambda.alarms.cloudwatch.amazonaws.com"
        },
        "Action": "lambda:InvokeFunction",
        "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
        "Condition": {
            "StringEquals": {
                "AWS:SourceAccount": "111122223333"
            }
        }
    }]
}
```

L'exemple suivant a un champ d'application plus restreint, ne permettant qu'à l'alarme spécifiée dans le compte indiqué d'invoquer la fonction.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "AlarmAction",
      "Effect": "Allow",
      "Principal": {
        "Service": "lambda.alarms.cloudwatch.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:444455556666:function:function-name",
      "Condition": {
        "StringEquals": {
          "AWS:SourceAccount": "111122223333",
          "AWS:SourceArn": "arn:aws:cloudwatch:us-east-1:111122223333:alarm:alarm-name"
        }
      }
    }]
}
```

Nous ne recommandons pas la création d'une politique ne spécifiant aucun compte source, car ce type de politique est vulnérable aux problèmes d'adjoints confus.

Objet d'événement envoyé depuis CloudWatch Lambda

Lorsque vous configurez une fonction Lambda en tant qu'action d'alarme, CloudWatch fournit une charge utile JSON à la fonction Lambda lorsqu'elle l'appelle. Cette charge utile JSON sert d'objet d'événement pour la fonction. Vous pouvez extraire des données de cet objet JSON et les utiliser dans votre fonction. Voici un exemple d'objet d'événement provenant d'une alarme de métrique.

```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:444455556666:alarm:lambda-demo-metric-
alarm',
  'accountId': '444455556666',
  'time': '2023-08-04T12:36:15.490+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'lambda-demo-metric-alarm',
    'state': {
      'value': 'ALARM',
      'reason': 'test',
      'timestamp': '2023-08-04T12:36:15.490+0000'
    },
    'previousState': {
      'value': 'INSUFFICIENT_DATA',
      'reason': 'Insufficient Data: 5 datapoints were unknown.',
      'reasonData':
 '{"version":"1.0","queryDate":"2023-08-04T12:31:29.591+0000","statistic":"Average","period":60
[],"threshold":5.0,"evaluatedDatapoints":[{"timestamp":"2023-08-04T12:30:00.000+0000"},
{"timestamp":"2023-08-04T12:29:00.000+0000"},
{"timestamp":"2023-08-04T12:28:00.000+0000"},
{"timestamp":"2023-08-04T12:27:00.000+0000"},
{"timestamp":"2023-08-04T12:26:00.000+0000"}]}',
      'timestamp': '2023-08-04T12:31:29.595+0000'
    },
    'configuration': {
      'description': 'Metric Alarm to test Lambda actions',
      'metrics': [
        {
          'id': '1234e046-06f0-a3da-9534-EXAMPLEe4c',
          'metricStat': {
            'metric': {
              'namespace': 'AWS/Logs',
              'name': 'CallCount',
              'dimensions': {
```

```
                'InstanceId': 'i-12345678'
              }
            },
            'period': 60,
            'stat': 'Average',
            'unit': 'Percent'
          },
          'returnData': True
        }
      ]
    }
  }
}
```

Voici un exemple d'objet d'événement provenant d'une alarme composite.

```
{
  'source': 'aws.cloudwatch',
  'alarmArn': 'arn:aws:cloudwatch:us-east-1:111122223333:alarm:SuppressionDemo.Main',
  'accountId': '111122223333',
  'time': '2023-08-04T12:56:46.138+0000',
  'region': 'us-east-1',
  'alarmData': {
    'alarmName': 'CompositeDemo.Main',
    'state': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:56:46.138+0000'
    },
    'previousState': {
      'value': 'ALARM',
      'reason': 'arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild transitioned to ALARM at Friday 04
 August, 2023 12:54:46 UTC',
      'reasonData': '{"triggeringAlarms":[{"arn":"arn:aws:cloudwatch:us-
east-1:111122223333:alarm:CompositeDemo.FirstChild","state":
{"value":"ALARM","timestamp":"2023-08-04T12:54:46.138+0000"}}]}',
      'timestamp': '2023-08-04T12:54:46.138+0000',
```

```
      'actionsSuppressedBy': 'WaitPeriod',
      'actionsSuppressedReason': 'Actions suppressed by WaitPeriod'
    },
    'configuration': {
      'alarmRule': 'ALARM(CompositeDemo.FirstChild) OR
 ALARM(CompositeDemo.SecondChild)',
      'actionsSuppressor': 'CompositeDemo.ActionsSuppressor',
      'actionsSuppressorWaitPeriod': 120,
      'actionsSuppressorExtensionPeriod': 180
    }
  }
}
```

Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes

Parfois, tous les points de données attendus pour une métrique ne sont pas signalés CloudWatch. Cela peut par exemple se produire lorsqu'une connexion est perdue, lorsqu'un serveur tombe en panne ou lorsqu'une métrique, de par sa conception, rapporte les données de façon intermittente uniquement.

CloudWatch vous permet de spécifier comment traiter les points de données manquants lors de l'évaluation d'une alarme. Cela vous aide à configurer votre alerte afin qu'elle passe à l'état `ALARM` uniquement lorsque cela s'avère approprié pour le type de données surveillées. Vous pouvez éviter les faux positifs lorsque les données manquantes n'indiquent pas de problème.

De la même manière que chaque alarme est toujours dans l'un des trois états suivants, chaque point de données spécifique signalé CloudWatch appartient à l'une des trois catégories suivantes :

- Non dépassé (seuil respecté)
- Dépassé (au-delà du seuil)
- Manquant

Pour chaque alarme, vous pouvez spécifier CloudWatch de traiter les points de données manquants comme suit :

- `notBreaching` : les points de données manquants sont traités comme étant corrects et dans les limites du seuil

- `breaching` : les points de données manquants sont traités comme étant incorrects au-delà du seuil
- `ignore` : l'état de l'alerte actuel est conservé
- `missing` : si tous les points de données de la plage d'évaluation des alertes sont manquants, l'alerte passe à INSUFFICIENT_DATA.

Le meilleur choix dépend du type de métrique et de l'objectif de l'alarme. Par exemple, si vous créez une alarme d'annulation d'application à l'aide d'une métrique qui rapporte des données en permanence, vous souhaiterez peut-être considérer les points de données manquants comme une violation, car cela peut indiquer un problème. En revanche, dans le cas d'une métrique qui génère des points de données uniquement en cas d'erreur, par exemple la métrique `ThrottledRequests` dans Amazon DynamoDB, vous traiteriez plutôt les données manquantes comme étant `notBreaching`. Le comportement par défaut est `missing`.

> **Important**
>
> Les alarmes configurées sur les métriques Amazon EC2 peuvent passer temporairement à l'état INSUFICIENT_DATA s'il manque des points de données métriques. Cela est rare, mais cela peut se produire lorsque le reporting des métriques est interrompu, même lorsque l'instance Amazon EC2 est saine. Pour les alarmes relatives aux métriques Amazon EC2 configurées pour effectuer des actions d'arrêt, d'arrêt, de redémarrage ou de restauration, nous vous recommandons de configurer ces alarmes de manière à traiter les données manquantes comme `missing` telles et à ce que ces alarmes ne se déclenchent que lorsqu'elles sont en état ALARM.

Choisir la meilleure option pour votre alerte permet d'éviter des changements de condition d'alerte superflus et trompeurs, mais également de fournir une indication plus précise de l'état du système.

> **Important**
>
> Les alertes qui évaluent les métriques dans l'espace de noms `AWS/DynamoDB` ignorent toujours les données manquantes, même si vous choisissez une option différente pour le traitement des données manquantes par l'alerte. Lorsqu'une métrique `AWS/DynamoDB` contient des données manquantes, les alertes qui évaluent cette métrique restent dans leur état actuel.

Évaluation de l'état de l'alerte lorsqu'il manque des données

Chaque fois qu'une alarme évalue s'il faut changer d'état, CloudWatch tente de récupérer un nombre de points de données supérieur au nombre spécifié comme périodes d'évaluation. Le nombre de points de données exact qu'il tente de récupérer dépend de la longueur de la période d'alerte et du fait qu'elle est ou non basée sur une métrique avec une résolution standard ou une haute résolution. La période des points de données qu'il tente de récupérer est la plage d'évaluation.

Une fois ces CloudWatch points de données récupérés, voici ce qui se passe :

- S'il ne manque aucun point de données dans la plage d'évaluation, CloudWatch évalue l'alarme en fonction des points de données les plus récents collectés. Le nombre de points de données évalués est égal aux Evaluation Periods (Périodes d'évaluation) pour l'alerte. Les points de données supplémentaires situés plus loin dans la plage d'évaluation ne sont pas nécessaires et sont ignorés.
- Si certains points de données de la plage d'évaluation sont manquants, mais que le nombre total de points de données existants qui ont été extraits avec succès de la plage d'évaluation est égal ou supérieur aux périodes d'évaluation de l'alarme, CloudWatch évalue l'état de l'alarme en fonction des points de données réels les plus récents qui ont été récupérés avec succès, y compris les points de données supplémentaires nécessaires situés plus loin dans la plage d'évaluation. Dans ce cas, la valeur que vous avez définie pour traiter les données manquantes n'est pas nécessaire et est ignorée.
- Si certains points de données de la plage d'évaluation sont manquants et que le nombre de points de données réels récupérés est inférieur au nombre de périodes d'évaluation de l'alarme, complétez CloudWatch les points de données manquants avec le résultat que vous avez spécifié sur la manière de traiter les données manquantes, puis évalue l'alarme. Cependant, tous les points de données réels de la plage d'évaluation sont inclus dans l'évaluation. CloudWatch n'utilise les points de données manquants que le moins de fois possible.

> **Note**
>
> Un cas particulier de ce comportement est que les CloudWatch alarmes peuvent réévaluer à plusieurs reprises le dernier ensemble de points de données pendant un certain temps après l'arrêt du flux de la métrique. Cette réévaluation peut entraîner l'alerte à changer d'état et à réexécuter des actions, si le changement d'état est survenu immédiatement avant que le flux

de la métrique ne soit interrompu. Pour atténuer ce comportement, utilisez des périodes plus courtes.

Les tableaux suivants illustrent des exemples du comportement d'évaluation de l'alerte. Dans le premier tableau, les points de données relatifs aux périodes d'alarme et d'évaluation sont tous deux égaux à 3. CloudWatch récupère les 5 points de données les plus récents lors de l'évaluation de l'alarme, au cas où certains des 3 points de données les plus récents seraient manquants. 5 est la plage d'évaluation de l'alarme.

La colonne 1 montre les 5 points de données les plus récents, car la plage d'évaluation est 5. Ces points de données sont affichés avec le point de données le plus récent sur la droite. 0 est un point de données en-deçà du seuil, X est un point de données au-delà du seuil et - est un point de données manquant.

La deuxième colonne indique combien des 3 points de données nécessaires sont absents. Même si les 5 points de données les plus récents sont évalués, seuls 3 d'entre eux (le paramètre pour les Périodes d'évaluation) sont nécessaires pour évaluer l'état de l'alerte. Le nombre de points de données dans la deuxième colonne est le nombre de points de données qui doivent être « renseignés » à l'aide du paramètre pour la façon dont les données manquantes sont traitées.

Dans les colonnes 3 à 6, les en-têtes de colonne sont les valeurs possibles pour la façon de traiter les données manquantes. Les lignes de ces colonnes indiquent l'état d'alerte défini pour chacune de ces méthodes possibles de traitement des données manquantes.

Points de données	Nombre de points de données qui doivent être remplis	MANQUANT	IGNORER	AU-DELÀ DU SEUIL	EN-DEÇÀ DU SEUIL
0 - X - X	0	OK	OK	OK	OK
- - - - 0	2	OK	OK	OK	OK
- - - - -	3	INSUFFICIENT_DATA	Conserver l'état actuel	ALARM	OK
0 X X - X	0	ALARM	ALARM	ALARM	ALARM

Points de données	Nombre de points de données qui doivent être remplis	MANQUANT	IGNORER	AU-DELÀ DU SEUIL	EN-DEÇÀ DU SEUIL
- - X - -	2	ALARM	Conserver l'état actuel	ALARM	OK

Dans la deuxième ligne du tableau précédent, l'alerte reste OK, même si les données manquantes sont traitées comme au-delà du seuil, car le seul point de données existant n'est pas au-delà du seuil. Cette valeur est évaluée avec deux points de données manquants qui sont traités comme au-delà du seuil. Lors de l'évaluation suivante de cette alerte, si les données sont toujours manquantes, l'état deviendra ALARM, étant donné que ce point de données en-deçà du seuil ne fera plus parti de la plage d'évaluation.

La troisième ligne, où les cinq points de données les plus récents sont manquants, illustre comment les différents paramètres de traitement des données manquantes affectent l'état d'alerte. Si les points de données manquants sont considérés comme une violation, l'alerte passe en état alerte, tandis que si elles sont considérées comme ne pas entrer en violation, l'alerte passe en état OK. Si les points de données manquants sont ignorés, l'alerte conserve l'état actuel qu'elle avait avant les points de données manquants. Et si les points de données manquants sont simplement considérés comme manquants, alors l'alerte n'a pas assez de données réelles récentes pour faire une évaluation, et passe dans INSUFFICIENT_DATA.

Dans la quatrième rangée, l'alerte passe à l'état ALARM dans tous les cas, car les trois points de données les plus récents sont en violation, et les Périodes d'évaluation ainsi que les Points de données à l'alerte de l'alerte sont tous deux réglés sur 3. Dans ce cas, le point de données manquant est ignoré et le paramètre relatif à l'évaluation des données manquantes n'est pas requis, car il y a 3 points de données réels à évaluer.

La ligne 5 représente un cas spécial d'évaluation d'alerte appelé état d'alerte prématurée. Pour plus d'informations, consultez Éviter les transitions prématurées vers l'état d'alerte.

Dans le tableau suivant, la valeur de Période est à nouveau définie sur 5 minutes et celle de Points de données avant l'alerte est seulement 2 alors que celle de Périodes d'évaluation est de 3. Il s'agit d'une alerte 2 sur 3, M sur N.

La plage d'évaluation est de 5. Il s'agit du nombre maximal de points de données récents qui sont récupérés et peuvent être utilisés au cas où certains points de données seraient manquants.

Points de données	Nbre de points de données manquants	MANQUANT	IGNORER	AU-DELÀ DU SEUIL	EN-DEÇÀ DU SEUIL
0 - X - X	0	ALARM	ALARM	ALARM	ALARM
0 0 X 0 X	0	ALARM	ALARM	ALARM	ALARM
0 - X - -	1	OK	OK	ALARM	OK
- - - - 0	2	OK	OK	ALARM	OK
- - - - X	2	ALARM	Conserver l'état actuel	ALARM	OK

Dans les lignes 1 et 2, l'alerte passe toujours à l'état ALARM, car 2 des 3 points de données les plus récents sont en train de franchir. Dans la ligne 2, les deux points de données les plus anciens de la plage d'évaluation ne sont pas nécessaires, car aucun des 3 points de données les plus récents n'est manquant, de sorte que ces deux points de données plus anciens sont ignorés.

Dans les lignes 3 et 4, l'alerte passe à l'état ALARM uniquement si les données manquantes sont traitées comme des violations, auquel cas les deux points de données manquants les plus récents sont tous deux traités comme des violations. Dans la ligne 4, ces deux points de données manquants qui sont traités comme étant au-delà du seuil fournissent les deux points de données au-delà du seuil pour déclencher l'état ALARM.

La ligne 5 représente un cas spécial d'évaluation d'alerte appelé état d'alerte prématurée. Pour plus d'informations, consultez la section suivante.

Éviter les transitions prématurées vers l'état d'alerte

CloudWatch l'évaluation des alarmes inclut une logique visant à éviter les fausses alarmes, lorsque l'alarme passe prématurément en état d'alarme lorsque les données sont intermittentes. L'exemple illustré à la ligne 5 des tableaux de la section précédente illustre cette logique. Dans ces lignes, et

dans les exemples suivants, la propriété Evaluation Periods (Périodes d'évaluation)est 3 et la plage d'évaluation est de 5 points de données. Datapoints to Alarm (Points de données à l'alerte) est défini sur 3, sauf pour l'exemple M sur N, où Datapoints to Alarm (Points de données à l'alerte) est défini sur 2.

Supposons que les données les plus récentes d'une alerte soient - - - - X, avec quatre points de données manquants, puis un point de données de violation comme point de données le plus récent. Étant donné que le point de données suivant peut être sans violation, l'alerte ne passe pas immédiatement dans l'état ALARM lorsque les données sont - - - - X ou - - - X - et Datapoints to Alarm (Points de données à l'alerte) est défini sur 3. De cette façon, les faux positifs sont évités lorsque le point de données suivant n'est pas en violation et que les données sont - - - X 0 ou - - X - 0.

Toutefois, si les derniers points de données sont - - X - -, l'alerte passe en état alerte même si les points de données manquants sont considérés comme manquants. En effet, les alertes sont conçues pour toujours passer à l'état ALARM lorsque le plus ancien point de données de violation disponible pendant les Evaluation Periods (Périodes d'évaluation) est au moins aussi ancien que la valeur des Datapoint to Alarm (Points de données à alerter), et que tous les autres points de données plus récents sont en violation ou manquants. Dans ce cas, l'alerte passe en état ALARM même si le nombre total de points de données disponibles est inférieur à M (Datapoints to Alarm (Points de données à l'alerte)).

Cette logique d'alerte s'applique également aux alertes M sur N. Si le point de données de violation le plus ancien au cours de la plage d'évaluation est au moins aussi ancien que la valeur de Datapoints to Alarm (Points de données à l'alerte), et que tous les points de données les plus récents sont soit en violation ou manquants, l'alerte passe en état ALARM quelle que soit la valeur de M (Datapoints to Alarm (Points de données à l'alerte)).

alertes haute résolution

Si vous définissez une alerte sur une métrique haute résolution, vous pouvez spécifier une alerte haute résolution avec une période de 10 secondes ou de 30 secondes ou vous pouvez définir une alerte régulière avec une période correspondant à n'importe quel multiple de 60 secondes. Les frais engendrés par des alertes haute résolution sont plus élevés. Pour plus d'informations sur les métriques haute résolution, consultez Publier des métriques personnalisées .

alertes sur les expressions mathématiques

Vous pouvez définir une alarme en fonction du résultat d'une expression mathématique basée sur une ou plusieurs CloudWatch mesures. Une expression mathématique utilisée pour une alerte peut inclure jusqu'à 10 métriques. Chaque métrique doit utiliser la même période.

Pour une alarme basée sur une expression mathématique, vous pouvez spécifier la manière dont vous souhaitez CloudWatch traiter les points de données manquants. Dans ce cas, un point de données est considéré comme manquant si l'expression mathématique ne renvoie aucune valeur pour ce point de données.

Les alertes basées sur des expressions mathématiques ne peuvent pas effectuer des actions Amazon EC2.

Pour en savoir plus sur les expressions mathématiques et la syntaxe de métrique, consultez Utilisation des mathématiques appliquées aux métriques.

CloudWatch Alarmes basées sur les percentiles et échantillons de données faibles

Lorsque vous définissez un centile comme statistique d'une alerte, vous pouvez préciser quelle action réaliser lorsque les données sont insuffisantes pour obtenir une estimation statistique de qualité. Vous pouvez décider que l'alerte doit évaluer la statistique quoi qu'il arrive et éventuellement qu'elle change d'état. Vous pouvez également décider que l'alerte doit ignorer la métrique si la taille de l'échantillon est réduite et attendre pour l'évaluer que les données soient en quantité suffisante pour être significatives statistiquement.

Pour les centiles entre 0,5 (inclusif) et 1,00 (exclusif), ce paramètre est utilisé lorsque moins de 10/(1-centile) points de données sont présents lors de la période d'évaluation. Par exemple, ce paramètre serait utilisé si moins de 1 000 échantillons étaient présents pour une alerte dans un centile p99. Pour les centiles entre 0 et 0,5 (exclusif), ce paramètre est utilisé lorsque moins de 10/centile points de données sont présents.

Caractéristiques communes des CloudWatch alarmes

Les fonctionnalités suivantes s'appliquent à toutes les CloudWatch alarmes :

- Il n'existe pas de limite au nombre d'alertes que vous pouvez créer. Pour créer ou mettre à jour une alarme, vous utilisez la CloudWatch console, l'action PutMetricAlarmAPI ou la put-metric-alarmcommande du AWS CLI.
- Les noms des alertes ne doivent contenir que des caractères UTF-8 et ne peuvent pas contenir de caractères de contrôle ASCII.
- Vous pouvez répertorier une ou toutes les alarmes actuellement configurées, ainsi que toutes les alarmes dans un état particulier à l'aide de la CloudWatch console, de l'action DescribeAlarmsAPI ou de la commande describe-alarm dans le. AWS CLI
- Vous pouvez désactiver et activer les alarmes à l'aide des actions DisableAlarmActionset de l'EnableAlarmActionsAPI ou disable-alarm-actionsdes enable-alarm-actionscommandes et du AWS CLI.
- Vous pouvez tester une alarme en la réglant sur n'importe quel état à l'aide de l'action SetAlarmStateAPI ou de la set-alarm-statecommande du AWS CLI. Ce changement de statut temporaire ne dure que jusqu'à la comparaison d'alerte suivante.
- Vous pouvez créer une alerte pour une métrique personnalisée avant de créer cette dernière. Pour que l'alerte soit valide, vous devez inclure toutes les dimensions pour la métrique personnalisée en plus de l'espace de nom et du nom de la métrique dans la définition de l'alerte. Pour ce faire, vous pouvez utiliser l'action PutMetricAlarmAPI ou la put-metric-alarmcommande du AWS CLI.
- Vous pouvez consulter l'historique d'une alarme à l'aide de la CloudWatch console, de l'action de l'DescribeAlarmHistoryAPI ou de la describe-alarm-historycommande du AWS CLI. CloudWatch conserve l'historique des alarmes pendant 30 jours. Chaque transition de statut est marquée par un horodatage unique. Dans de rares cas, votre historique peut afficher plus d'une notification pour un changement de statut. L'horodatage vous permet de confirmer les modifications de statut uniques.
- Vous pouvez ajouter des alarmes à vos favoris à l'aide de l'option Favoris et récents du volet de navigation de la CloudWatch console en passant le curseur sur l'alarme que vous souhaitez ajouter aux favoris et en choisissant le symbole en forme d'étoile à côté de celle-ci.
- Le nombre de périodes d'évaluation pour une alerte multiplié par la longueur de chaque période d'évaluation ne peut pas dépasser un jour.

> **Note**
>
> Certaines AWS ressources n'envoient pas de données métriques CloudWatch sous certaines conditions.
> Par exemple, il se peut qu'Amazon EBS n'envoie pas de données de métriques pour un volume disponible qui n'est pas lié à une instance Amazon EC2, car aucune activité de

métrique n'est à surveiller pour ce volume. Si vous avez une alerte définie pour ce type de métrique, vous pouvez remarquer que son état passe à `INSUFFICIENT_DATA`. Cela peut indiquer que votre ressource est inactive et ne signifie pas nécessairement la présence d'un problème. Vous pouvez spécifier la façon dont chaque alerte traite les données manquantes. Pour plus d'informations, consultez Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.

Recommandations relatives aux meilleures pratiques en matière d'alarme pour les AWS services

CloudWatch fournit des recommandations relatives aux alarmes de out-of-the boîte. Il s'agit d' CloudWatch alarmes que nous vous recommandons de créer pour les métriques publiées par d'autres AWS services. Ces recommandations peuvent vous aider à identifier les métriques pour lesquelles vous devez définir des alarmes afin de suivre les bonnes pratiques de surveillance. Les recommandations suggèrent également les seuils d'alarme à définir. Le respect de ces recommandations peut vous aider à ne pas manquer une importante surveillance de votre AWS infrastructure.

Pour trouver les recommandations d'alarme, utilisez la section des métriques de la CloudWatch console et sélectionnez le filtre des recommandations d'alarme. Si vous accédez aux alarmes recommandées dans la console, puis que vous créez une alarme recommandée, CloudWatch vous pouvez préremplir certains paramètres d'alarme. Pour certaines alarmes recommandées, la valeur du seuil d'alarme est également prédéfinie. Vous pouvez également utiliser la console pour télécharger les définitions infrastructure-as-code d'alarme pour les alarmes recommandées, puis utiliser ce code pour créer l'alarme dans AWS CloudFormation AWS CLI, le ou Terraform.

Vous pouvez également consulter la liste des alarmes recommandées dans Alarmes recommandées.

Les alarmes que vous créez vous sont facturées au même taux que toutes les autres alarmes que vous créez dans CloudWatch. L'utilisation des recommandations n'entraîne aucuns frais supplémentaires. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

Recherche et création d'alarmes recommandées

Procédez comme suit pour trouver les mesures qui vous CloudWatch recommandent de définir des alarmes et, éventuellement, de créer l'une de ces alarmes. La première procédure explique comment

rechercher les métriques contenant des alarmes recommandées et comment créer l'une de ces alarmes.

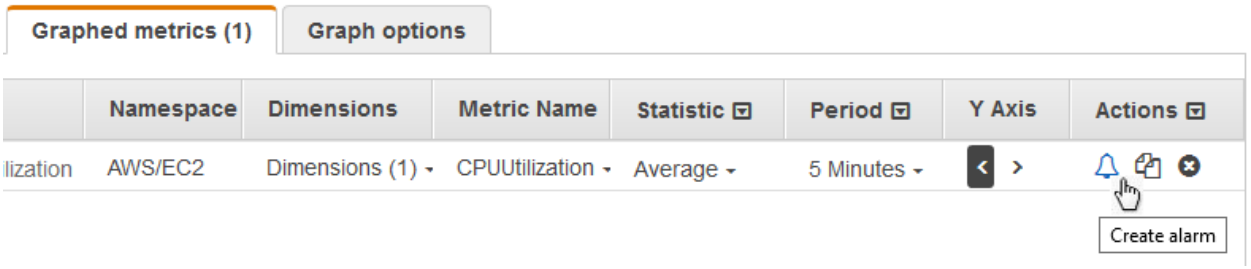
Vous pouvez également télécharger en bloc les définitions d' infrastructure-as-code alarme pour toutes les alarmes recommandées dans un espace de AWS noms, telles que `AWS/Lambda` ou`AWS/S3`. Vous trouverez ces instructions plus avant dans cette rubrique.

Pour rechercher les métriques contenant les alarmes recommandées et créer une seule alarme recommandée

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Au-dessus du tableau des métriques, sélectionnez Recommandations d'alarmes.

   La liste des espaces de noms de métriques est filtrée pour n'inclure que les métriques contenant des recommandations d'alarme et publiées par les services de votre compte.
4. Choisissez l'espace de noms d'un service.

   La liste des métriques de cet espace de noms est filtrée pour n'inclure que celles qui contiennent des recommandations d'alarme.
5. Pour connaître l'intention de l'alarme et le seuil recommandé pour une métrique, choisissez Afficher les détails.
6. Pour créer une alarme pour l'une des métriques, effectuez l'une des opérations suivantes :
   - Pour utiliser la console afin de créer l'alarme, procédez comme suit :
     a. Cochez la case correspondant à la métrique, puis cliquez sur l'onglet Graphiques des métriques.
     b. Choisissez l'icône d'alarme.



   L'assistant de création d'alarme apparaît, avec le nom de la métrique, les statistiques et la période renseignés en fonction de la recommandation d'alarme. Si la

recommandation inclut une valeur de seuil spécifique, cette valeur est également prédéfinie.

c. Choisissez Suivant.

d. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alarme passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

Pour que l'alerte envoie plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, choisissez Add notification (Ajouter une notification).

Pour que l'alerte n'envoie pas de notifications, choisissez Remove (Supprimer).

e. Pour que l'alerte exécute Auto Scaling ou des actions EC2, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer.

f. Lorsque vous avez terminé, choisissez Next (Suivant).

g. Saisissez un nom et une description pour l'alerte. Le nom ne doit contenir que des caractères ASCII. Sélectionnez ensuite Next (Suivant).

h. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

- Pour télécharger une définition infrastructure-as-code d'alarme à utiliser dans l'un ou l'autre AWS CloudFormation AWS CLI, ou dans Terraform, choisissez Télécharger le code d'alarme et sélectionnez le format souhaité. Le code téléchargé comportera les paramètres recommandés pour le nom de la métrique, les statistiques et le seuil.

Pour télécharger les définitions des infrastructure-as-code alarmes pour toutes les alarmes recommandées pour un AWS service

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Au-dessus du tableau des métriques, sélectionnez Recommandations d'alarmes.

   La liste des espaces de noms de métriques est filtrée pour n'inclure que les métriques contenant des recommandations d'alarme et publiées par les services de votre compte.
4. Choisissez l'espace de noms d'un service.

   La liste des métriques de cet espace de noms est filtrée pour n'inclure que celles qui contiennent des recommandations d'alarme.

5. Le code d'alarme Download indique le nombre d'alarmes recommandées pour les métriques de cet espace de noms. Pour télécharger les définitions infrastructure-as-code d'alarme pour toutes les alarmes recommandées, choisissez Télécharger le code d'alarme, puis choisissez le format de code souhaité.

Alarmes recommandées

Les sections suivantes répertorient les métriques pour lesquelles nous vous recommandons de définir des alarmes conformes aux bonnes pratiques. Pour chaque métrique, les dimensions, l'intention de l'alarme, le seuil recommandé, la justification du seuil, ainsi que la durée de la période et le nombre de points de données sont également affichés.

Certaines métriques peuvent apparaître deux fois dans la liste. Cela se produit lorsque différentes alarmes sont recommandées pour différentes combinaisons de dimensions de cette métrique.

Les points de données pour le déclenchement d'alarme sont le nombre de points de données qui doivent être violés pour que l'alarme passe en état ALARM. Les périodes d'évaluation sont le nombre de périodes prises en compte lors de l'évaluation de l'alarme. Si ces chiffres sont identiques, l'alarme ne passe en état ALARM que lorsque les valeurs de ce nombre de périodes consécutives dépassent le seuil. Si les points de données pour le déclenchement d'alarme sont inférieurs aux périodes d'évaluation, il s'agit d'une alarme « M sur N » et l'alarme passe à l'état ALARM si dès que le nombre de points de données pour le déclenchement d'alarme est dépassé au cours d'un ensemble de points de données des périodes d'évaluation. Pour plus d’informations, consultez Évaluation d'une alerte.

Rubriques

- Amazon API Gateway
- Amazon EC2 Auto Scaling
- Amazon CloudFront
- Amazon Cognito
- Amazon DynamoDB
- Amazon EBS
- Amazon EC2
- Amazon ElastiCache
- Amazon EC2 (AWS/ElasticGPUs)
- Amazon ECS

- Amazon ECS avec Container Insights
- Amazon EFS
- Amazon EKS avec Container Insights
- Amazon Kinesis Data Streams
- Lambda
- Aperçu Lambda
- Amazon VPC (AWS/NATGateway)
- AWS Lien privé (AWS/PrivateLinkEndpoints)
- AWS Lien privé (AWS/PrivateLinkServices)
- Amazon RDS
- Amazon Route 53 Public Data Plane
- Amazon S3
- S3ObjectLambda
- Amazon SNS
- Amazon SQS
- AWS VPN

Amazon API Gateway

4XXError

Dimensions :ApiName, scène

Description de l'alarme : cette alarme détecte un taux élevé d'erreurs côté client. Cela peut indiquer un problème dans les paramètres d'autorisation ou de requête client. Cela peut également signifier qu'une ressource a été supprimée ou qu'un client en demande une qui n'existe pas. Envisagez d'activer CloudWatch les journaux et de vérifier l'absence d'erreurs susceptibles d'être à l'origine des erreurs 4XX. En outre, pensez à activer CloudWatch les métriques détaillées pour afficher cette métrique par ressource et par méthode et affiner la source des erreurs. Des erreurs peuvent également être causées par le dépassement de la limite configurée. Si les réponses et les journaux signalent des taux élevés et inattendus d'erreurs 429, suivez ce guide pour résoudre ce problème.

Intention : cette alarme peut détecter des taux élevés d'erreurs côté client pour les requêtes API Gateway.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : le seuil suggéré détecte les cas où plus de 5 % du total des requêtes reçoivent des erreurs 4xx. Cependant, vous pouvez ajuster le seuil en fonction du trafic des requêtes et des taux d'erreur acceptables. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs 4xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

5XXError

Dimensions :ApiName, scène

Description de l'alarme : cette alarme permet de détecter un taux élevé d'erreurs côté serveur. Cela peut indiquer qu'il y a un problème sur le backend de l'API, le réseau ou l'intégration entre la passerelle d'API et l'API du backend. Cette documentation peut vous aider à résoudre la cause des erreurs 5xx.

Intention : cette alarme peut détecter des taux élevés d'erreurs côté serveur pour les requêtes API Gateway.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : le seuil suggéré détecte les cas où plus de 5 % du total des requêtes reçoivent des erreurs 5xx. Vous pouvez toutefois ajuster le seuil en fonction du trafic des requêtes et des taux d'erreur acceptables. Vous pouvez également analyser les données historiques pour déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs 5xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Count

Dimensions :ApiName, scène

Description de l'alarme : cette alarme permet de détecter un faible volume de trafic pour l'étape de l'API REST. Cela peut indiquer un problème lié à l'appel de l'API par l'application, par exemple lors de l'utilisation de points de terminaison incorrects. Cela peut également indiquer un problème lié à la configuration ou aux autorisations de l'API qui la rend inaccessible aux clients.

Intention : cette alarme peut détecter un volume de trafic étonnamment faible pour l'étape de l'API REST. Nous vous recommandons de créer cette alarme si votre API reçoit un nombre prévisible et régulier de requêtes dans des conditions normales. Si CloudWatch les métriques détaillées sont activées et que vous pouvez prévoir le volume de trafic normal par méthode et par ressource, nous vous recommandons de créer des alarmes alternatives afin de surveiller de manière plus précise les baisses de volume de trafic pour chaque ressource et méthode. Cette alarme n'est pas recommandée pour les API qui ne s'attendent pas à un trafic constant et régulier.

Statistique : SampleCount

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction de l'analyse des données historiques afin de déterminer le nombre de requêtes de référence attendu pour votre API. Le réglage du seuil à une valeur très élevée peut rendre l'alarme trop sensible pendant les périodes de faible trafic normal et attendu. À l'inverse, si elle est réglée sur une valeur très faible, l'alarme risque de ne pas détecter de petites baisses anormales du volume de trafic.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

Count

Dimensions : scèneApiName, ressource, méthode

Description de l'alarme : cette alarme permet de détecter un faible volume de trafic pour la ressource et la méthode de l'API REST au cours de l'étape. Cela peut indiquer un problème lié à l'appel de l'API par l'application, par exemple lors de l'utilisation de points de terminaison incorrects. Cela peut également indiquer un problème lié à la configuration ou aux autorisations de l'API qui la rend inaccessible aux clients.

Intention : cette alarme peut détecter un volume de trafic étonnamment faible pour la ressource et la méthode de l'API REST au cours de l'étape. Nous vous recommandons de créer cette alarme si votre API reçoit un nombre prévisible et régulier de requêtes dans des conditions normales. Cette alarme n'est pas recommandée pour les API qui ne s'attendent pas à un trafic constant et régulier.

Statistique : SampleCount

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction de l'analyse des données historiques afin de déterminer le nombre de requêtes de référence attendu pour votre API. Le réglage du seuil à une valeur très élevée peut rendre l'alarme trop sensible pendant les périodes de faible trafic normal et attendu. À l'inverse, si elle est réglée sur une valeur très faible, l'alarme risque de ne pas détecter de petites baisses anormales du volume de trafic.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

Count

Dimensions :ApiId, scène

Description de l'alarme : cette alarme permet de détecter un faible volume de trafic pour l'étape de l'API HTTP. Cela peut indiquer un problème lié à l'appel de l'API par l'application, par exemple lors de l'utilisation de points de terminaison incorrects. Cela peut également indiquer un problème lié à la configuration ou aux autorisations de l'API qui la rend inaccessible aux clients.

Intention : cette alarme peut détecter un volume de trafic étonnamment faible pour l'étape de l'API HTTP. Nous vous recommandons de créer cette alarme si votre API reçoit un nombre prévisible et régulier de requêtes dans des conditions normales. Si vous avez activé CloudWatch les métriques détaillées et que vous pouvez prévoir le volume de trafic normal par itinéraire, nous

vous recommandons de créer des alarmes alternatives afin de surveiller de manière plus précise les baisses de volume de trafic pour chaque itinéraire. Cette alarme n'est pas recommandée pour les API qui ne s'attendent pas à un trafic constant et régulier.

Statistique : SampleCount

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez la valeur du seuil en fonction de l'analyse des données historiques afin de déterminer le nombre de requêtes de référence attendu pour votre API. Le réglage du seuil à une valeur très élevée peut rendre l'alarme trop sensible pendant les périodes de faible trafic normal et attendu. À l'inverse, si elle est réglée sur une valeur très faible, l'alarme risque de ne pas détecter de petites baisses anormales du volume de trafic.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

Count

Dimensions : scèneApiId, ressource, méthode

Description de l'alarme : cette alarme permet de détecter un faible volume de trafic pour la route de l'API HTTP au cours de l'étape. Cela peut indiquer un problème lié à l'appel de l'API par l'application, par exemple lors de l'utilisation de points de terminaison incorrects. Cela peut également indiquer un problème lié à la configuration ou aux autorisations de l'API qui la rend inaccessible aux clients.

Intention : cette alarme peut détecter un volume de trafic étonnamment faible pour la route de l'API HTTP au cours de l'étape. Nous vous recommandons de créer cette alarme si votre API reçoit un nombre prévisible et régulier de requêtes dans des conditions normales. Cette alarme n'est pas recommandée pour les API qui ne s'attendent pas à un trafic constant et régulier.

Statistique : SampleCount

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez la valeur du seuil en fonction de l'analyse des données historiques afin de déterminer le nombre de requêtes de référence attendu pour votre API. Le

réglage du seuil à une valeur très élevée peut rendre l'alarme trop sensible pendant les périodes de faible trafic normal et attendu. À l'inverse, si elle est réglée sur une valeur très faible, l'alarme risque de ne pas détecter de petites baisses anormales du volume de trafic.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

IntegrationLatency

Dimensions :ApiId, scène

Description de l'alarme : cette alarme permet de détecter s'il existe une latence d'intégration élevée pour les demandes d'API d'une étape. Vous pouvez corréler la valeur de la métrique `IntegrationLatency` avec la métrique de latence correspondante de votre backend, telle que la métrique `Duration` pour les intégrations Lambda. Cela vous permet de déterminer si le backend de l'API met plus de temps à traiter les demandes des clients en raison de problèmes de performances, ou s'il existe une autre surcharge liée à l'initialisation ou au démarrage à froid. En outre, pensez à activer CloudWatch les journaux pour votre API et à vérifier les journaux pour détecter toute erreur susceptible d'être à l'origine des problèmes de latence élevée. En outre, pensez à activer CloudWatch les métriques détaillées pour avoir une vue de cette métrique par itinéraire, afin de vous aider à affiner la source de la latence d'intégration.

Intention : cette alarme peut détecter les cas où les requêtes API Gateway d'une étape présentent une latence d'intégration élevée. Nous recommandons cette alarme pour les WebSocket API, et nous la considérons comme facultative pour les API HTTP, car elles contiennent déjà des recommandations d'alarme distinctes pour la métrique de latence. Si CloudWatch les métriques détaillées sont activées et que vos exigences en matière de performances de latence d'intégration diffèrent par itinéraire, nous vous recommandons de créer des alarmes alternatives afin d'avoir une surveillance plus fine de la latence d'intégration pour chaque itinéraire.

Statistique : p90

Seuil recommandé : 2 000.0

Justification du seuil : la valeur de seuil suggérée ne fonctionne pas pour toutes les charges de travail de l'API. Toutefois, vous pouvez l'utiliser comme point de départ pour le seuil. Vous pouvez

ensuite choisir différentes valeurs de seuil en fonction de la charge de travail et des exigences de latence, de performances et de SLA acceptables pour l'API. S'il est acceptable que l'API ait une latence plus élevée en général, définissez une valeur de seuil plus élevée pour rendre l'alarme moins sensible. Toutefois, si l'API est censée fournir des réponses en temps quasi réel, définissez une valeur de seuil inférieure. Vous pouvez également analyser les données historiques afin de déterminer la latence de base attendue pour la charge de travail de l'application, puis régler la valeur du seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

IntegrationLatency

Dimensions : étapeApiId, itinéraire

Description de l'alarme : Cette alarme permet de détecter s'il existe une latence d'intégration élevée pour les demandes d' WebSocket API relatives à un itinéraire dans une étape. Vous pouvez corréler la valeur de la métrique `IntegrationLatency` avec la métrique de latence correspondante de votre backend, telle que la métrique `Duration` pour les intégrations Lambda. Cela vous permet de déterminer si le backend de l'API met plus de temps à traiter les demandes des clients en raison de problèmes de performances ou s'il existe une autre surcharge liée à l'initialisation ou au démarrage à froid. En outre, pensez à activer CloudWatch les journaux pour votre API et à vérifier les journaux pour détecter toute erreur susceptible d'être à l'origine des problèmes de latence élevée.

Intention : cette alarme peut détecter les cas où les demandes d'API Gateway pour une route dans une étape présentent une latence d'intégration élevée.

Statistique : p90

Seuil recommandé : 2 000.0

Justification du seuil : la valeur de seuil suggérée ne fonctionne pas pour toutes les charges de travail de l'API. Toutefois, vous pouvez l'utiliser comme point de départ pour le seuil. Vous pouvez ensuite choisir différentes valeurs de seuil en fonction de la charge de travail et des exigences de latence, de performances et de SLA acceptables pour l'API. S'il est acceptable

que l'API ait une latence plus élevée en général, vous pouvez définir une valeur de seuil plus élevée pour rendre l'alarme moins sensible. Toutefois, si l'API est censée fournir des réponses en temps quasi réel, définissez une valeur de seuil inférieure. Vous pouvez également analyser les données historiques afin de déterminer la latence de base attendue pour la charge de travail de l'application, puis régler la valeur du seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latence

Dimensions :ApiName, scène

Description de l'alarme : cette alarme détecte une latence élevée dans une étape. Recherchez la valeur de la métrique `IntegrationLatency` pour vérifier la latence du backend de l'API. Si les deux métriques sont globalement alignées, le backend de l'API est à l'origine d'une latence plus élevée et vous devriez examiner les problèmes à ce niveau. Envisagez également d'activer CloudWatch les journaux et de vérifier les erreurs susceptibles d'être à l'origine de cette latence élevée. En outre, pensez à activer CloudWatch les métriques détaillées pour afficher cette métrique par ressource et par méthode et affiner la source de la latence. Le cas échéant, reportez-vous aux guides résoudre les problèmes avec Lambda de résolution des problèmes pour résoudre les problèmes liés à mon point de terminaison d'API optimisé pour la périphérie.

Intention : cette alarme peut détecter les cas de latence élevée des requêtes API Gateway d'une étape. Si CloudWatch les métriques détaillées sont activées et que vos exigences en matière de performances de latence diffèrent pour chaque méthode et ressource, nous vous recommandons de créer des alarmes alternatives afin de surveiller plus précisément la latence pour chaque ressource et méthode.

Statistique : p90

Seuil recommandé : 2 500,0

Justification du seuil : la valeur de seuil suggérée ne fonctionne pas pour toutes les charges de travail d'API. Toutefois, vous pouvez l'utiliser comme point de départ pour le seuil. Vous pouvez ensuite choisir différentes valeurs de seuil en fonction de la charge de travail et des

exigences de latence, de performances et de SLA acceptables pour l'API. S'il est acceptable que l'API ait une latence plus élevée en général, vous pouvez définir une valeur de seuil plus élevée pour rendre l'alarme moins sensible. Toutefois, si l'API est censée fournir des réponses en temps quasi réel, définissez une valeur de seuil inférieure. Vous pouvez également analyser les données historiques pour déterminer la latence de référence attendue pour la charge de travail de l'application, puis ajuster la valeur du seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latence

Dimensions : scèneApiName, ressource, méthode

Description de l'alarme : cette alarme détecte une latence élevée pour une ressource et une méthode dans une étape. Recherchez la valeur de la métrique `IntegrationLatency` pour vérifier la latence du backend de l'API. Si les deux métriques sont globalement alignées, le backend de l'API est à l'origine d'une latence plus élevée et vous devriez examiner les problèmes de performances à ce niveau. Envisagez également d'activer CloudWatch les journaux et de vérifier les erreurs susceptibles d'être à l'origine de cette latence élevée. Vous pouvez également les guides résoudre les problèmes avec Lambda de résolution des problèmes pour résoudre les problèmes liés à mon point de terminaison d'API optimisé pour la périphérie, le cas échéant.

Intention : cette alarme peut détecter les cas où les demandes d'API Gateway pour une ressource et une méthode dans une étape présentent une latence élevée.

Statistique : p90

Seuil recommandé : 2 500,0

Justification du seuil : la valeur de seuil suggérée ne fonctionne pas pour toutes les charges de travail de l'API. Toutefois, vous pouvez l'utiliser comme point de départ pour le seuil. Vous pouvez ensuite choisir différentes valeurs de seuil en fonction de la charge de travail et des exigences de latence, de performances et de SLA acceptables pour l'API. S'il est acceptable que l'API ait une latence plus élevée en général, vous pouvez définir une valeur de seuil plus élevée pour rendre l'alarme moins sensible. Toutefois, si l'API est censée fournir des réponses en temps quasi

réel, définissez une valeur de seuil inférieure. Vous pouvez également analyser les données historiques afin de déterminer la latence de référence attendue pour la charge de travail de l'application, puis ajuster la valeur du seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latence

Dimensions :ApiId, scène

Description de l'alarme : cette alarme détecte une latence élevée dans une étape. Recherchez la valeur de la métrique `IntegrationLatency` pour vérifier la latence du backend de l'API. Si les deux métriques sont globalement alignées, le backend de l'API est à l'origine d'une latence plus élevée et vous devriez examiner les problèmes de performances à ce niveau. Envisagez également d'activer CloudWatch les journaux et de vérifier les erreurs susceptibles d'être à l'origine de cette latence élevée. En outre, pensez à activer CloudWatch les métriques détaillées pour afficher cette métrique par itinéraire et affiner la source de la latence. Vous pouvez également consulter le guide de résolution des problèmes d'intégration à Lambda, le cas échéant.

Intention : cette alarme peut détecter les cas de latence élevée des requêtes API Gateway d'une étape. Si CloudWatch les métriques détaillées sont activées et que vos exigences en matière de performances de latence diffèrent par itinéraire, nous vous recommandons de créer des alarmes alternatives afin de surveiller plus précisément la latence pour chaque itinéraire.

Statistique : p90

Seuil recommandé : 2 500,0

Justification du seuil : la valeur de seuil suggérée ne fonctionne pas pour toutes les charges de travail de l'API. Elle peut toutefois être utilisée comme point de départ pour le seuil. Vous pouvez ensuite choisir différentes valeurs de seuil en fonction de la charge de travail et des exigences de latence, de performances et de SLA acceptables pour l'API. S'il est acceptable que l'API présente une latence plus élevée en général, vous pouvez définir une valeur de seuil plus élevée pour la rendre moins sensible. Toutefois, si l'API est censée fournir des réponses en temps quasi réel, définissez une valeur de seuil inférieure. Vous pouvez également analyser les

données historiques afin de déterminer la latence de référence attendue pour la charge de travail de l'application, puis ajuster la valeur du seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Latence

Dimensions : scèneApiId, ressource, méthode

Description de l'alarme : cette alarme détecte une latence élevée pour une route au cours d'une étape. Recherchez la valeur de la métrique `IntegrationLatency` pour vérifier la latence du backend de l'API. Si les deux métriques sont globalement alignées, le backend de l'API est à l'origine d'une latence plus élevée et doit être examiné pour détecter des problèmes de performance. Envisagez également d'activer CloudWatch les journaux et de vérifier les erreurs susceptibles d'être à l'origine de cette latence élevée. Vous pouvez également consulter le guide de résolution des problèmes d'intégration à Lambda, le cas échéant.

Intention : cette alarme est utilisée pour détecter les cas où les demandes d'API Gateway pour une route au cours d'une étape présentent une latence élevée.

Statistique : p90

Seuil recommandé : 2 500,0

Justification du seuil : la valeur de seuil suggérée ne fonctionne pas pour toutes les charges de travail de l'API. Elle peut toutefois être utilisée comme point de départ pour le seuil. Vous pouvez ensuite choisir différentes valeurs de seuil en fonction de la charge de travail et des exigences de latence, de performances et de SLA acceptables pour l'API. S'il est acceptable que l'API ait une latence plus élevée en général, vous pouvez définir une valeur de seuil plus élevée pour rendre l'alarme moins sensible. Toutefois, si l'API est censée fournir des réponses en temps quasi réel, définissez une valeur de seuil inférieure. Vous pouvez également analyser les données historiques afin de déterminer la latence de référence attendue pour la charge de travail de l'application, puis ajuster la valeur du seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

4xx

Dimensions :ApiId, scène

Description de l'alarme : cette alarme détecte un taux élevé d'erreurs côté client. Cela peut indiquer un problème dans les paramètres d'autorisation ou de requête client. Cela peut également signifier qu'une route a été supprimée ou qu'un client en demande une qui n'existe pas dans l'API. Envisagez d'activer les CloudWatch journaux et de vérifier les erreurs susceptibles d'être à l'origine des erreurs 4xx. En outre, pensez à activer CloudWatch les métriques détaillées pour afficher cette métrique par itinéraire, afin de vous aider à affiner la source des erreurs. Des erreurs peuvent également être causées par le dépassement de la limite configurée. Si les réponses et les journaux signalent des taux élevés et inattendus d'erreurs 429, suivez ce guide pour résoudre ce problème.

Intention : cette alarme peut détecter des taux élevés d'erreurs côté client pour les requêtes API Gateway.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : le seuil suggéré détecte les cas où plus de 5 % du total des requêtes reçoivent des erreurs 4xx. Cependant, vous pouvez ajuster le seuil en fonction du trafic des requêtes et des taux d'erreur acceptables. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs 4xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

5xx

Dimensions :ApiId, scène

Description de l'alarme : cette alarme permet de détecter un taux élevé d'erreurs côté serveur. Cela peut indiquer qu'il y a un problème sur le backend de l'API, le réseau ou l'intégration entre la passerelle d'API et l'API du backend. Cette documentation peut vous aider à résoudre la cause des erreurs 5xx.

Intention : cette alarme peut détecter des taux élevés d'erreurs côté serveur pour les requêtes API Gateway.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : le seuil suggéré détecte les cas où plus de 5 % du total des requêtes reçoivent des erreurs 5xx. Cependant, vous pouvez ajuster le seuil en fonction du trafic des requêtes et des taux d'erreur acceptables. Vous pouvez également analyser les données historiques pour déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs 5xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

MessageCount

Dimensions :ApiId, scène

Description de l'alarme : Cette alarme permet de détecter un faible volume de trafic pour la phase WebSocket API. Cela peut indiquer un problème lorsque les clients appellent l'API, par exemple en utilisant des points de terminaison incorrects ou en raison de problèmes liés à l'envoi de messages par le backend aux clients. Cela peut également indiquer un problème lié à la configuration ou aux autorisations de l'API, la rendant inaccessible aux clients.

Intention : Cette alarme peut détecter un volume de trafic étonnamment faible pour l'étape de l' WebSocket API. Nous vous recommandons de créer cette alarme si votre API reçoit et envoie un nombre prévisible et régulier de messages dans des conditions normales. Si CloudWatch les métriques détaillées sont activées et que vous pouvez prévoir le volume de trafic normal par

itinéraire, il est préférable de créer des alarmes alternatives à celle-ci, afin d'avoir un suivi plus précis des baisses de volume de trafic pour chaque itinéraire. Nous ne recommandons pas cette alarme pour les API qui ne s'attendent pas à un trafic constant et régulier.

Statistique : SampleCount

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez la valeur du seuil en fonction de l'analyse des données historiques afin de déterminer le nombre de messages de référence attendu pour votre API. Le réglage du seuil à une valeur très élevée peut rendre l'alarme trop sensible en période de faible trafic normal et prévu. À l'inverse, si vous le réglez sur une valeur très faible, l'alarme risque de ne pas détecter de petites baisses anormales du volume de trafic.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

MessageCount

Dimensions : étapeApiId, itinéraire

Description de l'alarme : Cette alarme permet de détecter un faible volume de trafic pour la route de l' WebSocket API au cours de la phase. Cela peut indiquer un problème lié à l'appel de l'API par les clients, par exemple en utilisant des points de terminaison incorrects, ou des problèmes liés à l'envoi de messages par le backend aux clients. Cela peut également indiquer un problème lié à la configuration ou aux autorisations de l'API, la rendant inaccessible aux clients.

Intention : Cette alarme peut détecter un volume de trafic étonnamment faible pour la route de l' WebSocket API au cours de la phase. Nous vous recommandons de créer cette alarme si votre API reçoit et envoie un nombre prévisible et régulier de messages dans des conditions normales. Nous ne recommandons pas cette alarme pour les API qui ne s'attendent pas à un trafic constant et régulier.

Statistique : SampleCount

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction de l'analyse des données historiques afin de déterminer le nombre de messages de référence attendu pour votre API. Le réglage du seuil à une valeur très élevée peut rendre l'alarme trop sensible en période de faible trafic normal et prévu. À l'inverse, si vous le réglez sur une valeur très faible, l'alarme risque de ne pas détecter de petites baisses anormales du volume de trafic.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

ClientError

Dimensions :ApiId, scène

Description de l'alarme : cette alarme détecte un taux élevé d'erreurs client. Cela peut indiquer un problème dans les paramètres d'autorisation ou de message. Cela peut également signifier qu'une route a été supprimée ou qu'un client en demande une qui n'existe pas dans l'API. Envisagez d'activer les CloudWatch journaux et de vérifier les erreurs susceptibles d'être à l'origine des erreurs 4xx. En outre, pensez à activer CloudWatch les métriques détaillées pour afficher cette métrique par itinéraire, afin de vous aider à affiner la source des erreurs. Des erreurs peuvent également être causées par le dépassement de la limite configurée. Si les réponses et les journaux signalent des taux élevés et inattendus d'erreurs 429, suivez ce guide pour résoudre ce problème.

Intention : Cette alarme peut détecter des taux élevés d'erreurs client pour les messages WebSocket API Gateway.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : le seuil suggéré détecte les cas où plus de 5 % du total des requêtes reçoivent des erreurs 4xx. Vous pouvez ajuster le seuil en fonction du trafic des requêtes ainsi qu'en fonction de vos taux d'erreur acceptables. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs 4xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ExecutionError

Dimensions :ApiId, scène

Description de l'alarme : cette alarme permet de détecter un taux élevé d'erreurs d'exécution. Cela peut être dû à des erreurs 5xx liées à votre intégration, à des problèmes d'autorisation ou à d'autres facteurs empêchant l'invocation réussie de l'intégration, tels que la limitation ou la suppression de l'intégration. Envisagez d'activer les CloudWatch journaux pour votre API et de vérifier le type et la cause des erreurs dans les journaux. En outre, pensez à activer CloudWatch les métriques détaillées pour avoir une vue de cette métrique par itinéraire, afin de vous aider à affiner la source des erreurs. Cette documentation peut également vous aider à résoudre la cause de toute erreur de connexion.

Intention : Cette alarme peut détecter des taux élevés d'erreurs d'exécution pour les messages WebSocket API Gateway.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : le seuil suggéré détecte les cas où plus de 5 % du total des requêtes présentent des erreurs d'exécution. Vous pouvez ajuster le seuil en fonction du trafic des requêtes, ainsi qu'en fonction de vos taux d'erreur acceptables. Vous pouvez analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs d'exécution fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon EC2 Auto Scaling

GroupInServiceCapacity

Dimensions : AutoScalingGroupName

Description de l'alarme : cette alarme permet de détecter lorsque la capacité du groupe est inférieure à la capacité requise pour votre charge de travail. Pour résoudre le problème, vérifiez vos activités de dimensionnement pour détecter les échecs de lancement et confirmez que la configuration de capacité souhaitée est correcte.

Objectif : cette alarme peut détecter une faible disponibilité dans votre groupe Auto Scaling en raison d'échecs de lancement ou de lancements suspendus.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur du seuil doit être la capacité minimale requise pour exécuter votre charge de travail. Dans la plupart des cas, vous pouvez le configurer pour qu'il corresponde à la GroupDesiredCapacity métrique.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

Amazon CloudFront

5 xxErrorRate

Dimensions :DistributionId, Région=Global

Description de l'alarme : Cette alarme surveille le pourcentage de réponses d'erreur 5xx provenant de votre serveur d'origine, afin de vous aider à détecter si le CloudFront service rencontre des problèmes. Veuillez consulter la section Résolution des réponses d'erreur de votre origine pour obtenir des informations qui vous aideront à comprendre les problèmes liés à votre serveur. En outre, l'activation de métriques supplémentaires permet d'obtenir des métriques d'erreur détaillées.

Intention : Cette alarme est utilisée pour détecter les problèmes liés au traitement des demandes provenant du serveur d'origine ou les problèmes de communication entre le serveur d'origine CloudFront et votre serveur d'origine.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de la tolérance pour les réponses 5xx. Vous pouvez analyser les données historiques et les tendances, puis définir le seuil en conséquence. Les erreurs 5xx pouvant être causées par des problèmes transitoires, nous vous recommandons de définir le seuil sur une valeur supérieure à 0 afin que l'alarme ne soit pas trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

OriginLatency

Dimensions :DistributionId, Région=Global

Description de l'alarme : l'alarme permet de vérifier si le serveur d'origine met trop de temps à répondre. Si le serveur met trop de temps à répondre, cela peut entraîner un délai d'expiration. Référez-vous à recherchez et corrigez des réponses retardées à partir des applications sur votre serveur d'origine si vous rencontrez des valeurs `OriginLatency` constamment élevées.

Intention : cette alarme est utilisée pour détecter les problèmes liés au fait que le serveur d'origine met trop de temps à répondre.

Statistique : p90

Seuil recommandé : dépend de votre situation

Justification du seuil : vous devez calculer la valeur d'environ 80 % du délai d'expiration de la réponse d'origine et utiliser le résultat comme valeur de seuil. Si cette métrique est toujours proche de la valeur du délai d'expiration de la réponse d'origine, il se peut que vous commenciez à rencontrer des erreurs 504.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

FunctionValidationErrors

Dimensions :DistributionId, FunctionName, Région=Global

Description de l'alarme : Cette alarme vous aide à surveiller les erreurs de validation CloudFront des fonctions afin que vous puissiez prendre des mesures pour les résoudre. Analysez les journaux des CloudWatch fonctions et examinez le code de la fonction pour trouver et résoudre la cause première du problème. Consultez la section Restrictions relatives aux fonctions de périphérie pour comprendre les erreurs de configuration courantes des CloudFront fonctions.

Intention : Cette alarme est utilisée pour détecter les erreurs de validation des CloudFront fonctions.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : une valeur supérieure à 0 indique une erreur de validation. Nous vous recommandons de définir le seuil à 0 car les erreurs de validation impliquent un problème lorsque CloudFront les fonctions repassent à CloudFront. Par exemple, CloudFront nécessite l'en-tête HTTP Host pour traiter une demande. Rien n'empêche un utilisateur de supprimer l'en-tête Host dans son code de CloudFront fonctions. Mais lorsque vous CloudFront recevez la réponse et que l'en-tête Host est manquant, une CloudFront erreur de validation est générée.

Période : 60

Points de données pour le déclenchement d'alarme : 2

Période d'évaluation : 2

Opérateur de comparaison : GREATER_THAN_THRESHOLD

FunctionExecutionErrors

Dimensions :DistributionId, FunctionName, Région=Global

Description de l'alarme : Cette alarme vous aide à surveiller les erreurs d'exécution CloudFront des fonctions afin que vous puissiez prendre des mesures pour les résoudre. Analysez les journaux des CloudWatch fonctions et examinez le code de la fonction pour trouver et résoudre la cause première du problème.

Intention : Cette alarme est utilisée pour détecter les erreurs d'exécution des CloudFront fonctions.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : nous recommandons de définir le seuil à 0, car une erreur d'exécution indique un problème avec le code qui se produit lors de l'exécution.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

FunctionThrottles

Dimensions :DistributionId, FunctionName, Région=Global

Description de l'alarme : Cette alarme vous permet de vérifier si votre CloudFront fonction est limitée. Si votre fonction est limitée, cela signifie que son exécution prend trop de temps. Pour éviter les limitations de fonction, envisagez d'optimiser le code de la fonction.

Objectif : Cette alarme peut détecter lorsque votre CloudFront fonction est limitée afin que vous puissiez réagir et résoudre le problème pour une expérience client fluide.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : nous recommandons de définir le seuil à 0, afin de permettre une résolution plus rapide des limitations de fonction.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon Cognito

SignUpThrottles

Dimensions :UserPool, UserPoolClient

Description de l'alarme : cette alarme surveille le nombre de requêtes limitées. Si les utilisateurs sont constamment limités, vous devez augmenter la limite en demandant une augmentation du quota de service. Consultez Quotas dans Amazon Cognito pour savoir comment demander une augmentation des quotas. Pour prendre des mesures proactives, pensez à suivre l'usage des quotas.

Intention : cette alarme permet de surveiller l'apparition de demandes d'inscription limitées. Cela peut vous aider à savoir quand prendre des mesures pour atténuer toute dégradation de l'expérience d'inscription. La limitation persistante des demandes est une expérience négative pour l'inscription des utilisateurs.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : un groupe d'utilisateurs bien dimensionné ne devrait pas être soumis à une limitation qui s'étend sur plusieurs points de données. Ainsi, le seuil typique pour une charge de travail attendue doit être de zéro. Pour une charge de travail irrégulière avec des pics fréquents, vous pouvez analyser les données historiques afin de déterminer la limitation acceptable pour la charge de travail de l'application, puis vous pouvez ajuster le seuil en conséquence. Une demande limitée doit être réitérée afin de minimiser l'impact sur l'application.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

SignInThrottles

Dimensions :UserPool, UserPoolClient

Description de l'alarme : cette alarme surveille le nombre de demandes d'authentification utilisateur limitées. Si les utilisateurs sont constamment limités, vous devrez peut-être augmenter la limite en demandant une augmentation du quota de service. Consultez Quotas dans Amazon Cognito pour savoir comment demander une augmentation des quotas. Pour prendre des mesures proactives, pensez à suivre l'usage des quotas.

Intention : cette alarme permet de surveiller l'apparition de demandes de connexion limitées. Cela peut vous aider à savoir quand prendre des mesures pour atténuer toute dégradation de l'expérience de connexion. La limitation persistante des demandes est une mauvaise expérience d'authentification des utilisateurs.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : un groupe d'utilisateurs bien dimensionné ne devrait pas être soumis à une limitation qui s'étend sur plusieurs points de données. Ainsi, le seuil typique pour une charge de travail attendue doit être de zéro. Pour une charge de travail irrégulière avec des pics fréquents, vous pouvez analyser les données historiques afin de déterminer la limitation acceptable pour la charge de travail de l'application, puis vous pouvez ajuster le seuil en conséquence. Une demande limitée doit être réitérée afin de minimiser l'impact sur l'application.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

TokenRefreshThrottles

Dimensions :UserPool, UserPoolClient

Description de l'alarme : vous pouvez définir la valeur seuil en fonction du trafic de la requête et pour qu'elle corresponde à une limitation acceptable pour les requêtes d'actualisation des jetons. La limitation est utilisée pour protéger votre système contre un trop grand nombre de requêtes. Cependant, il est également important de vérifier que vous n'êtes pas en situation de sous-

allocation pour votre trafic normal. Vous pouvez analyser les données historiques pour trouver la limitation acceptable pour la charge de travail de l'application, puis vous pouvez régler votre seuil d'alarme pour qu'il soit supérieur à votre niveau de limitation acceptable. Les requêtes limitées doivent être réessayées par l'application/le service, car elles sont transitoires. Par conséquent, une valeur très faible du seuil peut rendre l'alarme sensible.

Intention : cette alarme permet de surveiller l'occurrence de requêtes d'actualisation de jetons limitées. Cela peut vous aider à savoir quand prendre des mesures pour atténuer les problèmes potentiels, afin de garantir une expérience utilisateur fluide ainsi que le bon fonctionnement et la fiabilité de votre système d'authentification. La limitation persistante des demandes est une mauvaise expérience d'authentification des utilisateurs.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur du seuil peut également être définie/ajustée en fonction du trafic de la requête ainsi que d'une limitation acceptable pour les requêtes d'actualisation des jetons. Les limitations sont là pour protéger votre système contre un trop grand nombre de requêtes, mais il est également important de vérifier que vous n'êtes pas en situation de sous-allocation pour votre trafic normal et de voir si cela en est la cause. Les données historiques peuvent également être analysées pour déterminer le niveau de limitation acceptable pour la charge de travail de l'application et le seuil peut être réglé à un niveau plus élevé que votre niveau de limitation acceptable habituel. Les requêtes limitées doivent être réessayées par l'application/le service, car elles sont transitoires. Par conséquent, une valeur très faible du seuil peut rendre l'alarme sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

FederationThrottles

UserPoolDimensioni : UserPoolClient, IdentityProvider

Description de l'alarme : cette alarme surveille le nombre de requêtes de fédération d'identité limitées. Si vous constatez régulièrement des limitations, cela peut indiquer que vous devez

augmenter la limite en demandant une augmentation du quota de service. Consultez Quotas dans Amazon Cognito pour savoir comment demander une augmentation des quotas.

Intention : cette alarme permet de surveiller l'apparition de requêtes de fédération d'identité limitées. Cela peut vous aider à apporter des réponses proactives aux problèmes de performance ou aux erreurs de configuration, et à garantir une expérience d'authentification fluide pour vos utilisateurs. La limitation persistante des demandes est une mauvaise expérience d'authentification des utilisateurs.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : vous pouvez définir le seuil en fonction du trafic de la requête et pour qu'il corresponde à la limitation acceptable pour les requêtes de fédération d'identité. La limitation est utilisée pour protéger votre système contre un trop grand nombre de requêtes. Cependant, il est également important de vérifier que vous n'êtes pas en situation de sous-allocation pour votre trafic normal. Vous pouvez analyser les données historiques pour trouver la limitation acceptable pour la charge de travail de l'application, puis vous pouvez régler votre seuil d'alerte pour qu'il soit supérieur à votre niveau de limitation acceptable. Les requêtes limitées doivent être réessayées par l'application/le service, car elles sont transitoires. Par conséquent, une valeur très faible du seuil peut rendre l'alarme sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon DynamoDB

AccountProvisionedReadCapacityUtilization

Dimensions : Aucune

Description de l'alarme : cette alarme détecte si la capacité de lecture du compte atteint sa limite allouée. Vous pouvez augmenter le quota du compte pour le taux d'utilisation de la capacité de lecture si cela se produit. Vous pouvez consulter vos quotas actuels pour les unités de capacité de lecture et demander des augmentations à l'aide des Service Quotas.

Objectif : l'alarme peut détecter si le taux d'utilisation de la capacité de lecture du compte est proche de celle de la capacité de lecture allouée. Si le taux d'utilisation atteint sa limite maximale, DynamoDB commence à limiter les requêtes de lecture.

Statistique : maximum

Seuil recommandé : 80,0

Justification du seuil : fixez le seuil à 80 %, pour que des mesures (telles que l'augmentation des limites du compte) puissent être prises avant que le compte n'atteigne sa pleine capacité afin d'éviter tout ralentissement.

Période : 300

Points de données pour le déclenchement d'alarme : 2

Période d'évaluation : 2

Opérateur de comparaison : GREATER_THAN_THRESHOLD

AccountProvisionedWriteCapacityUtilization

Dimensions : Aucune

Description de l'alarme : cette alarme détecte si la capacité d'écriture du compte atteint sa limite allouée. Vous pouvez augmenter le quota du compte pour le taux d'utilisation de la capacité d'écriture si cela se produit. Vous pouvez consulter vos quotas actuels pour les unités de capacité d'écriture et demander des augmentations à l'aide des Service Quotas.

Intention : cette alarme peut détecter si le taux d'utilisation de la capacité d'écriture du compte est proche de celle de la capacité d'écriture allouée. Si le taux d'utilisation atteint sa limite maximale, DynamoDB commence à limiter les requêtes d'écriture.

Statistique : maximum

Seuil recommandé : 80,0

Justification du seuil : fixez le seuil à 80 %, pour que des mesures (telles que l'augmentation des limites du compte) puissent être prises avant que le compte n'atteigne sa pleine capacité afin d'éviter tout ralentissement.

Période : 300

Points de données pour le déclenchement d'alarme : 2

Période d'évaluation : 2

Opérateur de comparaison : GREATER_THAN_THRESHOLD

AgeOfOldestUnreplicatedRecord

Dimensions :TableName, DelegatedOperation

Description de l'alarme : cette alarme détecte le retard de réplication vers un flux de données Kinesis. Dans des conditions normales de fonctionnement, la valeur `AgeOfOldestUnreplicatedRecord` ne devrait être que de quelques millisecondes. Ce nombre augmente en fonction des tentatives de réplication infructueuses causées par des choix de configuration contrôlés par le client. Les exemples de configuration contrôlés par le client qui conduisent à des tentatives de réplication infructueuses sont une capacité de flux de données Kinesis sous-allouée qui conduit à une limitation excessive ou une mise à jour manuelle des stratégies d'accès du flux de données Kinesis qui empêche DynamoDB d'ajouter des données au flux de données. Pour maintenir cette métrique à un niveau aussi bas que possible, vous devez veiller à une allocation correcte de la capacité du flux de données Kinesis et vous assurer que les autorisations de DynamoDB sont inchangées.

Intention : cette alarme peut surveiller les tentatives de réplication infructueuses et le retard de réplication qui en résulte dans le flux de données Kinesis.

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du délai de réplication souhaité, mesuré en millisecondes. Cette valeur dépend des exigences de votre charge de travail et des performances attendues.

Période : 300

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

FailedToReplicateRecordCount

Dimensions :TableName, DelegatedOperation

Description de l'alarme : cette alarme détecte le nombre de registres que DynamoDB n'a pas pu répliquer sur votre flux de données Kinesis. Certains éléments de plus de 34 Ko peuvent augmenter en taille pour modifier les registres de données supérieurs à la limite de taille d'élément de 1 Mo de Kinesis Data Streams. Cette augmentation de taille se produit lorsque les éléments de plus de 34 Ko incluent un grand nombre de valeurs d'attribut booléennes ou vides. Les valeurs d'attribut booléennes et vides sont stockées sous la forme d'un octet dans DynamoDB. Toutefois, elles peuvent atteindre 5 octets lorsqu'elles sont sérialisées à l'aide de JSON standard pour la réplication Kinesis Data Streams. DynamoDB ne peut pas répliquer de tels registres de modification dans votre flux de données Kinesis. DynamoDB ignore ces registres de données de modification et continue automatiquement la réplication des registres suivants.

Intention : cette alarme peut surveiller le nombre de registres que DynamoDB n'a pas pu répliquer sur votre flux de données Kinesis en raison de la limite de taille des éléments de Kinesis Data Streams.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : définissez le seuil sur 0 pour détecter les enregistrements que DynamoDB n'a pas réussi à répliquer.

Période : 60

Points de données pour le déclenchement d'alarme : 1

Période d'évaluation : 1

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensions : TableName

Description de l'alarme : cette alarme détecte si un grand nombre de requêtes de lecture sont limitées pour la table DynamoDB. Pour résoudre ce problème, veuillez consulter Résolution des problèmes de limitation dans Amazon DynamoDB.

Intention : cette alarme peut détecter une limitation persistante des requêtes de lecture adressées à la table DynamoDB. Une limitation persistante des requêtes de lecture peut avoir un impact négatif sur les opérations de lecture de votre charge de travail et réduire l'efficacité globale du système.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du trafic de lecture attendu pour la table DynamoDB, en tenant compte d'un niveau de limitation acceptable. Il est important de vérifier si vous êtes sous-approvisionné et si vous ne provoquez pas une limitation régulière. Vous pouvez également analyser les données historiques pour trouver le niveau de limitation acceptable pour la charge de travail de l'application, puis régler le seuil pour qu'il soit supérieur à votre niveau de limitation habituel. Les requêtes limitées doivent être réitérées par l'application ou le service, car elles sont transitoires. Par conséquent, un seuil très bas peut rendre l'alarme trop sensible et provoquer des transitions d'état indésirables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ReadThrottleEvents

Dimensions :TableName, GlobalSecondaryIndexName

Description de l'alarme : cette alarme détecte si un nombre élevé de requêtes de lecture sont limitées pour l'index secondaire global de la table DynamoDB. Pour résoudre ce problème, veuillez consulter Résolution des problèmes de limitation dans Amazon DynamoDB.

Objectif : l'alarme peut détecter une limitation persistante des requêtes de lecture pour l'index secondaire global de la table DynamoDB. Une limitation persistante des requêtes de lecture peut avoir un impact négatif sur les opérations de lecture de votre charge de travail et réduire l'efficacité globale du système.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du trafic de lecture attendu pour la table DynamoDB, en tenant compte d'un niveau de limitation acceptable. Il est important de vérifier si vous êtes sous-approvisionné et si vous ne provoquez pas une limitation régulière. Vous pouvez également analyser les données historiques pour trouver un niveau de limitation acceptable pour

la charge de travail de l'application, puis régler le seuil pour qu'il soit supérieur à votre niveau de limitation acceptable habituel. Les requêtes limitées doivent être réitérées par l'application ou le service, car elles sont transitoires. Par conséquent, un seuil très bas peut rendre l'alarme trop sensible et provoquer des transitions d'état indésirables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ReplicationLatency

Dimensions :TableName, ReceivingRegion

Description de l'alarme : l'alarme détecte si le réplica d'une région pour la table globale est en retard par rapport à la région source. La latence peut augmenter si une AWS région se dégrade et que vous disposez d'une table de réplication dans cette région. Dans ce cas, vous pouvez rediriger temporairement les activités de lecture et d'écriture de votre application vers une autre AWS région. Si vous utilisez 2017.11.29 (ancienne) de tables globales, vérifiez que les unités de capacité d'écriture (WCU) sont identiques pour chaque réplica de table. Vous pouvez également vous assurer de suivre les recommandations de la section Bonnes pratiques et exigences pour la gestion de la capacité.

Intention : l'alarme peut détecter si le réplica de table d'une région est en retard par rapport à la réplication des modifications d'une autre région. Cela pourrait entraîner une divergence entre votre réplica et les autres réplicas. Il est utile de connaître la latence de réplication de chaque AWS région et d'avertir si cette latence de réplication augmente continuellement. La réplication de la table s'applique uniquement aux tables globales.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de votre cas d'utilisation. Les latences de réplication supérieures à 3 minutes font généralement l'objet d'une enquête. Passez en revue le caractère critique et les exigences du délai de réplication et analysez les tendances historiques, puis sélectionnez le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

SuccessfulRequestLatency

Dimensions :TableName, Fonctionnement

Description de l'alarme : cette alarme détecte une latence élevée pour le fonctionnement de la table DynamoDB (indiquée par la valeur de la dimension de l'`0peration` dans l'alarme). Consultez ce document de résolution des problèmes de latence dans Amazon DynamoDB.

Objectif : cette alarme peut détecter une latence élevée pour le fonctionnement de la table DynamoDB. Une latence plus élevée pour les opérations peut avoir un impact négatif sur l'efficacité globale du système.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : DynamoDB fournit une latence moyenne d'un chiffre en millisecondes pour les opérations singleton telles que,, etc. GetItem PutItem Toutefois, vous pouvez définir le seuil en fonction d'une tolérance acceptable pour la latence en fonction du type d'opération et de la table concernés par la charge de travail. Vous pouvez analyser les données historiques de cette métrique afin de déterminer la latence habituelle pour l'opération sur la table, puis définir le seuil sur un nombre qui représente le délai critique pour l'opération.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_THRESHOLD

SystemErrors

Dimensions : TableName

Description de l'alarme : cette alarme détecte un nombre élevé et persistant d'erreurs système pour les requêtes de tables DynamoDB. Si les erreurs 5xx persistent, ouvrez le tableau de bord de l'état d'un service AWS pour vérifier l'absence de problèmes opérationnels liés au

service. Vous pouvez utiliser cette alarme pour être averti en cas de problème de service interne prolongé lié à DynamoDB et elle vous aide à établir une corrélation avec le problème rencontré par votre application cliente. Veuillez consulter Gestion des erreurs avec DynamoDB pour plus d'informations.

Objectif : cette alarme peut détecter des erreurs système persistantes pour les requêtes de table DynamoDB. Les erreurs système indiquent des erreurs de service internes provenant de DynamoDB et permettent d'établir une corrélation avec le problème rencontré par le client.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du trafic attendu, en tenant compte d'un niveau acceptable d'erreurs système. Vous pouvez également analyser les données historiques pour déterminer le nombre d'erreurs acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs système doivent être réitérées par l'application/le service, car elles sont transitoires. Par conséquent, un seuil très bas peut rendre l'alarme trop sensible et provoquer des transitions d'état indésirables.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ThrottledPutRecordCount

Dimensions :TableName, DelegatedOperation

Description de l'alarme : cette alarme détecte les enregistrements limités par votre flux de données Kinesis lors de la réplication de la capture des données de modification vers Kinesis. Cette limitation est due à une capacité de flux de données Kinesis insuffisante. Si vous rencontrez une limitation excessive et régulière, il se peut que vous deviez augmenter le nombre de partitions de flux Kinesis en proportion du débit d'écriture observé de votre table. Pour en savoir plus sur la détermination de la taille d'un flux de données Kinesis, consultez Détermination de la taille initiale d'un flux de données Kinesis.

Intention : cette alarme peut surveiller le nombre de registres qui ont été limités par votre flux de données Kinesis en raison d'une capacité insuffisante de flux de données Kinesis.

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : vous pourriez rencontrer une certaine limitation lors de pics d'utilisation exceptionnels, mais les enregistrements limités devraient rester aussi bas que possible pour éviter une latence de réplication plus élevée (DynamoDB tente à nouveau d'envoyer des enregistrements limités au flux de données Kinesis). Définissez le seuil sur une valeur qui peut vous aider à détecter une limitation excessive régulière. Vous pouvez également analyser les données historiques de cette métrique afin de déterminer les taux de limitation acceptables pour la charge de travail de l'application. Ajustez le seuil à une valeur tolérée par l'application en fonction de votre cas d'utilisation.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_THRESHOLD

UserErrors

Dimensions : Aucune

Description de l'alarme : cette alarme détecte un nombre élevé et persistant d'erreurs utilisateur concernant les requêtes de table DynamoDB. Vous pouvez consulter les journaux des applications clientes pendant la période d'émission pour voir pourquoi les requêtes ne sont pas valides. Vous pouvez vérifier le code d'état HTTP 400 pour voir le type d'erreur que vous recevez et prendre les mesures nécessaires. Vous devrez peut-être corriger la logique de l'application pour créer des requêtes valides.

Intention : cette alarme peut détecter des erreurs utilisateur persistantes concernant les requêtes de table DynamoDB. Les erreurs de l'utilisateur liées aux opérations demandées signifient que le client produit des requêtes non valides et qu'elles échouent.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil à zéro pour détecter toute erreur côté client. Vous pouvez également le régler sur une valeur plus élevée si vous souhaitez éviter que l'alarme ne se

déclenche en cas de très faible nombre d'erreurs. Décidez en fonction de votre cas d'utilisation et du trafic pour les requêtes.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensions : TableName

Description de l'alarme : cette alarme détecte si un grand nombre de requêtes d'écriture sont limitées pour la table DynamoDB. Veuillez consulter Résolution des problèmes de limitation dans Amazon DynamoDB pour résoudre ce problème.

Intention : cette alarme peut détecter une limitation persistante des requêtes d'écriture dans la table DynamoDB. La limitation persistante des requêtes d'écriture peut avoir un impact négatif sur les opérations d'écriture de votre charge de travail et réduire l'efficacité globale du système.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du trafic d'écriture attendu pour la table DynamoDB, en tenant compte d'un niveau de limitation acceptable. Il est important de vérifier si vous êtes sous-approvisionné et si vous ne provoquez pas une limitation régulière. Vous pouvez également analyser les données historiques pour trouver un niveau de limitation acceptable pour la charge de travail de l'application, puis régler le seuil pour qu'il soit supérieur à votre niveau de limitation acceptable habituel. Les requêtes limitées doivent être réessayées par l'application/le service, car elles sont transitoires. Par conséquent, un seuil très bas peut rendre l'alarme trop sensible et provoquer des transitions d'état indésirables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

WriteThrottleEvents

Dimensions :TableName, GlobalSecondaryIndexName

Description de l'alarme : cette alarme détecte si un grand nombre de requêtes d'écriture sont limitées pour l'index secondaire global de la table DynamoDB. Veuillez consulter Résolution des problèmes de limitation dans Amazon DynamoDB pour résoudre ce problème.

Intention : cette alarme peut détecter une limitation persistante des requêtes d'écriture pour l'index secondaire global de la table DynamoDB. La limitation persistante des requêtes d'écriture peut avoir un impact négatif sur les opérations d'écriture de votre charge de travail et réduire l'efficacité globale du système.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du trafic d'écriture attendu pour la table DynamoDB, en tenant compte d'un niveau de limitation acceptable. Il est important de vérifier si vous êtes sous-approvisionné et si vous ne provoquez pas une limitation régulière. Vous pouvez également analyser les données historiques pour trouver le niveau de limitation acceptable pour la charge de travail de l'application, puis régler le seuil à une valeur supérieure à votre niveau de limitation acceptable habituel. Les requêtes limitées doivent être réessayées par l'application/le service, car elles sont transitoires. Par conséquent, une valeur très faible peut rendre l'alarme trop sensible et provoquer des transitions d'état indésirables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon EBS

VolumeStalledIOcheck

Dimensions :VolumeId, InstanceId

Description de l'alarme : Cette alarme vous permet de surveiller les performances d'E/S de vos volumes Amazon EBS. Cette vérification détecte les problèmes sous-jacents liés à l'infrastructure Amazon EBS, tels que les problèmes matériels ou logiciels sur les sous-systèmes de stockage sous-jacents aux volumes Amazon EBS, les problèmes matériels sur l'hôte physique qui ont un impact sur l'accessibilité des volumes Amazon EBS depuis votre instance Amazon EC2, et permet de détecter les problèmes de connectivité entre l'instance et les volumes Amazon EBS. Si la vérification des E/S bloqués échoue, vous pouvez soit attendre AWS que le problème soit résolu, soit prendre des mesures telles que le remplacement du volume concerné ou l'arrêt et le redémarrage de l'instance à laquelle le volume est connecté. Dans la plupart des cas, lorsque cette métrique échoue, Amazon EBS diagnostique et restaure automatiquement votre volume en quelques minutes.

Intention : Cette alarme peut détecter l'état de vos volumes Amazon EBS afin de déterminer à quel moment ces volumes sont altérés et ne peuvent pas terminer les opérations d'E/S.

Statistique : maximum

Seuil recommandé : 1,0

Justification du seuil : lorsqu'une vérification de l'état échoue, la valeur de cette métrique est 1. Le seuil est réglé de telle sorte que chaque fois que le vérification de l'état échoue, l'alarme soit en état ALARM.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EC2

CPUUtilization

Dimensions : InstanceId

Description de l'alarme : cette alarme permet de surveiller le taux d'utilisation du processeur d'une instance EC2. Selon l'application, des niveaux d'utilisation élevés et constants peuvent être normaux. Mais si les performances sont dégradées et que l'application n'est pas limitée par les E/

S du disque, la mémoire ou les ressources réseau, un processeur au maximum peut indiquer un goulot d'étranglement des ressources ou des problèmes de performance de l'application. Un taux d'utilisation élevé du processeur peut indiquer qu'une mise à niveau vers une instance utilisant un CPU plus puissant est nécessaire. Si la surveillance détaillée est activée, vous pouvez modifier la période à 60 secondes au lieu de 300 secondes. Pour plus d'informations, veuillez consulter Activer ou désactiver la surveillance détaillée pour vos instances.

Objectif : cette alarme est utilisée pour détecter un taux d'utilisation élevé du processeur.

Statistique : moyenne

Seuil recommandé : 80,0

Justification du seuil : vous pouvez généralement définir le seuil d'utilisation du processeur entre 70 et 80 %. Toutefois, vous pouvez ajuster cette valeur en fonction de votre niveau de performance acceptable et des caractéristiques de charge de travail. Pour certains systèmes, une utilisation constamment élevée du processeur peut être normale et ne pas indiquer un problème, tandis que pour d'autres, cela peut être une source de préoccupation. Analysez les données historiques d'utilisation du processeur pour identifier l'utilisation, déterminer quelle utilisation du processeur est acceptable pour votre système et définissez le seuil en conséquence.

Période : 300

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

StatusCheckFailed

Dimensions : InstanceId

Description de l'alarme : cette alarme permet de surveiller à la fois les vérifications de l'état de système et les contrôles de statut d'instance. Si l'un ou l'autre type de vérification de l'état échoue, cette alarme doit être en état ALARM.

Objectif : cette alarme est utilisée pour détecter les problèmes sous-jacents liés aux instances, notamment les échecs de vérification de l'état du système et les échecs de vérification de l'état des instances.

Statistique : maximum

Seuil recommandé : 1,0

Justification du seuil : lorsqu'une vérification de l'état échoue, la valeur de cette métrique est 1. Le seuil est réglé de telle sorte que chaque fois que le vérification de l'état échoue, l'alarme soit en état ALARM.

Période : 300

Points de données pour le déclenchement d'alarme : 2

Période d'évaluation : 2

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

StatusCheckFailed_EBS attachés

Dimensions : InstanceId

Description de l'alarme : Cette alarme vous permet de vérifier si les volumes Amazon EBS attachés à une instance sont accessibles et capables d'effectuer des opérations d'E/S. Cette vérification d'état détecte les problèmes sous-jacents liés à l'infrastructure informatique ou Amazon EBS, tels que les suivants :

- Problèmes matériels ou logiciels sur les sous-systèmes de stockage sous-jacents aux volumes Amazon EBS
- Problèmes matériels sur l'hôte physique qui ont un impact sur l'accessibilité des volumes Amazon EBS
- Problèmes de connectivité entre l'instance et les volumes Amazon EBS

Lorsque la vérification du statut EBS jointe échoue, vous pouvez soit attendre qu'Amazon résolve le problème, soit prendre une mesure telle que le remplacement des volumes concernés ou l'arrêt et le redémarrage de l'instance.

Intention : Cette alarme est utilisée pour détecter les volumes Amazon EBS inaccessibles attachés à une instance. Ils peuvent provoquer des défaillances dans les opérations d'E/S.

Statistique : maximum

Seuil recommandé : 1,0

Justification du seuil : lorsqu'une vérification de l'état échoue, la valeur de cette métrique est 1. Le seuil est réglé de telle sorte que chaque fois que le vérification de l'état échoue, l'alarme soit en état ALARM.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Amazon ElastiCache

CPUUtilization

Dimensions :CacheClusterId, CacheNodeId

Description de l'alarme : Cette alarme permet de surveiller l'utilisation du processeur pour l'ensemble de l' ElastiCache instance, y compris les processus du moteur de base de données et les autres processus exécutés sur l'instance. AWS Elasticache prend en charge deux types de moteurs : Memcached et Redis. Lorsque vous atteignez un taux d'utilisation élevé du processeur sur un nœud Memcached, vous devez envisager d'augmenter le type d'instance ou d'ajouter de nouveaux nœuds de cache. Pour Redis, si votre charge de travail principale provient de requêtes de lecture, vous devriez envisager d'ajouter d'autres réplicas en lecture à votre cluster de cache. Si votre charge de travail principale provient de requêtes d'écriture, vous devriez envisager d'ajouter des partitions supplémentaires pour répartir la charge de travail sur un plus grand nombre de nœuds primaires si vous opérez en mode cluster, ou d'augmenter le type d'instance si vous exécutez Redis en mode non-cluster.

Objectif : Cette alarme est utilisée pour détecter une utilisation élevée du processeur par ElastiCache les hôtes. Il est utile d'avoir une vue d'ensemble de l'utilisation du processeur sur la totalité de l'instance, y compris les processus non liés au moteur.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil sur le pourcentage qui reflète un niveau d'utilisation critique du processeur pour votre application. Pour Memcached, le moteur peut utiliser jusqu'à un nombre de cœurs égal à num_threads. Pour Redis, le moteur est principalement monothread, mais peut utiliser des cœurs supplémentaires si disponibles pour accélérer les E/S. Dans la plupart des cas, vous pouvez définir le seuil à environ 90 % de votre processeur disponible.

Comme Redis est à thread unique, la valeur réelle du seuil doit être calculée en tant que fraction de la capacité totale du nœud.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

CurrConnections

Dimensions :CacheClusterId, CacheNodeId

Description de l'alarme : cette alarme détecte un nombre élevé de connexions, ce qui peut indiquer une charge importante ou des problèmes de performance. Une augmentation constante de `CurrConnections` pourrait entraîner l'épuisement des 65 000 connexions disponibles. Cela peut indiquer que les connexions se sont mal fermées du côté de l'application et ont été laissées établies du côté du serveur. Vous devriez envisager d'utiliser le regroupement des connexions ou des délais d'expiration pour les connexions inactives afin de limiter le nombre de connexions établies avec le cluster, ou pour Redis, envisager d'ajuster la valeur de tcp-keepalive sur votre cluster afin de détecter et d'éliminer les pairs potentiellement morts.

Objectif : L'alarme vous aide à identifier le nombre élevé de connexions susceptibles d'avoir un impact sur les performances et la stabilité de votre ElastiCache cluster.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de la plage de connexions acceptable pour votre cluster. Passez en revue la capacité et la charge de travail attendue de votre ElastiCache cluster et analysez le nombre historique de connexions lors d'une utilisation normale pour établir une base de référence, puis sélectionnez un seuil en conséquence. N'oubliez pas que chaque nœud peut prendre en charge jusqu'à 65 000 connexions simultanées.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_THRESHOLD

DatabaseMemoryUsagePercentage

Dimensions : CacheClusterId

Description de l'alarme : cette alarme vous permet de surveiller le taux d'utilisation de la mémoire de votre cluster. Lorsque votre `DatabaseMemoryUsagePercentage` atteint 100 %, la politique Redis maxmemory est déclenchée et des expulsions peuvent avoir lieu en fonction de la politique sélectionnée. Si aucun objet du cache ne correspond à la politique d'expulsion, les opérations d'écriture échouent. Certaines charges de travail prévoient ou dépendent d'expulsions, mais dans le cas contraire, vous devrez augmenter la capacité de mémoire de votre cluster. Vous pouvez monter votre cluster en puissance en ajoutant d'autres nœuds primaires, ou l'augmenter en utilisant un type de nœud plus large. Reportez-vous à la section Scaling ElastiCache for Redis clusters pour plus de détails.

Objectif : cette alarme est utilisée pour détecter un taux d'utilisation élevé de la mémoire de votre cluster afin d'éviter les défaillances lors de l'écriture sur votre cluster. Il est utile de savoir à quel moment vous devrez augmenter votre cluster si votre application ne prévoit pas d'expulsions.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : en fonction des besoins en mémoire de votre application et de la capacité de mémoire de votre ElastiCache cluster, vous devez définir le seuil sur le pourcentage qui reflète le niveau critique d'utilisation de la mémoire du cluster. Vous pouvez utiliser les données historiques d'utilisation de la mémoire comme référence pour un seuil d'utilisation de mémoire acceptable.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

EngineCPUUtilization

Dimensions : CacheClusterId

Description de l'alarme : Cette alarme permet de surveiller l'utilisation du processeur d'un thread du moteur Redis au sein de l' ElastiCache instance. Les causes courantes d'un processeur surchargé sont les suivantes : des commandes à exécution longue qui consomment beaucoup de processeur, un nombre élevé de requêtes, une augmentation du nombre de nouvelles demandes de connexion client en peu de temps et des expulsions élevées lorsque le cache ne dispose pas de suffisamment de mémoire pour contenir de nouvelles données. Vous devriez envisager de dimensionner ElastiCache les clusters Redis en ajoutant plus de nœuds ou en augmentant le type d'instance.

Intention : cette alarme est utilisée pour détecter un taux d'utilisation élevé du processeur par un thread du moteur Redis. C'est utile si vous souhaitez surveiller l'utilisation du processeur par le moteur de base de données lui-même.

Statistique : moyenne

Seuil recommandé : 90,0

Justification du seuil : définissez le seuil sur un pourcentage qui reflète le niveau d'utilisation critique du processeur pour votre application. Vous pouvez comparer votre cluster à l'aide de votre application et de la charge de travail attendue pour corréler EngineCPUUtilization et les performances comme référence, puis définir le seuil en conséquence. Dans la plupart des cas, vous pouvez définir le seuil à environ 90 % de la capacité de votre processeur.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ReplicationLag

Dimensions : CacheClusterId

Description de l'alarme : Cette alarme permet de surveiller l'état de réplication de votre ElastiCache cluster. Un retard de réplication élevé signifie que le nœud primaire ou le réplica ne peuvent pas suivre le rythme de la réplication. Si votre activité d'écriture est trop élevée, envisagez de redimensionner votre cluster en ajoutant des nœuds primaires ou en utilisant un type de nœud plus important. Reportez-vous à la section Scaling ElastiCache for Redis clusters

pour plus de détails. Si vos réplicas en lecture sont surchargées par le nombre de requêtes de lecture, envisagez d'en ajouter d'autres.

Intention : cette alarme est utilisée pour détecter un délai entre les mises à jour des données sur le nœud primaire et leur synchronisation avec le nœud de réplica. Cela permet de garantir la cohérence des données d'un nœud de cluster de réplica en lecture.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction des exigences de votre application et de l'impact potentiel du retard de réplication. Vous devez tenir compte des taux d'écriture attendus de votre application et des conditions du réseau pour déterminer le délai de réplication acceptable.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon EC2 (**`AWS/ElasticGPUs`**)

GPU ConnectivityCheckFailed

Dimensions :InstanceId, eGPUID

Description de l'alarme : cette alarme permet de détecter les défaillances de connexion entre l'instance et l'accélérateur Elastic Graphics. Elastic Graphics utilise le réseau de l'instance pour envoyer des commandes OpenGL à une carte graphique attachée à distance. Par ailleurs, un bureau exécutant une application OpenGL avec un accélérateur Elastic Graphics est généralement accessible à l'aide d'une technologie d'accès à distance. Il est important de distinguer les problèmes de performance liés au rendu OpenGL ou à la technologie d'accès distant au bureau. Pour en savoir plus sur le problème, veuillez consulter Examiner les problèmes de performance des applications.

Intention : cette alarme est utilisée pour détecter des problèmes de connectivité entre l'instance et l'accélérateur Elastic Graphics.

Statistique : maximum

Seuil recommandé : 0,0

Justification du seuil : la valeur du seuil de 1 indique que la connectivité a échoué.

Période : 300

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

GPU HealthCheckFailed

Dimensions :InstanceId, eGPUID

Description de l'alarme : cette alarme vous permet de savoir si l'état de l'accélérateur graphique Elastic est défectueux. Si l'accélérateur n'est pas en bon état, veuillez consulter les étapes de dépannage dans Résoudre les problèmes de statut Non sain.

Intention : cette alarme est utilisée pour détecter si l'accélérateur Elastic Graphics n'est pas sain.

Statistique : maximum

Seuil recommandé : 0,0

Justification du seuil : la valeur de seuil de 1 indique un échec de la vérification de l'état.

Période : 300

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon ECS

CPUReservation

Dimensions : ClusterName

Description de l'alarme : cette alarme vous aide à détecter une réserve de processeurs élevée du cluster ECS. Une réserve de processeur élevée peut indiquer que le cluster n'a plus de processeurs enregistrés pour la tâche. Pour résoudre le problème, vous pouvez ajouter de la capacité, mettre à l'échelle le cluster ou configurer l'autoscaling.

Intention : l'alarme est utilisée pour détecter si le nombre total d'unités de processeur réservées par les tâches du cluster atteint le nombre total d'unités de processeur enregistrées pour le cluster. Cela vous permet de savoir quand augmenter le cluster. Le fait d'atteindre le nombre total d'unités de processeur pour le cluster peut entraîner un manque d'unités de processeur pour les tâches. Si le dimensionnement géré par les fournisseurs de capacité EC2 est activé ou si vous avez associé Fargate aux fournisseurs de capacité, cette alarme n'est pas recommandée.

Statistique : moyenne

Seuil recommandé : 90,0

Justification du seuil : définissez le seuil de réserve de processeur à 90 %. Vous pouvez également choisir une valeur inférieure en fonction des caractéristiques du cluster.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

CPUUtilization

Dimensions :ClusterName, ServiceName

Description de l'alarme : cette alarme vous aide à détecter un taux d'utilisation élevé du processeur par le service ECS. Si aucun déploiement ECS n'est en cours, une utilisation maximale du processeur peut indiquer un goulot d'étranglement des ressources ou des problèmes de performances des applications. Pour résoudre le problème, vous pouvez augmenter la limite du processeur.

Objectif : cette alarme est utilisée pour détecter un taux d'utilisation élevé du processeur par le service ECS. Un taux d'utilisation élevé et constant du processeur peut indiquer un goulot d'étranglement des ressources ou des problèmes de performance des applications.

Statistique : moyenne

Seuil recommandé : 90,0

Justification du seuil : les métriques de service relatives à l'utilisation du processeur peuvent dépasser 100 % d'utilisation. Toutefois, nous vous recommandons de surveiller la métrique d'utilisation élevée du processeur pour éviter d'affecter les autres services. Réglez le seuil à environ 90 à 95 %. Nous vous recommandons de mettre à jour vos définitions de tâches afin de refléter l'utilisation réelle dans le but d'éviter de futurs problèmes avec d'autres services.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

MemoryReservation

Dimensions : ClusterName

Description de l'alarme : cette alarme vous aide à détecter une réserve de mémoire élevée du cluster ECS. Une réserve de mémoire élevée peut indiquer un goulot d'étranglement des ressources pour le cluster. Pour résoudre le problème, analysez les performances de la tâche de service afin de déterminer si l'utilisation de la mémoire peut être optimisée. Vous pouvez également enregistrer plus de mémoire ou configurer l'autoscaling.

Objectif : l'alarme est utilisée pour détecter si le nombre total d'unités de mémoire réservées par les tâches du cluster atteint le nombre total d'unités de mémoire enregistrées pour le cluster. Cela vous permet de savoir quand augmenter le cluster. Si vous atteignez le nombre total d'unités de mémoire pour le cluster, celui-ci peut être incapable de lancer de nouvelles tâches. Si le dimensionnement géré par les fournisseurs de capacité EC2 est activé ou si vous avez associé Fargate à des fournisseurs de capacité, cette alarme n'est pas recommandée.

Statistique : moyenne

Seuil recommandé : 90,0

Justification du seuil : définissez le seuil de réserve du mémoire à 90 %. Vous pouvez l'ajuster à une valeur inférieure en fonction des caractéristiques du cluster.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

HTTPCode_Target_5XX_Count

Dimensions :ClusterName, ServiceName

Description de l'alarme : cette alarme vous aide à détecter un nombre élevé d'erreurs côté serveur pour le service ECS. Cela peut indiquer que des erreurs empêchent le serveur de répondre aux requêtes. Pour résoudre le problème, consultez les journaux de vos applications.

Objectif : cette alarme est utilisée pour détecter un nombre élevé d'erreurs côté serveur pour le service ECS.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : calculez la valeur d'environ 5 % de votre trafic moyen et utilisez cette valeur comme point de départ pour le seuil. Vous pouvez trouver le trafic moyen à l'aide de la métrique `RequestCount`. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence. Les erreurs 5xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

TargetResponseTime

Dimensions :ClusterName, ServiceName

Description de l'alarme : cette alarme vous aide à détecter un temps de réponse cible élevé pour les demandes de service ECS. Cela peut indiquer que certains problèmes empêchent le service

de traiter les demandes à temps. Pour résoudre le problème, vérifiez la métrique CPUUtilization pour voir si le service manque de processeur, ou vérifiez le taux d'utilisation du processeur des autres services en aval dont dépend votre service.

Intention : cette alarme est utilisée pour détecter un temps de réponse cible élevé pour les demandes de service ECS.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de votre cas d'utilisation. Passez en revue la criticité et les exigences du temps de réponse cible du service et analysez le comportement historique de cette métrique afin de déterminer des seuils raisonnables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon ECS avec Container Insights

EphemeralStorageUtilized

Dimensions :ClusterName, ServiceName

Description de l’alarme : cette alarme vous aide à détecter le stockage éphémère élevé utilisé par le cluster Fargate. Si le stockage éphémère est constamment élevé, vous pouvez vérifier l’utilisation du stockage éphémère et augmenter ce dernier.

Intention : cette alarme est utilisée pour détecter une utilisation élevée du stockage éphémère du cluster Fargate. L’utilisation constante d’un stockage éphémère élevé peut indiquer que le disque est plein et peut entraîner une défaillance du conteneur.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil à environ 90 % de la taille du stockage éphémère. Vous pouvez ajuster cette valeur en fonction de votre utilisation acceptable du stockage éphémère du cluster Fargate. Pour certains systèmes, un stockage éphémère constamment élevé peut être normal, tandis que pour d'autres, cela peut entraîner une défaillance du conteneur.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

RunningTaskCount

Dimensions :ClusterName, ServiceName

Description de l'alarme : cette alarme vous aide à détecter un faible nombre de tâches en cours d'exécution du service ECS. Si le nombre de tâches en cours d'exécution est trop faible, cela peut indiquer que l'application ne peut pas gérer la charge du service et cela peut entraîner des problèmes de performances. Si aucune tâche n'est en cours d'exécution, il est possible que le service Amazon ECS ne soit pas disponible ou qu'il s'agisse de problèmes de déploiement.

Intention : cette alarme est utilisée pour détecter si le nombre de tâches en cours d'exécution est trop faible. Un faible nombre de tâches en cours d'exécution constant peut indiquer le déploiement du service ECS ou des problèmes de performance.

Statistique : moyenne

Seuil recommandé : 0,0

Justification du seuil : vous pouvez ajuster le seuil en fonction du nombre minimal de tâches en cours d'exécution du service ECS. Si le nombre de tâches en cours est égal à 0, le service Amazon ECS ne sera pas disponible.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_OR_EQUAL_TO_THRESHOLD

instance_filesystem_utilization

InstanceIdDimensioni : ContainerInstanceId, ClusterName

Description de l’alarme : cette alarme vous aide à détecter un taux d’utilisation élevé du système de fichiers du cluster Amazon ECS. Si le taux d’utilisation du système de fichiers est constamment élevé, vérifiez l’utilisation du disque.

Intention : cette alarme est utilisée pour détecter un taux d’utilisation élevé du système de fichiers du cluster Amazon ECS. Un taux d’utilisation élevé et constant du système de fichiers peut indiquer un goulot d’étranglement des ressources ou des problèmes de performance des applications, et peut empêcher l’exécution de nouvelles tâches.

Statistique : moyenne

Seuil recommandé : 90,0

Justification du seuil : vous pouvez généralement définir le seuil d’utilisation du processeur entre 90 et 95 %. Vous pouvez ajuster cette valeur en fonction du niveau de capacité acceptable du système de fichiers du cluster Amazon ECS. Pour certains systèmes, un taux d’utilisation élevé et constant du système de fichiers peut être normal et ne pas être le signe d’un problème, tandis que pour d’autres, cela peut être source de préoccupation, entraîner des problèmes de performances et empêcher l’exécution de nouvelles tâches.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon EFS

PercentIOLimit

Dimensions : FileSystemId

Description de l'alarme : cette alarme permet de garantir que la charge de travail reste dans les limites d'E/S disponibles pour le système de fichiers. Si la métrique atteint régulièrement sa limite d'E/S, envisagez de déplacer l'application vers un système de fichiers utilisant les performances

d'E/S maximales comme mode. Pour résoudre les problèmes, vérifiez les clients connectés au système de fichiers et les applications des clients qui limitent le système de fichiers.

Intention : cette alarme est utilisée pour détecter à quel point le système de fichiers est proche d'atteindre la limite d'E/S du mode de performance à usage général. Un pourcentage d'E/S élevé et persistant peut indiquer que le système de fichiers ne peut pas s'adapter suffisamment aux requêtes d'E/S et qu'il peut constituer un goulot d'étranglement pour les applications qui utilisent le système de fichiers.

Statistique : moyenne

Seuil recommandé : 100,0

Justification du seuil : lorsque le système de fichiers atteint sa limite d'E/S, il risque de répondre plus lentement aux requêtes de lecture et d'écriture. Par conséquent, il est recommandé de surveiller la métrique afin d'éviter toute incidence sur les applications qui utilisent le système de fichiers. Le seuil peut être fixé aux alentours de 100 %. Toutefois, cette valeur peut être ajustée à une valeur inférieure en fonction des caractéristiques du système de fichiers.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

BurstCreditBalance

Dimensions : FileSystemId

Description de l'alarme : cette alarme permet de garantir qu'un solde de crédit de débordement est disponible pour l'utilisation du système de fichiers. Lorsqu'aucun crédit de débordement n'est disponible, l'accès des applications au système de fichiers est limité en raison du faible débit. Si la métrique tombe régulièrement à 0, envisagez de changer le mode de débit en mode de débit Elastic ou alloué.

Objectif : cette alarme est utilisée pour détecter un faible solde de crédit de débordement du système de fichiers. Un faible solde de crédit de débordement persistant peut être un indicateur du ralentissement du débit et de l'augmentation de la latence des E/S.

Statistique : moyenne

Seuil recommandé : 0,0

Justification du seuil : lorsque le système de fichiers n'a plus de crédits en rafale et même si le débit de référence est inférieur, EFS continue de fournir un débit mesuré de 1 MiBps à tous les systèmes de fichiers. Cependant, il est recommandé de surveiller le faible solde de crédit de débordement de la métrique afin d'éviter que le système de fichiers ne constitue un goulot d'étranglement des ressources pour les applications. Le seuil peut être défini autour de 0 octet.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : LESS_THAN_OR_EQUAL_TO_THRESHOLD

Amazon EKS avec Container Insights

node_cpu_utilization

Dimensions : ClusterName

Description de l'alarme : cette alarme permet de détecter une utilisation élevée du processeur dans les composants master du cluster EKS. Si le taux d'utilisation est constamment élevé, cela peut indiquer la nécessité de remplacer vos composants master par des instances dotées d'un processeur plus puissant ou de mettre à l'échelle le système horizontalement.

Objectif : cette alarme permet de surveiller l'utilisation du processeur par les composants master du cluster EKS afin que les performances du système ne se dégradent pas.

Statistique : maximum

Seuil recommandé : 80,0

Justification du seuil : il est recommandé de définir le seuil à un niveau inférieur ou égal à 80 % afin de laisser suffisamment de temps pour corriger le problème avant que le système ne commence à en ressentir l'impact.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

node_filesystem_utilization

Dimensions : ClusterName

Description de l'alarme : cette alarme permet de détecter un taux d'utilisation élevé du système de fichiers dans les composants master du cluster EKS. Si le taux d'utilisation est constamment élevé, vous devrez peut-être mettre à jour vos composants master pour augmenter le volume de disque, ou vous devrez peut-être effectuer une mise à l'échelle horizontale.

Objectif : cette alarme permet de surveiller le taux d'utilisation du système de fichiers par les composants master du cluster EKS. Si le taux d'utilisation atteint 100 %, cela peut entraîner une défaillance de l'application, des blocages d'E/S sur le disque, l'expulsion du pod ou un arrêt complet du nœud.

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : si la pression sur le disque est suffisante (ce qui signifie que le disque est plein), les nœuds sont marqués comme non sains et les pods sont expulsés du nœud. Les pods d'un nœud soumis à une pression sur le disque sont expulsés lorsque le système de fichiers disponible est inférieur aux seuils d'expulsion définis sur le kubelet. Définissez le seuil d'alarme afin de disposer de suffisamment de temps pour réagir avant que le nœud ne soit expulsé du cluster.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

node_memory_utilization

Dimensions : ClusterName

Description de l'alarme : cette alarme permet de détecter un taux d'utilisation élevé de la mémoire dans les composants master du cluster EKS. Si le taux d'utilisation est constamment élevé, cela

peut indiquer la nécessité de mettre à l'échelle le nombre de réplicas de pods ou d'optimiser votre application.

Objectif : cette alarme permet de surveiller le taux d'utilisation de la mémoire par les composants master du cluster EKS afin que les performances du système ne se dégradent pas.

Statistique : maximum

Seuil recommandé : 80,0

Justification du seuil : il est recommandé de définir le seuil à un niveau inférieur ou égal à 80 % afin de disposer de suffisamment de temps pour corriger le problème avant que le système ne commence à en ressentir l'impact.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

pod_cpu_utilization_over_pod_limit

Dimensions : espace ClusterName de noms, service

Description de l'alarme : cette alarme permet de détecter un taux d'utilisation élevé du processeur dans les pods du cluster EKS. Si le taux d'utilisation est constamment élevé, cela peut indiquer la nécessité d'augmenter la limite du processeur pour le pod concerné.

Objectif : cette alarme permet de surveiller le taux d'utilisation du processeur par les pods appartenant à un service Kubernetes dans le cluster EKS, afin que vous puissiez rapidement identifier si le pod d'un service consomme plus de processeur que prévu.

Statistique : maximum

Seuil recommandé : 80,0

Justification du seuil : il est recommandé de définir le seuil à un niveau inférieur ou égal à 80 % afin de disposer de suffisamment de temps pour corriger le problème avant que le système ne commence à en ressentir l'impact.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

pod_memory_utilization_over_pod_limit

Dimensions : espace ClusterName de noms, service

Description de l'alarme : cette alarme permet de détecter un taux d'utilisation élevé de la mémoire dans les pods du cluster EKS. Si le taux d'utilisation est constamment élevé, cela peut indiquer la nécessité d'augmenter la limite de mémoire pour le pod concerné.

Objectif : cette alarme permet de surveiller le taux d'utilisation de la mémoire par les pods du cluster EKS afin que les performances du système ne se dégradent pas.

Statistique : maximum

Seuil recommandé : 80,0

Justification du seuil : il est recommandé de définir le seuil à un niveau inférieur ou égal à 80 % afin de disposer de suffisamment de temps pour corriger le problème avant que le système ne commence à en ressentir l'impact.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon Kinesis Data Streams

GetRecords.IteratorAgeMilliseconds

Dimensions : StreamName

Description de l'alarme : cette alarme peut détecter si l'ancienneté maximale de l'itérateur est trop élevée. Pour les applications de traitement de données en temps réel, configurez la conservation des données en fonction de la tolérance du délai. Il s'agit généralement de quelques minutes. Pour les applications qui traitent des données historiques, utilisez cette métrique pour surveiller la

vitesse de rattrapage. Une solution rapide pour arrêter la perte de données consiste à augmenter la période de conservation pendant que vous résolvez le problème. Vous pouvez également augmenter le nombre de travailleurs qui traitent les dossiers dans votre application client. Les causes les plus courantes de l'augmentation progressive de l'ancienneté de l'itérateur sont l'insuffisance des ressources physiques ou une logique de traitement des enregistrements qui ne s'est pas adaptée à l'augmentation du débit des flux. Veuillez consulter ce lien pour plus de détails.

Intention : cette alarme est utilisée pour détecter si les données de votre flux vont expirer parce qu'elles ont été conservées trop longtemps ou parce que le traitement des enregistrements est trop lent. Il vous permet d'éviter la perte de données après avoir atteint 100 % de la durée de conservation du flux.

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de la période de rétention du flux et de la tolérance du délai de traitement des enregistrements. Passez en revue vos exigences et analysez les tendances historiques, puis définissez le seuil au nombre de millisecondes qui représente un délai de traitement critique. Si l'ancienneté de l'itérateur dépasse 50 % de la période de conservation (par défaut 24 heures, configurable jusqu'à 365 jours), il y a un risque de perte de données suite à l'expiration des enregistrements. Vous pouvez surveiller la métrique pour vous assurer qu'aucune de vos partitions n'approche cette limite.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

GetRecords. Succès

Dimensions : StreamName

Description de l'alarme : cette métrique augmente chaque fois que vos clients lisent avec succès les données de votre flux. `GetRecords` ne renvoie aucune donnée lorsqu'il renvoie une exception. L'exception la plus courante est `ProvisionedThroughputExceededException`

parce que le taux de requête pour le flux est trop élevé, ou parce que le débit disponible est déjà servi pour la seconde donnée. Réduisez la fréquence ou la taille de vos requêtes. Pour plus d'informations, veuillez consulter Quotas et limites de flux dans le guide du développeur Amazon Kinesis Data Streams, et Comportement des nouvelles tentatives et backoff exponentiel dans AWS (langue française non garantie).

Intention : cette alarme peut détecter si la récupération des enregistrements du flux par les consommateurs échoue. En réglant une alarme sur cette métrique, vous pouvez détecter de manière proactive tout problème lié à la consommation de données, tel qu'une augmentation du taux d'erreur ou une diminution du nombre de récupérations réussies. Cela vous permet de prendre des mesures opportunes pour résoudre les problèmes potentiels et maintenir un pipeline de traitement des données fluide.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : selon l'importance de récupérer les enregistrements du flux, définissez le seuil en fonction de la tolérance de votre application pour les enregistrements ayant échoué. Le seuil doit être le pourcentage correspondant d'opérations réussies. Vous pouvez utiliser les données GetRecords métriques historiques comme référence pour le taux d'échec acceptable. Vous devez également envisager de nouvelles tentatives lorsque vous définissez le seuil, car les enregistrements ayant échoué peuvent être réessayés. Cela permet d'éviter que des pics transitoires ne déclenchent des alertes inutiles.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_THRESHOLD

PutRecord. Succès

Dimensions : StreamName

Description de l'alarme : cette alarme détecte lorsque le nombre d'opérations `PutRecord` ayant échoué dépasse le seuil. Examinez les journaux des producteurs de données pour trouver les causes profondes des défaillances. La raison la plus courante est l'insuffisance du débit provisionné sur la partition à l'origine du `ProvisionedThroughputExceededException`. Cela

se produit parce que le taux de requêtes pour le flux est trop élevé ou que le débit que l'on tente d'ingérer dans la partition est trop élevé. Réduisez la fréquence ou la taille de vos requêtes. Pour plus d'informations, consultez Streams Limits and Error Retries et Exponential Backoff in. AWS

Intention : cette alarme peut détecter si l'ingestion d'enregistrements dans le flux échoue. Elle vous aide à identifier les problèmes liés à l'écriture de données dans le flux. En réglant une alarme sur cette métrique, vous pouvez détecter de manière proactive les problèmes rencontrés par les producteurs lors de la publication de données dans le flux, tels que l'augmentation du taux d'erreur ou la diminution du nombre d'enregistrements publiés avec succès. Cela vous permet de prendre des mesures en temps opportun pour résoudre les problèmes potentiels et maintenir un processus d'ingestion de données fiable.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : selon l'importance du traitement et de l'ingestion de données pour votre service, définissez le seuil en fonction de la tolérance de votre application à l'égard des enregistrements manquants. Le seuil doit être le pourcentage correspondant d'opérations réussies. Vous pouvez utiliser les données PutRecord métriques historiques comme référence pour le taux d'échec acceptable. Vous devez également envisager de nouvelles tentatives lorsque vous définissez le seuil, car les enregistrements ayant échoué peuvent être réessayés.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_THRESHOLD

PutRecords.FailedRecords

Dimensions : StreamName

Description de l'alarme : cette alarme détecte lorsque le nombre de `PutRecords` ayant échoué dépasse le seuil. Kinesis Data Streams tente de traiter tous les enregistrements de chaque requête `PutRecords`, mais un seul échec d'enregistrement n'arrête pas le traitement des enregistrements suivants. La principale raison de ces défaillances est le dépassement du débit d'un flux ou d'une partition individuelle. Les causes les plus courantes sont les pics de trafic et les latences du réseau qui font que les enregistrements arrivent au flux de manière inégale. Vous

devez détecter les enregistrements traités sans succès et les retenter dans un appel ultérieur. Reportez-vous à la section Gestion des défaillances lors de l'utilisation PutRecords pour plus de détails.

Objectif : cette alarme peut détecter des défaillances constantes lors de l'utilisation d'une opération par lots pour ajouter des enregistrements à votre flux. En réglant une alarme sur cette métrique, vous pouvez détecter de manière proactive une augmentation du nombre d'enregistrements échoués, ce qui vous permet de prendre des mesures opportunes pour résoudre les problèmes sous-jacents et garantir un processus d'ingestion de données fluide et fiable.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du nombre d'enregistrements ayant échoué en fonction de la tolérance de l'application pour les enregistrements échoués. Vous pouvez utiliser les données historiques comme référence pour la valeur d'échec acceptable. Vous devez également envisager de nouvelles tentatives lors de la définition du seuil, car les enregistrements ayant échoué peuvent être réessayés lors des appels suivants PutRecords .

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ReadProvisionedThroughputExceeded

Dimensions : StreamName

Description de l'alarme : l'alarme suit le nombre d'enregistrements qui entraînent une limitation de la capacité de débit de lecture. Si vous constatez que vous êtes constamment limité, vous devriez envisager d'ajouter des partitions supplémentaires à votre flux afin d'augmenter le débit de lecture alloué. Si plusieurs applications consommateur s'exécutent sur le flux et qu'elles partagent la même limite `GetRecords`, nous vous recommandons d'enregistrer de nouvelles applications consommateur via Enhanced Fan-Out. Si l'ajout de partitions supplémentaires ne réduit pas le nombre de limitations, il se peut qu'une partition « chaude » soit lue plus souvent que les autres partitions. Activez la surveillance améliorée, trouvez la partition « chaude » et divisez-la.

Intention : cette alarme peut détecter si les consommateurs sont limités lorsqu'ils dépassent le débit de lecture que vous avez prévu (déterminé par le nombre de partitions que vous possédez). Dans ce cas, vous ne pourrez pas lire à partir du flux, et le flux pourra commencer à être sauvegardé.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : les requêtes limitées peuvent généralement être réessayées. Par conséquent, le fait de fixer le seuil à zéro rend l'alarme trop sensible. Cependant, une limitation persistante peut avoir un impact sur la lecture du flux et devrait déclencher l'alarme. Définissez le seuil sur un pourcentage en fonction des requêtes limitées pour l'application et réessayez les configurations.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

SubscribeToShardEvent.MillisBehindLatest

Dimensions :StreamName, ConsumerName

Description de l'alarme : cette alarme détecte lorsque le délai de traitement des enregistrements dans l'application dépasse le seuil. Des problèmes transitoires tels que l'échec des opérations d'API d'une application en aval peuvent entraîner une augmentation soudaine de la métrique. Vous devriez vérifier s'ils se produisent régulièrement. Cela est souvent dû au fait que le consommateur ne traite pas les enregistrements assez rapidement en raison de ressources physiques insuffisantes ou d'une logique de traitement des enregistrements qui n'a pas été mise à l'échelle en fonction de l'augmentation du débit des flux. Le blocage des appels sur le chemin critique est souvent à l'origine de ralentissements dans le traitement des enregistrements. Vous pouvez augmenter votre parallélisme en augmentant le nombre de partitions. Vous devez également vérifier que les nœuds de traitement sous-jacents disposent de ressources physiques suffisantes pendant les pics de demande.

Intention : cette alarme peut détecter un retard dans l'événement d'abonnement à la partition du flux. Cela indique un retard de traitement et peut aider à identifier les problèmes potentiels liés

aux performances de l'application client ou à l'état général du flux. Lorsque le délai de traitement devient important, vous devez examiner et corriger les éventuels obstacles ou inefficiences des applications destinées aux consommateurs afin de garantir le traitement des données en temps réel et de minimiser les retards dans le traitement des données.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : La valeur de seuil recommandée pour cette alarme dépend fortement du délai que votre application peut tolérer. Passez en revue les exigences de votre application et analysez les tendances historiques, puis sélectionnez un seuil en conséquence. Lorsque l' SubscribeToShard appel aboutit, votre client commence à recevoir SubscribeToShardEvent des événements via la connexion permanente pendant 5 minutes au maximum, après quoi vous devez appeler à SubscribeToShard nouveau pour renouveler l'abonnement si vous souhaitez continuer à recevoir des enregistrements.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

WriteProvisionedThroughputExceeded

Dimensions : StreamName

Description de l'alarme : cette alarme détecte le moment où le nombre d'enregistrements entraînant une limitation de la capacité du débit d'écriture a atteint le seuil. Lorsque vos producteurs dépassent le débit d'écriture prévu (déterminé par le nombre de partitions dont vous disposez), ils sont limités et vous ne pouvez pas ajouter d'enregistrements dans le flux. Pour faire face à une limitation persistante, vous devriez envisager d'ajouter des partitions à votre flux. Cela augmente le débit d'écriture que vous avez alloué et empêche toute limitation future. Vous devez également prendre en compte le choix de la clé de partition lors de l'ingestion d'enregistrements. Une clé de partition aléatoire est préférable, car elle répartit les enregistrements de manière uniforme sur les partitions du flux, dans la mesure du possible.

Intention : cette alarme peut détecter si vos producteurs sont rejetés pour avoir écrit des enregistrements en raison de la limitation du flux ou de la partition. Si votre flux est en mode

provisionné, le réglage de cette alarme vous permet de prendre des mesures proactives lorsque le flux de données atteint ses limites, ce qui vous permet d'optimiser la capacité allouée ou de prendre les mesures de dimensionnement appropriées pour éviter les pertes de données et garantir un traitement fluide des données.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : les requêtes limitées peuvent généralement être réessayées. Par conséquent, le fait de définir le seuil à zéro rend l'alarme trop sensible. Cependant, une limitation persistante peut avoir un impact sur l'écriture dans le flux, et vous devez définir le seuil d'alarme pour le détecter. Définissez le seuil sur un pourcentage en fonction des requêtes limitées pour l'application et réessayez les configurations.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Lambda

ClaimedAccountConcurrency

Dimensions : Aucune

Description de l'alarme : Cette alarme permet de vérifier si la simultanéité de vos fonctions Lambda se rapproche de la limite de simultanéité de votre compte au niveau de la région. Une fonction commence à être limitée si elle atteint la limite de simultanéité. Vous pouvez prendre les mesures suivantes pour éviter la limitation.

1. Demandez une augmentation de la simultanéité dans cette région.
2. Identifiez et réduisez toute simultanéité réservée inutilisée ou provisionnée.
3. Identifiez les problèmes de performance des fonctions afin d'améliorer la vitesse de traitement et donc le débit.
4. Augmentez la taille du lot des fonctions afin que davantage de messages soient traités à chaque appel de fonction.

Objectif : Cette alarme peut détecter de manière proactive si la simultanéité de vos fonctions Lambda se rapproche du quota de simultanéité régional de votre compte, afin que vous puissiez agir en conséquence. Les fonctions sont limitées si le quota de `ClaimedAccountConcurrency` simultanéité du compte atteint au niveau de la région. Si vous utilisez la simultanéité réservée (RC) ou la concurrence provisionnée (PC), cette alarme vous donne une meilleure visibilité sur l'utilisation de la simultanéité qu'une alarme activée. `ConcurrentExecutions`

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : vous devez calculer la valeur d'environ 90 % du quota de simultanéité défini pour le compte dans la région et utiliser le résultat comme valeur de seuil. Par défaut, votre compte dispose d'un quota de simultanéité de 1 000 pour toutes les fonctions d'une région. Cependant, vous devez vérifier le quota de votre compte depuis le tableau de bord des Services Quotas.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Erreurs

Dimensions : FunctionName

Description de l'alarme : cette alarme détecte un nombre élevé d'erreurs. Les erreurs de fonction incluent les exceptions levées par votre code et par l'environnement d'exécution Lambda. Vous pouvez consulter les journaux relatifs à la fonction pour diagnostiquer le problème.

Objectif : l'alarme permet de détecter un nombre élevé d'erreurs lors des invocations de fonctions.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil sur un nombre supérieur à zéro. La valeur exacte peut dépendre de la tolérance aux erreurs de votre application. Comprenez le caractère critique

des invocations gérées par la fonction. Pour certaines applications, toute erreur peut être inacceptable, tandis que d'autres applications peuvent autoriser une certaine marge d'erreur.

Période : 60

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Throttles

Dimensions : FunctionName

Description de l'alarme : cette alarme détecte un nombre élevé de requêtes d'invocation limitées. La limitation se produit lorsqu'aucune simultanéité n'est disponible pour une augmentation. Plusieurs approches permettent de résoudre ce problème. 1) Demandez une augmentation de la simultanéité auprès du AWS Support de cette région. 2) Identifier les problèmes de performance de la fonction afin d'améliorer la vitesse de traitement et donc le débit. 3) Augmenter la taille du lot de la fonction, de sorte que davantage de messages soient traités à chaque invocation de fonction.

Intention : l'alarme permet de détecter un nombre élevé de requêtes d'invocation limitées pour une fonction Lambda. Il est important de savoir si les requêtes sont constamment rejetées en raison de la limitation et si vous devez améliorer les performances de la fonction Lambda ou augmenter la capacité de simultanéité pour éviter une limitation persistante.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil sur un nombre supérieur à zéro. La valeur exacte du seuil peut dépendre de la tolérance de l'application. Définissez le seuil en fonction de son utilisation et des exigences de mise à l'échelle de la fonction.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

Durée

Dimensions : FunctionName

Description de l'alarme : cette alarme détecte les longues durées de traitement d'un événement par une fonction Lambda. Les longues durées peuvent être dues à des modifications du code de la fonction qui allongent son exécution ou à un allongement de l'exécution des dépendances de la fonction.

Objectif : cette alarme peut détecter une longue durée d'exécution d'une fonction Lambda. Une durée d'exécution élevée indique qu'une fonction met plus de temps à être invoquée et peut également avoir un impact sur la capacité d'invocation simultanée si Lambda gère un plus grand nombre d'événements. Il est essentiel de savoir si le temps d'exécution de la fonction Lambda est constamment plus long que prévu.

Statistique : p90

Seuil recommandé : dépend de votre situation

Justification du seuil : le seuil de durée dépend de votre application et de vos charges de travail, ainsi que de vos exigences en matière de performances. Pour les exigences de haute performance, fixez le seuil à un délai plus court pour voir si la fonction répond aux attentes. Vous pouvez également analyser les données historiques pour les métriques de durée afin de déterminer si le temps nécessaire correspond aux attentes de performance de la fonction, puis définir le seuil sur une durée supérieure à la moyenne historique. Assurez-vous de définir un seuil inférieur au délai d'expiration de fonction configuré.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ConcurrentExecutions

Dimensions : FunctionName

Description de l'alarme : cette alarme permet de vérifier si la simultanéité de la fonction se rapproche de la limite de simultanéité de votre compte au niveau de la région. Une fonction

commence à être limitée si elle atteint la limite de simultanéité. Vous pouvez prendre les mesures suivantes pour éviter la limitation.

1. Demandez une augmentation de la simultanéité dans cette région.
2. Identifiez les problèmes de performance des fonctions afin d'améliorer la vitesse de traitement et donc le débit.
3. Augmentez la taille du lot des fonctions afin que davantage de messages soient traités à chaque appel de fonction.

Pour obtenir une meilleure visibilité sur la simultanéité réservée et l'utilisation de la simultanéité provisionnée, définissez plutôt une alarme sur la nouvelle métrique. `ClaimedAccountConcurrency`

Objectif : cette alarme peut détecter de manière proactive si la simultanéité de la fonction se rapproche du quota de simultanéité de votre compte au niveau de la région, afin que vous puissiez agir en conséquence. Une fonction est limitée si elle atteint le quota de simultanéité du compte au niveau de la région.

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : fixez le seuil à environ 90 % du quota de simultanéité défini pour le compte dans la région. Par défaut, votre compte dispose d'un quota de simultanéité de 1 000 pour toutes les fonctions d'une région. Cependant, vous pouvez vérifier le quota de votre compte, car il peut être augmenté en contactant le AWS support.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Aperçu Lambda

Nous vous recommandons de définir des alarmes conformes aux bonnes pratiques pour les métriques Lambda Insights suivantes.

memory_utilization

Dimensions : function_name

Description de l'alarme : cette alarme est utilisée pour détecter si le taux d'utilisation de la mémoire par une fonction Lambda se rapproche de la limite configurée. Pour résoudre les problèmes, vous pouvez essayer de 1) Optimiser votre code. 2) Dimensionner correctement votre allocation de mémoire en estimant avec précision les besoins en mémoire. Vous pouvez vous référer à Lambda Power Tuning pour cela. 3) Utiliser le regroupement des connexions. Veuillez consulter le billet de blog Using Amazon RDS Proxy with Lambda au sujet du regroupement de connexions pour une base de données RDS. 4) Vous pouvez également envisager de concevoir vos fonctions de manière à éviter de stocker de grandes quantités de données en mémoire entre les invocations.

Intention : cette alarme est utilisée pour détecter si le taux d'utilisation de la mémoire pour la fonction Lambda se rapproche de la limite configurée.

Statistique : moyenne

Seuil suggéré : 90,0

Justification du seuil : définissez le seuil à 90 % pour recevoir une alerte lorsque le taux d'utilisation de la mémoire dépasse 90 % de la mémoire allouée. Vous pouvez l'ajuster à une valeur inférieure si la charge de travail liée à l'utilisation de la mémoire vous préoccupe. Vous pouvez également vérifier les données historiques de cette métrique et définir le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

ComparisonOperator: SUPÉRIEUR AU SEUIL

Amazon VPC (**`AWS/NATGateway`**)

ErrorPortAllocation

Dimensions : NatGatewayId

Description de l'alarme : cette alarme permet de détecter les cas où la passerelle NAT n'est pas en mesure d'allouer des ports à de nouvelles connexions. Pour résoudre ce problème, veuillez consulter Résoudre les erreurs d'allocation de port sur la passerelle NAT.

Intention : cette alarme est utilisée pour détecter si la passerelle NAT n'a pas été en mesure d'allouer un port source.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : si la valeur de ErrorPortAllocation est supérieure à zéro, cela signifie que trop de connexions simultanées vers une seule destination populaire sont ouvertes via NatGateway.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

PacketsDropCount

Dimensions : NatGatewayId

Description de l'alarme : cette alarme permet de détecter quand des paquets sont abandonnés par la passerelle NAT. Cela peut être dû à un problème avec la passerelle NAT. Consultez le tableau de bord de santé du AWS service pour connaître l'état de la passerelle AWS NAT dans votre région. Cela peut vous aider à établir une corrélation entre le problème de réseau et le trafic utilisant la passerelle NAT.

Intention : cette alarme est utilisée pour détecter si des paquets sont abandonnés par la passerelle NAT.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : il convient de calculer la valeur de 0,01 % du trafic total sur la passerelle NAT et d'utiliser ce résultat comme valeur seuil. Utilisez les données historiques du trafic sur la passerelle NAT pour déterminer le seuil.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

AWS Lien privé (**`AWS/PrivateLinkEndpoints`**)

PacketsDropped

Dimensions : VPC Id, VPC Endpoint Id, Endpoint Type, Subnet Id, Service Name

Description de l'alarme : cette alarme permet de détecter si le terminal ou le service du point de terminaison est défectueux en surveillant le nombre de paquets abandonnés par le point de terminaison. Notez que les paquets de plus de 8 500 octets arrivant au point de terminaison d'un VPC sont abandonnés. Pour la résolution de ce problème, veuillez consulter Problèmes de connectivité entre le point de terminaison d'un VPC d'interface et un service de point de terminaison.

Intention : cette alarme est utilisée pour détecter si le point de terminaison ou le service de point de terminaison n'est pas sain.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil en fonction du cas d'utilisation. Si vous souhaitez être informé de l'état non sain d'un point de terminaison ou d'un service de point de terminaison, vous devez fixer un seuil bas afin de pouvoir résoudre le problème avant qu'une perte de données considérable ne se produise. Vous pouvez utiliser les données historiques pour comprendre la tolérance aux paquets abandonnés et définir le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

AWS Lien privé (**`AWS/PrivateLinkServices`**)

RstPacketsSent

Dimensions : Service Id, Load Balancer Arn, Az

Description de l'alarme : cette alarme vous aide à détecter les cibles non saines d'un service de point de terminaison en fonction du nombre de paquets de réinitialisation envoyés aux points de terminaison. Lorsque vous corrigez des erreurs de connexion avec un client de votre service, vous pouvez vérifier si le service réinitialise les connexions avec la RstPacketsSent métrique ou si quelque chose d'autre échoue sur le chemin réseau.

Intention : cette alarme est utilisée pour détecter les cibles non saines d'un service de point de terminaison.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : le seuil dépend du cas d'utilisation. Si votre cas d'utilisation peut tolérer que les cibles ne soient pas saines, vous pouvez définir un seuil élevé. Si le scénario d'utilisation ne tolère pas les cibles non saines, vous pouvez définir le seuil à un niveau très bas.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon RDS

CPUUtilization

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet de surveiller un taux d’utilisation élevé et constant du processeur. Le taux d’utilisation du processeur mesure le temps de non inactivité. Envisagez d’utiliser la Surveillance améliorée ou Performance Insights pour déterminer quel temps d’attente occupe le plus de temps processeur (`guest`, `irq`, `wait`, `nice`, etc.) pour MariaDB, MySQL,

Oracle et PostgreSQL. Évaluez ensuite quelles requêtes consomment le plus de ressources processeur. Si vous ne parvenez pas à ajuster votre charge de travail, envisagez de passer à une classe d'instance de base de données plus importante.

Intention : cette alarme est utilisée pour détecter une utilisation élevée constante du processeur afin d'éviter des temps de réponse et des délais d'expiration très élevés. Si vous souhaitez vérifier la microsaturation de l'utilisation du processeur, vous pouvez définir une durée d'évaluation des alarmes plus courte.

Statistique : moyenne

Seuil recommandé : 90,0

Justification du seuil : les pics aléatoires de consommation du processeur ne nuisent peut-être pas aux performances de la base de données, mais une charge processeur élevée et prolongée peut gêner les requêtes de la base de données à venir. En fonction de la charge de travail globale de la base de données, une charge processeur élevée au niveau de votre instance RDS/Aurora peut dégrader les performances globales.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

DatabaseConnections

Dimensioni : DB InstanceIdentifier

Description de l'alarme : cette alarme détecte un nombre élevé de connexions. Passez en revue les connexions existantes et mettez fin à celles qui sont en état de « veille » ou qui ne sont pas correctement fermées. Envisagez d'utiliser le regroupement de connexions pour limiter le nombre de nouvelles connexions. Vous pouvez également augmenter la taille de l'instance de base de données pour utiliser une classe avec plus de mémoire et donc une valeur par défaut plus élevée pour « max_connections » ou augmenter la valeur « max_connections » dans RDS et Aurora MySQL et PostgreSQL pour la classe actuelle si elle peut supporter votre charge de travail.

Intention : cette alarme est utilisée pour empêcher le rejet de connexions lorsque le nombre maximum de connexions à la base de données est atteint. Cette alarme n'est pas recommandée

si vous changez fréquemment de classe d’instance de base de données, car cela modifie la mémoire et le nombre maximal de connexions par défaut.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : le nombre de connexions autorisées dépend de la taille de votre classe d’instance de base de données et des paramètres spécifiques au moteur de base de données relatifs aux processus/connexions. Vous devez évaluer une valeur comprise entre 90 et 95 % du nombre maximal de connexions pour votre base de données et utiliser ce pourcentage comme valeur seuil.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

% EBS ByteBalance

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet de surveiller un faible pourcentage de crédits de débit restants. Pour résoudre les problèmes, vérifiez les problèmes de latence dans RDS.

Intention : cette métrique indique le pourcentage de crédit de débit restant dans le compartiment de débordement. Un faible pourcentage d’équilibrage des octets peut entraîner des problèmes de goulot d’étranglement au niveau du débit. Cette alarme n’est pas recommandée pour les instances Aurora PostgreSQL.

Statistique : moyenne

Seuil recommandé : 10,0

Justification du seuil : un solde de crédit de débit inférieur à 10 % est considéré comme faible et vous devez définir le seuil en conséquence. Vous pouvez également définir un seuil inférieur si votre application peut tolérer un débit inférieur pour la charge de travail.

Période : 60

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : LESS_THAN_THRESHOLD

EBSIOBalance%

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet de surveiller le faible pourcentage de crédits IOPS restants. Pour résoudre les problèmes, veuillez consulter la rubrique Problèmes de latence dans RDS.

Intention : cette métrique indique le pourcentage de crédits d’E/S restant dans le compartiment de débordement. Un faible pourcentage de solde d’IOPS peut entraîner des problèmes de blocage des IOPS. Cette alarme n’est pas recommandée pour les instances Aurora.

Statistique : moyenne

Seuil recommandé : 10,0

Justification du seuil : un solde de crédits IOPS inférieur à 10 % est considéré comme faible et vous pouvez définir le seuil en conséquence. Vous pouvez également définir un seuil inférieur si votre application peut tolérer un taux d’IOPS inférieur pour la charge de travail.

Période : 60

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : LESS_THAN_THRESHOLD

FreeableMemory

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet de surveiller une faible quantité de mémoire libérable, ce qui peut signifier qu’il y a un pic dans les connexions à la base de données ou que votre instance est soumise à une forte sollicitation de mémoire. Vérifiez la pression de la mémoire en surveillant les CloudWatch métriques pour `SwapUsage` « en plus de`FreeableMemory`. Si la consommation de mémoire de l’instance est fréquemment trop élevée, c’est le signe que vous

devriez vérifier votre charge de travail ou mettre à niveau votre classe d'instance. Pour l'instance en lecture de la base de données Aurora, envisagez d'ajouter d'autres instances en lecture de la base de données au cluster. Pour plus d'informations sur le dépannage d'Aurora, veuillez consulter la rubrique Problèmes liés à la mémoire libérable.

Intention : cette alarme est utilisée pour éviter de manquer de mémoire, ce qui peut entraîner le rejet de connexions.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : en fonction de la charge de travail et de la classe d'instance, différentes valeurs de seuil peuvent être appropriées. Idéalement, la mémoire disponible ne devrait pas être inférieure à 25 % de la mémoire totale pendant de longues périodes. Pour Aurora, vous pouvez fixer un seuil proche de 5 %, car une métrique proche de 0 signifie que l'instance de la base de données a été mise à l'échelle autant qu'elle le pouvait. Vous pouvez analyser le comportement historique de cette métrique afin de déterminer des seuils raisonnables.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : LESS_THAN_THRESHOLD

FreeLocalStorage

Dimensioni : DB InstanceIdentifier

Description de l'alarme : cette alarme permet de surveiller le faible niveau de stockage local disponible. Aurora Édition compatible avec PostgreSQL utilise le stockage local pour stocker les journaux d'erreurs et les fichiers temporaires. Aurora MySQL utilise le stockage local pour stocker les journaux d'erreurs, les journaux généraux, les journaux de requêtes lentes, les journaux d'audit et les tables temporaires autres qu'InnoDB. Ces volumes de stockage locaux sont sauvegardés par Amazon EBS et peuvent être étendus en utilisant une classe d'instance de base de données plus grande. Pour résoudre les problèmes, vérifiez Aurora Édition compatible avec PostgreSQL et Édition compatible avec MySQL.

Intention : cette alarme est utilisée pour détecter dans quelle mesure l'instance de base de données Aurora est proche de la limite de stockage locale, si vous n'utilisez pas Aurora sans

serveur v2 ou version ultérieure. Le stockage local peut atteindre sa capacité maximale lorsque vous stockez des données non persistantes, telles que des tables temporaires et des fichiers journaux, dans le stockage local. Cette alarme peut empêcher qu'une out-of-space erreur ne se produise lorsque votre instance de base de données n'a plus de stockage local.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : vous devez évaluer environ 10 à 20 % de la quantité de stockage disponible en fonction de la vitesse et de la tendance de l'utilisation du volume, puis utiliser ce pourcentage comme valeur de seuil pour prendre des mesures proactives avant que le volume n'atteigne sa limite.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_THRESHOLD

FreeStorageSpace

Dimensioni : DB InstanceIdentifier

Description de l'alarme : cette alarme surveille la faible quantité d'espace de stockage disponible. Envisagez d'augmenter le stockage de votre base de données si vous approchez fréquemment des limites de capacité de stockage. Prévoyez une marge de manœuvre pour faire face aux augmentations imprévues des besoins de vos applications. Vous pouvez également envisager d'activer l'autoscaling du stockage RDS. Pensez également à libérer de l'espace en supprimant les données et les journaux inutilisés ou périmés. Pour plus d'informations, vérifiez le document RDS manque d'espace de stockage et le document Problèmes de stockage PostgreSQL.

Intention : cette alarme permet d'éviter les problèmes de stockage saturés. Cela permet d'éviter les temps d'arrêt qui surviennent lorsque votre instance de base de données est à court d'espace de stockage. Nous ne recommandons pas l'utilisation de cette alarme si l'option autoscaling du stockage est activée ou si vous modifiez fréquemment la capacité de stockage de l'instance de la base de données.

Statistique : minimum

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur du seuil dépend de l’espace de stockage actuellement alloué. En règle générale, vous devez évaluer la valeur de 10 % de l’espace de stockage alloué et utiliser ce pourcentage comme valeur de seuil.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_THRESHOLD

MaximumUsedTransactionIdentifiants

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet d’empêcher l’encapsulation des identifiants de transaction pour PostgreSQL. Reportez-vous aux étapes de résolution des problèmes décrites dans ce blog pour étudier et résoudre le problème. Vous pouvez également consulter ce blog pour vous familiariser davantage avec les concepts d’autovacuum, les problèmes courants et les meilleures pratiques.

Intention : cette alarme est utilisée pour empêcher l’encapsulation des identifiants de transaction pour PostgreSQL.

Statistique : moyenne

Seuil recommandé : 1,0E9

Justification du seuil : fixer ce seuil à 1 milliard devrait vous donner le temps d’étudier le problème. La valeur par défaut de autovacuum_freeze_max_age est de 200 millions. Si l’âge de la transaction la plus ancienne est de 1 milliard, autovacuum a du mal à maintenir ce seuil en dessous de l’objectif de 200 millions d’identifiants de transaction.

Période : 60

Points de données pour le déclenchement d'alarme : 1

Période d'évaluation : 1

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ReadLatency

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet de surveiller une latence de lecture élevée. Si la latence de stockage est élevée, c’est parce que la charge de travail dépasse les limites de ressources. Vous pouvez examiner l’utilisation des E/S par rapport à l’instance et à la configuration du stockage alloué. Reportez-vous au document Résoudre les problèmes de latence des volumes Amazon EBS causée par un goulot d’étranglement IOPS. Pour Aurora, vous pouvez passer à une classe d’instance dotée d’une configuration de stockage I/O-Optimized. Veuillez consulter l’article de blog Planning I/O in Aurora pour obtenir des conseils.

Intention : cette alarme est utilisée pour détecter une latence de lecture élevée. Les disques de base de données ont généralement une faible latence de lecture/écriture, mais ils peuvent connaître des problèmes susceptibles d’entraîner des opérations à latence élevée.

Statistique : p90

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de votre cas d'utilisation. Les latences de lecture supérieures à 20 millisecondes sont un motif raisonnable d’investigation. Vous pouvez également définir un seuil plus élevé si votre application peut supporter une latence plus élevée pour les opérations de lecture. Examinez la criticité et les exigences de la latence de lecture et analysez le comportement historique de cette métrique afin de déterminer des niveaux de seuil raisonnables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

ReplicaLag

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme vous aide à comprendre le nombre de secondes de retard d’un réplica par rapport à l’instance principale. Un réplica en lecture PostgreSQL consigne un

délai de réplication pouvant atteindre cinq minutes si aucune transaction utilisateur ne se produit sur l'instance de base de données source. Lorsque la ReplicaLag métrique atteint 0, la réplique a rattrapé l'instance de base de données principale. Si la ReplicaLag métrique renvoie -1, la réplication n'est actuellement pas active. Pour obtenir des conseils relatifs à RDS PostgreSQL, consultez les meilleures pratiques en matière de réplication et pour le `ReplicaLag` dépannage et les erreurs associées, consultez la section Résolution des problèmes. ReplicaLag

Intention : cette alarme peut détecter le décalage de réplication qui reflète la perte de données pouvant survenir en cas de défaillance de l'instance principale. Si le réplica prend trop de retard par rapport à l'instance principale et que cette dernière tombe en panne, les données qui se trouvaient dans l'instance principale seront manquantes dans le réplica.

Statistique : maximum

Seuil recommandé : 60,0

Justification du seuil : le délai acceptable dépend généralement de l'application. Nous recommandons de ne pas dépasser 60 secondes.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : GREATER_THAN_THRESHOLD

WriteLatency

Dimensioni : DB InstanceIdentifier

Description de l'alarme : cette alarme permet de surveiller une latence d'écriture élevée. Si la latence de stockage est élevée, c'est parce que la charge de travail dépasse les limites de ressources. Vous pouvez examiner l'utilisation des E/S par rapport à l'instance et à la configuration du stockage alloué. Reportez-vous au document Résoudre les problèmes de latence des volumes Amazon EBS causée par un goulot d'étranglement IOPS. Pour Aurora, vous pouvez passer à une classe d'instance dotée d'une configuration de stockage I/O-Optimized. Veuillez consulter l'article de blog Planning I/O in Aurora pour obtenir des conseils.

Intention : cette alarme est utilisée pour détecter une latence d'écriture élevée. Bien que les disques de base de données présentent généralement une faible latence de lecture/écriture, ils

peuvent connaître des problèmes susceptibles d’entraîner des opérations à latence élevée. La surveillance de ce phénomène vous permettra de vous assurer que la latence du disque est aussi faible que prévu.

Statistique : p90

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de votre cas d'utilisation. Les latences d’écriture supérieures à 20 millisecondes sont un motif raisonnable d’investigation. Vous pouvez également définir un seuil plus élevé si votre application peut supporter une latence plus élevée pour les opérations d’écriture. Examinez la criticité et les exigences de la latence d’écriture et analysez le comportement historique de cette métrique afin de déterminer des niveaux de seuil raisonnables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

DBLoad

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet de surveiller une charge de base de données élevée. Si le nombre de processus dépasse le nombre de vCPU, les processus sont mis en file d'attente. Lorsque la file d’attente augmente, les performances sont affectées. Si la charge de la base de données dépasse souvent le nombre maximal de processeurs virtuels et que l’état d’attente principal est CPU, cela signifie que le processeur est surchargé. Dans ce cas, vous pouvez surveiller `CPUUtilization`, `DBLoadCPU` et mettre des tâches en file d’attente dans Performance Insights/Surveillance améliorée. Vous pouvez décider de limiter les connexions à l’instance, d’ajuster les requêtes SQL dont la charge processeur est élevée ou d’opter pour une classe d’instance plus grande. Quel que soit leur état d'attente, les instances élevées et régulières indiquent que des problèmes de goulots d'étranglement ou de conflits de ressources devront peut-être être résolus.

Intention : cette alarme est utilisée pour détecter une charge élevée de la base de données. Une charge de base de données élevée peut entraîner des problèmes de performances dans

l’instance de base de données. Cette alarme ne s’applique pas aux instances de base de données sans serveur.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur du nombre maximal de vCPU est déterminée par le nombre de cœurs de vCPU (processeur virtuel) de votre instance de base de données. En fonction du nombre maximal de processeurs virtuels, différentes valeurs de seuil peuvent être appropriées. Idéalement, la charge de la base de données ne doit pas dépasser la limite de vCPU.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

AuroraVolumeBytesLeftTotal

Dimensioni : DB ClusterIdentifier

Description de l’alarme : cette alarme permet de surveiller un volume total restant faible. Lorsque le volume total restant atteint la limite de taille, le cluster signale une out-of-space erreur. Le stockage Aurora se met automatiquement à l’échelle en fonction des données contenues dans le volume du cluster et s’étend jusqu’à 128 TiB ou 64 TiB en fonction de la version du moteur de base de données. Envisagez de réduire l’espace de stockage en supprimant les tables et les bases de données dont vous n’avez plus besoin. Pour plus d’informations sur le dimensionnement du stockage, veuillez consulter la rubrique Dimensionnement du stockage.

Intention : cette alarme est utilisée pour détecter à quel point le cluster Aurora est proche de la limite de taille de volume. Cette alarme peut empêcher qu'une out-of-space erreur ne se produise lorsque votre cluster manque d'espace. Ce paramètre n’est disponible que pour Aurora.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : vous devez évaluer 10 à 20 % de la quantité de stockage disponible en fonction de la vitesse et de la tendance de l’utilisation du volume, puis utiliser ce pourcentage

comme valeur de seuil pour prendre des mesures proactives avant que le volume n'atteigne sa limite.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_THRESHOLD

AuroraBinlogReplicaLag

Dimensions : DBClusterIdentifier, Role=Writer

Description de l'alarme : cette alarme permet de surveiller l'état d'erreur de la réplication de l'instance Aurora d'enregistreur. Pour plus d'informations, consultez la section Réplication de clusters de bases de données Aurora MySQL entre AWS régions. Pour résoudre les problèmes, veuillez consulter la rubrique Problèmes de réplication Aurora MySQL.

Intention : cette alarme est utilisée pour détecter si l'instance d'enregistreur est dans un état d'erreur et n'est pas en mesure de répliquer la source. Ce paramètre n'est disponible que pour Aurora.

Statistique : moyenne

Seuil recommandé : -1,0

Justification du seuil : nous vous recommandons d'utiliser -1 comme valeur seuil, car Aurora MySQL publie cette valeur si la réplique est en état d'erreur.

Période : 60

Points de données pour le déclenchement d'alarme : 2

Période d'évaluation : 2

Opérateur de comparaison : LESS_THAN_OR_EQUAL_TO_THRESHOLD

BlockedTransactions

Dimensioni : DB InstanceIdentifier

Description de l'alarme : cette alarme permet de surveiller un nombre élevé de transactions bloquées dans une instance de base de données Aurora. Les transactions bloquées peuvent se

terminer par une restauration (rollback) ou une validation (commit). Un taux élevé de simultanéité, des interruptions de transaction ou des transactions de longue durée peuvent entraîner le blocage des transactions. Pour résoudre les problèmes, veuillez consulter la documentation d'Aurora MySQL.

Intention : cette alarme est utilisée pour détecter un nombre élevé de transactions bloquées dans une instance de base de données Aurora afin d'empêcher les restaurations de transactions et la dégradation des performances.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : vous devez évaluer 5 % de toutes les transactions de votre instance à l'aide de la métrique `ActiveTransactions` et utiliser ce pourcentage comme valeur de seuil. Vous pouvez également examiner la criticité et les exigences des transactions bloquées et analyser le comportement historique de cette métrique afin de déterminer des niveaux de seuil raisonnables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

BufferCacheHitRatio

Dimensioni : DB InstanceIdentifier

Description de l'alarme : cette alarme vous aide à surveiller un faible taux d'accès au cache du cluster Aurora. Un faible taux de réussite indique que vos requêtes sur cette instance de base de données vont fréquemment sur le disque. Pour résoudre les problèmes, examinez votre charge de travail pour voir quelles requêtes sont à l'origine de ce comportement, et consultez le document Recommandations RAM d'une instance de base de données.

Intention : cette alarme est utilisée pour détecter un faible taux d'accès au cache constant afin d'empêcher une baisse durable des performances de l'instance Aurora.

Statistique : moyenne

Seuil recommandé : 80,0

Justification du seuil : vous pouvez définir le seuil du taux d'accès au cache tampon à 80 %. Toutefois, vous pouvez ajuster cette valeur en fonction de votre niveau de performance acceptable et des caractéristiques de charge de travail.

Période : 60

Points de données pour le déclenchement d'alarme : 10

Période d'évaluation : 10

Opérateur de comparaison : LESS_THAN_THRESHOLD

EngineUptime

Dimensions : DBClusterIdentifier, Role=Writer

Description de l'alarme : cette alarme permet de surveiller les faibles temps d'arrêt de l'instance de base de données d'enregistreur. L'instance de base de données d'enregistreur peut tomber en panne en raison d'un redémarrage, d'une maintenance, d'une mise à niveau ou d'un basculement. Lorsque le temps de fonctionnement atteint zéro en raison d'un basculement dans le cluster, et que le cluster possède un ou plusieurs réplicas Aurora, un réplica Aurora est promu en tant qu'instance d'enregistreur principale lors d'un événement d'échec. Pour augmenter la disponibilité de votre cluster de base de données, envisagez de créer une ou plusieurs répliques Aurora dans deux ou plusieurs zones de disponibilité différentes. Pour plus d'informations, vérifiez les facteurs qui influent sur les temps d'arrêt d'Aurora.

Intention : cette alarme est utilisée pour détecter si l'instance de base de données d'enregistreur Aurora est en panne. Cela permet d'éviter une défaillance prolongée de l'instance d'enregistreur en raison d'un crash ou d'un basculement.

Statistique : moyenne

Seuil recommandé : 0,0

Justification du seuil : un événement d'échec se traduit par une brève interruption, pendant laquelle les opérations de lecture et d'écriture échouent avec une exception. Cependant, le service est généralement restauré en moins de 60 secondes, et souvent en moins de 30 secondes.

Période : 60

Points de données pour le déclenchement d'alarme : 2

Période d'évaluation : 2

Opérateur de comparaison : LESS_THAN_OR_EQUAL_TO_THRESHOLD

RollbackSegmentHistoryListLength

Dimensioni : DB InstanceIdentifier

Description de l’alarme : cette alarme permet de surveiller une durée constante et élevée de l’historique des segments de restauration d’une instance Aurora. Une durée élevée de la liste d’historique InnoDB indique qu’un grand nombre d’anciennes versions de lignes, de requêtes et d’arrêts de la base de données sont devenus plus lents. Pour plus d’informations et pour résoudre les problèmes, veuillez consulter la documentation La longueur de la liste d’historique InnoDB a considérablement augmenté.

Intention : cette alarme est utilisée pour détecter la longueur élevée et constante de l’historique des segments de restauration. Cela peut vous aider à éviter une dégradation durable des performances et une utilisation élevée du processeur dans l’instance Aurora. Ce paramètre n’est disponible que pour Aurora.

Statistique : moyenne

Seuil recommandé : 1 000 000,0

Justification du seuil : fixer ce seuil à 1 million devrait vous donner le temps d’étudier le problème. Toutefois, vous pouvez ajuster cette valeur en fonction de votre niveau de performance acceptable et des caractéristiques de charge de travail.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

StorageNetworkThroughput

Dimensions : DBClusterIdentifier, Role=Writer

Description de l’alarme : cette alarme permet de surveiller le débit élevé du réseau de stockage. Si le débit du réseau de stockage dépasse la bande passante du réseau totale de l’instance EC2, il peut en résulter une latence élevée en lecture et en écriture, ce qui peut entraîner une

dégradation des performances. Vous pouvez vérifier le type de votre instance EC2 depuis AWS la console. Pour résoudre les problèmes, vérifiez les modifications apportées aux latences d'écriture/de lecture et déterminez si vous avez également déclenché une alarme pour cette métrique. Si c'est le cas, évaluez votre schéma de charge de travail pendant les périodes où l'alarme s'est déclenchée. Cela peut vous aider à déterminer si vous pouvez optimiser votre charge de travail afin de réduire le volume total du trafic réseau. Si cela n'est pas possible, vous devrez peut-être envisager de mettre votre instance à l'échelle.

Intention : cette alarme est utilisée pour détecter un débit élevé du réseau de stockage. La détection d'un débit élevé peut empêcher les pertes de paquets réseau et la dégradation des performances.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : vous devez évaluer environ 80 % à 90 % de la bande passante du réseau totale du type d'instance EC2, puis utiliser ce pourcentage comme valeur de seuil pour agir de manière proactive avant que les paquets réseau ne soient affectés. Vous pouvez également examiner la criticité et les exigences du débit du réseau de stockage et analyser le comportement historique de cette métrique afin de déterminer des seuils raisonnables.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon Route 53 Public Data Plane

HealthCheckStatus

Dimensions : HealthCheckId

Description de l'alarme : cette alarme permet de détecter les points de terminaison non sains selon les outils de surveillance de l'état. Pour comprendre la raison d'un échec entraînant un état non sain, utilisez l'onglet des outils de surveillance de l'état de la console de surveillance de l'état Route 53 pour consulter l'état de chaque région ainsi que le dernier échec de surveillance de

l'état. L'onglet d'état indique également la raison pour laquelle le point de terminaison est signalé comme n'étant pas sain. Reportez-vous aux étapes de résolution des problèmes.

Intention : cette alarme utilise les outils de surveillance de l'état Route 53 pour détecter les points de terminaison non sains.

Statistique : moyenne

Seuil recommandé : 1,0

Justification du seuil : l'état du point de terminaison est signalé par la valeur 1 lorsqu'il est sain. Tout ce qui est inférieur à 1 n'est pas sain.

Période : 60

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : LESS_THAN_THRESHOLD

Amazon S3

4xxErrors

Dimensions :BucketName, FilterId

Description de l'alarme : cette alarme nous permet de signaler le nombre total de codes d'état d'erreur 4xx créés en réponse aux requêtes des clients. Les codes d'erreur 403 peuvent indiquer une politique IAM incorrecte, et les codes d'erreur 404 peuvent indiquer un mauvais comportement de l'application client, par exemple. L'activation de la journalisation des accès au serveur S3 vous aidera à identifier l'origine du problème à l'aide des champs Code de statut HTTP et Code d'erreur. Pour en savoir plus sur le code d'erreur, veuillez consulter Réponses aux erreurs.

Objectif : cette alarme est utilisée pour créer une base de référence pour les taux d'erreur 4xx typiques afin que vous puissiez examiner toute anomalie susceptible d'indiquer un problème de configuration.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : le seuil recommandé est de détecter si plus de 5 % du total des requêtes reçoivent des erreurs 4xx. Les erreurs 4xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible. Vous pouvez également ajuster le seuil en fonction de la charge des requêtes, en tenant compte d'un niveau acceptable des erreurs 4xx. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

5xxErrors

Dimensions :BucketName, FilterId

Description de l'alarme : cette alarme vous permet de détecter un grand nombre d'erreurs côté serveur. Ces erreurs indiquent qu'un client a émis une requête que le serveur n'a pas pu traiter. Cela peut vous aider à établir une corrélation entre le problème auquel votre application est confrontée à cause de S3. Pour plus d'informations qui vous aideront à gérer ou à réduire efficacement les erreurs, veuillez consulter Optimisation des modèles de conception des performances. Des erreurs peuvent également être causées par un problème avec S3. Consultez le tableau de bord de l'état du service AWS pour connaître l'état d'Amazon S3 dans votre région.

Intention : cette alarme peut aider à détecter si l'application rencontre des problèmes dus à des erreurs 5xx.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : nous recommandons de définir le seuil pour détecter si plus de 5 % du total des requêtes reçoivent une erreur 5xx. Vous pouvez toutefois ajuster le seuil en fonction du trafic des requêtes, ainsi que des taux d'erreur acceptables. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

OperationsFailedReplication

SourceBucketDimensioni : DestinationBucket, RuleId

Description de l'alarme : cette alarme aide à comprendre un échec de réplication. Cette métrique suit l'état des nouveaux objets répliqués à l'aide de S3 CRR ou S3 SRR, et suit également les objets existants répliqués à l'aide de la réplication par lots S3. Veuillez consulter Résolution des problèmes de réplication pour plus de détails.

Intention : cette alarme est utilisée pour détecter l'échec d'une opération de réplication.

Statistique : maximum

Seuil recommandé : 0,0

Justification du seuil : cette métrique émet une valeur de 0 pour les opérations réussies, et rien lorsqu'aucune opération de réplication n'est effectuée dans la minute. Lorsque la métrique émet une valeur supérieure à 0, cela signifie que l'opération de réplication a échoué.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

S3ObjectLambda

4xxErrors

Dimensions :AccessPointName, DataSource ARN

Description de l'alarme : cette alarme nous aide à signaler le nombre total de codes d'erreur 4xx créés en réponse aux requêtes des clients. L'activation de la journalisation des accès au serveur S3 vous aidera à identifier l'origine du problème à l'aide des champs Code de statut HTTP et Code d'erreur.

Objectif : cette alarme est utilisée pour créer une base de référence pour les taux d'erreur 4xx typiques afin que vous puissiez examiner toute anomalie susceptible d'indiquer un problème de configuration.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : nous vous recommandons de définir le seuil pour détecter si plus de 5 % du total des requêtes reçoivent une erreur 4xx. Les erreurs 4xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible. Vous pouvez également ajuster le seuil en fonction de la charge des requêtes, en tenant compte d'un niveau acceptable des erreurs 4xx. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

5xxErrors

Dimensions :AccessPointName, DataSource ARN

Description de l'alarme : cette alarme permet de détecter un nombre élevé d'erreurs côté serveur. Ces erreurs indiquent qu'un client a émis une requête que le serveur n'a pas pu traiter. Ces erreurs peuvent être dues à un problème avec S3. Consultez le tableau de bord de l'état du service AWS pour connaître l'état d'Amazon S3 dans votre région. Cela peut vous aider à établir une corrélation entre le problème auquel votre application est confrontée à cause de S3. Pour obtenir des informations qui vous aideront à gérer ou à réduire efficacement ces erreurs, veuillez consulter Optimisation des modèles de conception des performances.

Intention : cette alarme peut aider à détecter si l'application rencontre des problèmes dus à des erreurs 5xx.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : nous recommandons de définir le seuil pour détecter si plus de 5 % du total des requêtes reçoivent des erreurs 5xx. Vous pouvez toutefois ajuster le seuil en fonction du trafic des requêtes, ainsi que des taux d'erreur acceptables. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

LambdaResponse4xx

Dimensions :AccessPointName, DataSource ARN

Description de l'alarme : cette alarme vous aide à détecter et à diagnostiquer les défaillances (500) lors des appels à S3 Object Lambda. Ces erreurs peuvent être causées par des erreurs ou de mauvaises configurations dans la fonction Lambda chargée de répondre à vos requêtes. L'étude des flux de CloudWatch log de la fonction Lambda associée au point d'accès Object Lambda peut vous aider à identifier l'origine du problème en fonction de la réponse de S3 Object Lambda.

Intention : Cette alarme est utilisée pour détecter les erreurs du client 4xx lors des WriteGetObjectResponse appels.

Statistique : moyenne

Seuil recommandé : 0,05

Justification du seuil : nous vous recommandons de définir le seuil pour détecter si plus de 5 % du total des requêtes reçoivent une erreur 4xx. Les erreurs 4xx fréquentes doivent faire l'objet d'une alarme. Cependant, le réglage d'une valeur très faible pour le seuil peut rendre l'alarme trop sensible. Vous pouvez également ajuster le seuil en fonction de la charge des requêtes, en tenant compte d'un niveau acceptable des erreurs 4xx. Vous pouvez également analyser les données historiques afin de déterminer le taux d'erreur acceptable pour la charge de travail de l'application, puis ajuster le seuil en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon SNS

NumberOfMessagesPublished

Dimensions : TopicName

Description de l'alarme : cette alarme peut détecter lorsque le nombre de messages SNS publiés est trop faible. Pour résoudre les problèmes, vérifiez pourquoi les diffuseurs de publication envoient moins de trafic.

Objectif : cette alarme vous permet de surveiller et de détecter de manière proactive les baisses importantes du nombre de notifications publiées. Cela vous aide à identifier les problèmes potentiels liés à votre application ou à vos processus métier, afin que vous puissiez prendre les mesures appropriées pour maintenir le flux de notifications attendu. Vous devez créer cette alarme si vous vous attendez à ce que le trafic de votre système soit réduit au minimum.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : le nombre de messages publiés doit correspondre au nombre attendu de messages publiés pour votre application. Vous pouvez également analyser les données historiques, les tendances et le trafic pour trouver le bon seuil.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_THRESHOLD

NumberOfNotificationsDelivered

Dimensions : TopicName

Description de l'alarme : cette alarme peut détecter lorsque le nombre de messages SNS délivrés est trop faible. Cela peut être dû à la désinscription involontaire d'un point de terminaison ou à un événement SNS qui retarde les messages.

Intention : cette alarme vous aide à détecter une baisse du volume des messages délivrés. Vous devez créer cette alarme si vous vous attendez à ce que le trafic de votre système soit réduit au minimum.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : le nombre de messages délivrés doit être conforme au nombre attendu de messages produits et au nombre de consommateurs. Vous pouvez également analyser les données historiques, les tendances et le trafic pour trouver le bon seuil.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : LESS_THAN_THRESHOLD

NumberOfNotificationsFailed

Dimensions : TopicName

Description de l'alarme : cette alarme peut détecter lorsque le nombre de messages SNS ayant échoué est trop élevé. Pour résoudre les problèmes d'échec des notifications, activez la journalisation dans CloudWatch Logs. La consultation des journaux peut vous aider à identifier les abonnés défaillants, ainsi que les codes de statut qu'ils renvoient.

Objectif : cette alarme vous aide à détecter de manière proactive les problèmes liés à l'envoi des notifications et à prendre les mesures appropriées pour y remédier.

Statistique : somme

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement de l'impact des notifications échouées. Passez en revue les SLA fournis à vos utilisateurs finaux, la

tolérance aux pannes et le caractère critique des notifications, analysez les données historiques, puis sélectionnez un seuil en conséquence. Le nombre de notifications ayant échoué doit être de 0 pour les rubriques qui n'ont que des abonnements SQS, Lambda ou Firehose.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidAttributes

Dimensions : TopicName

Description de l'alarme : cette alarme permet de surveiller et de résoudre les problèmes potentiels avec le diffuseur de publication ou les abonnés. Vérifiez si un diffuseur de publication publie des messages dont les attributs ne sont pas valides ou si un filtre inapproprié est appliqué à un abonné. Vous pouvez également analyser CloudWatch les journaux pour identifier la cause première du problème.

Intention : l'alarme est utilisée pour détecter si les messages publiés ne sont pas valides ou si des filtres inappropriés ont été appliqués à un abonné.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : les attributs non valides sont presque toujours le résultat d'une erreur de la part du diffuseur de publication. Nous vous recommandons de définir le seuil à 0, car aucun attribut non valide n'est attendu dans un système sain.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

NumberOfNotificationsFilteredOut-InvalidMessageBody

Dimensions : TopicName

Description de l'alarme : cette alarme permet de surveiller et de résoudre les problèmes potentiels avec le diffuseur de publication ou les abonnés. Vérifiez si un diffuseur de publication publie des messages dont le corps de message n'est pas valide ou si un filtre inapproprié est appliqué à un abonné. Vous pouvez également analyser CloudWatch les journaux pour identifier la cause première du problème.

Intention : l'alarme est utilisée pour détecter si les messages publiés ne sont pas valides ou si des filtres inappropriés ont été appliqués à un abonné.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : les corps de message non valides sont presque toujours le résultat d'une erreur de la part du diffuseur de publication. Nous vous recommandons de définir le seuil à 0, car aucun corps de message non valide n'est attendu dans un système sain.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

NumberOfNotificationsRedrivenToDlq

Dimensions : TopicName

Description de l'alarme : cette alarme permet de surveiller le nombre de messages déplacés vers une file d'attente de lettres mortes.

Intention : l'alarme est utilisée pour détecter les messages placés dans une file d'attente de lettres mortes. Nous vous recommandons de créer cette alarme lorsque SNS est couplé à SQS, Lambda ou Firehose.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : dans un système sain, quel que soit le type d'abonné, les messages ne doivent pas être placés dans la file d'attente de lettres mortes. Nous vous recommandons d'être averti si des messages arrivent dans la file d'attente, afin que vous puissiez identifier et traiter la

cause première, et éventuellement rediriger les messages dans la file d'attente de lettres mortes afin d'éviter toute perte de données.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

NumberOfNotificationsFailedToRedriveToDlq

Dimensions : TopicName

Description de l'alarme : cette alarme permet de surveiller les messages qui n'ont pas pu être déplacés vers une file d'attente de lettres mortes. Vérifiez si votre file d'attente de lettres mortes existe et qu'elle est correctement configurée. Vérifiez également que le SNS est autorisé à accéder à la file d'attente de lettres mortes. Veuillez consulter la documentation relative aux files d'attente de lettres mortes pour en savoir plus.

Intention : l'alarme est utilisée pour détecter les messages qui n'ont pas pu être déplacés vers une file d'attente de lettres mortes.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : si les messages ne peuvent pas être déplacés vers la file d'attente de lettres mortes, il s'agit presque toujours d'une erreur. La recommandation pour le seuil est 0, ce qui signifie que tous les messages dont le traitement échoue doivent pouvoir atteindre la file d'attente de lettres mortes lorsque celle-ci a été configurée.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

SMS en MonthToDateSpent dollars américains

Dimensions : TopicName

Description de l'alarme : l'alarme permet de vérifier si votre compte dispose d'un quota suffisant pour que SNS puisse envoyer des messages. Si vous atteignez votre quota, SNS ne sera pas en mesure de délivrer de SMS. Pour plus d'informations sur la définition de votre quota de dépenses mensuel par SMS, ou pour savoir comment demander une augmentation du quota de dépenses avec AWS, consultez la section Configuration des préférences de messagerie SMS.

Intention : cette alarme est utilisée pour détecter si votre compte dispose d'un quota suffisant pour que vos SMS soient envoyés avec succès.

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil conformément au quota (limite de dépenses du compte) pour le compte. Choisissez un seuil qui vous informe suffisamment tôt que vous atteignez votre limite de quota afin que vous ayez le temps de demander une augmentation.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

SMS SuccessRate

Dimensions : TopicName

Description de l'alarme : cette alarme permet de surveiller le taux d'échec des livraisons de SMS. Vous pouvez configurer Cloudwatch Logs pour comprendre la nature de la défaillance et prendre des mesures en conséquence.

Intention : cette alarme est utilisée pour détecter les échecs de livraison de SMS.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : définissez le seuil d'alarme en fonction de votre tolérance en cas d'échec de livraison de SMS.

Période : 60

Points de données pour le déclenchement d'alarme : 5

Période d'évaluation : 5

Opérateur de comparaison : GREATER_THAN_THRESHOLD

Amazon SQS

ApproximateAgeOfOldestMessage

Dimensions : QueueName

Description de l'alarme : cette alarme surveille l'âge du plus ancien message de la file d'attente. Vous pouvez utiliser cette alarme pour vérifier si vos clients traitent les messages SQS à la vitesse souhaitée. Envisagez d'augmenter le nombre de clients ou le débit des clients afin de réduire l'âge des messages. Cette métrique peut être utilisée en combinaison avec `ApproximateNumberOfMessagesVisible` pour déterminer l'ampleur du backlog de files d'attente et la rapidité avec laquelle les messages sont traités. Pour éviter que les messages ne soient supprimés avant leur traitement, pensez à configurer la file d'attente de lettres mortes afin de mettre de côté les messages potentiels de type « poison pill ».

Intention : Cette alarme est utilisée pour détecter si l'âge du message le plus ancien de la QueueName file d'attente est trop élevé. Un âge élevé peut indiquer que les messages ne sont pas traités assez rapidement ou que certains messages considérés comme « poison pill » sont bloqués dans la file d'attente et ne peuvent pas être traités.

Statistique : maximum

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement du temps de traitement des messages attendu. Vous pouvez utiliser les données historiques pour calculer le temps moyen de traitement des messages, puis définir le seuil à 50 % de plus que le temps de traitement maximal attendu des messages SQS par les consommateurs de files d'attente.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesNotVisible

Dimensions : QueueName

Description de l'alarme : cette alarme permet de détecter un grand nombre de messages en vol en ce qui concerne `QueueName`. Pour résoudre les problèmes, vérifiez comment empêcher l'augmentation du backlog de messages.

Objectif : cette alarme est utilisée pour détecter un grand nombre de messages en vol dans la file d'attente. Si les consommateurs ne suppriment pas les messages dans le délai imparti, lorsque la file d'attente est interrogée, les messages réapparaissent dans la file d'attente. Pour les files d'attente FIFO, il peut y avoir un maximum de 20 000 messages en vol. Si vous atteignez ce quota, SQS ne renvoie aucun message d'erreur. Une file d'attente FIFO examine les 20 000 premiers messages pour déterminer les groupes de messages disponibles. Cela signifie que si vous avez un arriéré de messages dans un seul groupe de messages, vous ne pouvez pas consommer les messages d'autres groupes de messages envoyés à la file d'attente ultérieurement tant que vous n'avez pas correctement consommé les messages du backlog.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : la valeur de seuil recommandée pour cette alarme dépend fortement du nombre attendu de messages en vol. Vous pouvez utiliser les données historiques pour calculer le nombre maximum attendu de messages en vol et définir le seuil à 50 % au-dessus de cette valeur. Si les utilisateurs de la file d'attente traitent des messages, mais ne les suppriment pas, ce nombre augmentera soudainement.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

ApproximateNumberOfMessagesVisible

Dimensions : QueueName

Description de l'alarme : cette alarme détecte que le backlog de la file d'attente de messages est plus important que prévu, ce qui indique que les consommateurs sont trop lents ou qu'il n'y en a pas assez. Envisagez d'augmenter le nombre de consommateurs ou de les accélérer si cette alarme passe en état ALARM.

Intention : cette alarme est utilisée pour détecter si le nombre de messages de la file d'attente active est trop élevé et si les consommateurs sont lents à traiter les messages ou s'il n'y en a pas assez pour les traiter.

Statistique : moyenne

Seuil recommandé : dépend de votre situation

Justification du seuil : un nombre étonnamment élevé de messages visibles indique que les messages ne sont pas traités par le consommateur au rythme attendu. Vous devez prendre en compte les données historiques lorsque vous définissez ce seuil.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : GREATER_THAN_OR_EQUAL_TO_THRESHOLD

NumberOfMessagesSent

Dimensions : QueueName

Description de l'alarme : cette alarme permet de détecter si aucun message n'est envoyé par un producteur en ce qui concerne `QueueName`. Pour résoudre les problèmes, vérifiez la raison pour laquelle le producteur n'envoie pas de messages.

Intention : cette alarme est utilisée pour détecter le moment où un producteur arrête d'envoyer des messages.

Statistique : somme

Seuil recommandé : 0,0

Justification du seuil : si le nombre de messages envoyés est égal à 0, cela signifie que le producteur n'envoie aucun message. Si le TPS de cette file d'attente est faible, augmentez-en le nombre EvaluationPeriods en conséquence.

Période : 60

Points de données pour le déclenchement d'alarme : 15

Période d'évaluation : 15

Opérateur de comparaison : LESS_THAN_OR_EQUAL_TO_THRESHOLD

AWS VPN

TunnelState

Dimensions : VpnId

Description de l'alarme : cette alarme vous aide à comprendre si l'état d'un ou de plusieurs tunnels est HORS SERVICE. Pour résoudre les problèmes, veuillez consulter Résoudre les problèmes liés aux tunnels VPN.

Intention : cette alarme est utilisée pour détecter si au moins un tunnel est à l'état HORS SERVICE pour ce VPN, afin que vous puissiez dépanner le VPN concerné. Cette alarme sera toujours à l'état ALARM pour les réseaux qui n'ont qu'un seul tunnel configuré.

Statistique : minimum

Seuil recommandé : 1,0

Justification du seuil : une valeur inférieure à 1 indique qu'au moins un tunnel est à l'état HORS SERVICE.

Période : 300

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : LESS_THAN_THRESHOLD

TunnelState

Dimensions : TunnelIpAddress

Description de l'alarme : cette alarme vous permet de savoir si l'état de ce tunnel est HORS SERVICE. Pour résoudre les problèmes, veuillez consulter Résoudre les problèmes liés aux tunnels VPN.

Intention : cette alarme est utilisée pour détecter si le tunnel est à l'état HORS SERVICE, afin que vous puissiez dépanner le VPN concerné. Cette alarme sera toujours à l'état ALARM pour les réseaux qui n'ont qu'un seul tunnel configuré.

Statistique : minimum

Seuil recommandé : 1,0

Justification du seuil : une valeur inférieure à 1 indique que le tunnel est à l'état HORS SERVICE.

Période : 300

Points de données pour le déclenchement d'alarme : 3

Période d'évaluation : 3

Opérateur de comparaison : LESS_THAN_THRESHOLD

Créer des alertes sur les métriques

Les étapes décrites dans les sections suivantes expliquent comment créer des CloudWatch alarmes sur des métriques.

Création d'une CloudWatch alarme basée sur un seuil statique

Vous choisissez une CloudWatch métrique pour l'alarme à surveiller, ainsi que le seuil pour cette métrique. L'alerte passe à l'état `ALARM` quand la métrique dépasse le seuil pendant un certain nombre de périodes d'évaluation.

Si vous créez une alarme dans un compte configuré en tant que compte de surveillance dans l'observabilité CloudWatch entre comptes, vous pouvez configurer l'alarme pour qu'elle surveille une métrique dans un compte source associé à ce compte de surveillance. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

Pour créer une alerte basée sur une métrique unique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Select Metric (Sélectionner une métrique).

5. Effectuez l’une des actions suivantes :
   - Choisissez l'espace de noms du service qui contient la métrique de votre choix. Ensuite, choisissez des options au fur et à mesure qu'elles s'affichent pour affiner les choix. Lorsqu'une liste des métriques apparaît, sélectionnez la case à cocher en regard de la métrique voulue.
   - Dans le champ de recherche, saisissez le nom d'une métrique, l'ID du compte, l'étiquette du compte, la dimension ou l'ID de la ressource. Ensuite, choisissez l'un des résultats et continuez jusqu'à ce qu'une liste des métriques s'affiche. Cochez la case en regard de la métrique voulue.
6. Sélectionnez l'onglet Graphed metrics (Graphiques des métriques).
   a. Sous Statistiques, choisissez l'une des statistiques ou l'un des centiles prédéfinis, ou spécifiez un centile personnalisé (par exemple, **`p95.45`**).
   b. Sous Période, choisissez la période d'évaluation de l'alerte. Lors de l'évaluation de l'alerte, chaque période est regroupée en un point de données.

      Vous pouvez également choisir que la légende de l'axe des Y s'affiche à gauche ou à droite lors de la création de l'alerte. Cette préférence est utilisée uniquement lorsque vous créez l'alerte.
   c. Choisissez Select metric (Sélectionner une métrique).

      La page Specify metric and conditions (Spécifier les métriques et les conditions) apparaît, présentant un graphique et d'autres informations sur la métrique et la statistique que vous avez sélectionnées.
7. Sous Conditions, spécifiez les éléments suivants :
   a. Pour Whenever ***`metric`*** is (À chaque fois que la métrique est), spécifiez si la métrique doit être supérieure à, inférieure à ou égale au seuil. Dans than... (à...), spécifiez la valeur de seuil.
   b. 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 pour la première valeur un nombre inférieur à celui de la seconde valeur. Pour plus d'informations, consultez . Évaluation d'une alerte.

c. 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, consultez . Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.

d. Si l'alerte utilise un centile comme statistique surveillée, une zone Percentiles with low samples (Centiles avec exemples de bas niveau) s'affiche. Utilisez-la pour choisir si vous souhaitez évaluer ou ignorer les cas avec des taux d'échantillons faibles. Si vous sélectionnez ignore (ignorer : conserver l'état d'alerte), l'état actuel de l'alerte est toujours conservé lorsque la taille de l'échantillon est trop réduite. Pour plus d'informations, consultez . CloudWatch Alarmes basées sur les percentiles et échantillons de données faibles.

8. Choisissez Next (Suivant).

9. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alerte passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

   Pour que l'alerte envoie plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, choisissez Add notification (Ajouter une notification).

   Dans CloudWatch l'observabilité entre comptes, vous pouvez choisir d'envoyer des notifications à plusieurs AWS comptes. Par exemple, à la fois au compte de surveillance et au compte source.

   Pour que l'alerte n'envoie pas de notifications, choisissez Remove (Supprimer).

10. Pour que l'alarme exécute des actions Auto Scaling, EC2, Lambda ou Systems Manager, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer. Les alertes peuvent effectuer des actions du Systems Manager uniquement lorsqu'elles passent à l'état ALARM. Pour plus d'informations sur les actions de Systems Manager, consultez les sections Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

    > **Note**
    >
    > Pour créer une alerte qui exécute une action SSM Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager.

11. Lorsque vous avez terminé, choisissez Next (Suivant).

12. Saisissez un nom et une description pour l'alerte. Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes. Sélectionnez ensuite Next (Suivant).
13. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

Vous pouvez également ajouter des alertes à un tableau de bord. Pour plus d'informations, consultez Ajouter ou supprimer un widget d'alarme dans un CloudWatch tableau de bord .

Création d'une CloudWatch alarme basée sur une expression mathématique métrique

Pour créer une alarme basée sur une expression mathématique métrique, choisissez une ou plusieurs CloudWatch métriques à utiliser dans l'expression. Ensuite, spécifiez l'expression, le seuil et les périodes d'évaluation.

Vous ne pouvez pas créer d'alerte basée sur l'expression SEARCH. En effet, les expressions de recherche renvoient plusieurs séries temporelles, et une alerte basée sur une expression mathématique ne peut regarder qu'une seule série temporelle.

Créer une alerte basée sr une expression mathématique appliquée à une métrique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), puis All alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Select Metric (Sélectionner une métrique) et effectuez l'une des actions suivantes :
   - Sélectionnez un espace de noms dans le menu déroulant AWS namespaces (Espaces de noms) ou dans le menu déroulant Custom namespaces (Espace de noms personnalisés). Une fois que vous avez sélectionné un espace de noms, vous continuez à choisir des options jusqu'à ce qu'une liste de métriques apparaisse, dans laquelle vous cochez la case en regard de la métrique appropriée.
   - Utilisez le champ de recherche pour trouver une métrique, un ID de compte, une dimension ou un ID de ressource. Une fois que vous avez saisi la métrique, la dimension ou l'ID

de ressource, vous continuez à choisir des options jusqu'à ce qu'une liste de métriques apparaisse, dans laquelle vous cochez la case en regard de la métrique appropriée.

5. (Facultatif) Si vous souhaitez ajouter une autre métrique à une expression mathématique appliquée à une métrique, vous pouvez utiliser la zone de recherche pour trouver une métrique spécifique. Vous pouvez ajouter jusqu'à 10 métriques à une expression mathématique appliquée à une métrique.
6. Sélectionnez l'onglet Graphed metrics (Métriques sous forme graphique). Pour chacune des métriques que vous avez précédemment ajoutées, effectuez les actions suivantes :

   a. Sous la colonne Statistic (Statistique), sélectionnez le menu déroulant. Dans le menu déroulant, choisissez l'une des statistiques ou des centiles prédéfinis. Utilisez la zone de recherche du menu déroulant pour spécifier un centile personnalisé.

   b. Sous la colonne Period (Période), sélectionnez le menu déroulant. Dans le menu déroulant, choisissez l'une des périodes d'évaluation prédéfinies.

      Lors de la création de votre alerte, vous pouvez choisir que la légende de l'axe des Y s'affiche à gauche ou à droite de votre graphique.

   > **Note**
   >
   > Lors de CloudWatch l'évaluation des alarmes, les périodes sont agrégées en points de données uniques.

7. Choisissez le menu déroulant Add math (Ajouter des mathématiques), puis sélectionnez Start with an empty expression (Commencer par une expression vide) dans la liste des expressions mathématiques prédéfinies appliquées à une métrique.

   Après avoir sélectionné Start with an empty expression (Commencer par une expression vide), une zone d'expression mathématique apparaît à l'endroit où vous appliquez ou modifiez des expressions mathématiques.
8. Dans la zone d'expression mathématique, saisissez votre expression mathématique, puis sélectionnez Apply (Appliquer).

   Après avoir sélectionné Apply (Appliquer), une colonne ID apparaît à côté de la colonne Label (Étiquette).

Pour utiliser une métrique ou le résultat d'une autre expression mathématique appliquée à une métrique dans le cadre de la formule de votre expression mathématique actuelle, vous utilisez la valeur indiquée sous la colonne ID. Pour modifier la valeur de l'ID, vous devez sélectionner l' pen-and-paper icône à côté de la valeur actuelle. La nouvelle valeur doit commencer par une lettre minuscule et peut comprendre des chiffres, des lettres et des traits de soulignement. Le fait de remplacer la valeur de l'ID par un nom plus descriptif peut faciliter la compréhension de votre graphique d'alerte.

Pour plus d'informations sur les fonctions disponibles pour les mathématiques appliquées aux métriques, consultez Syntaxe et fonctions des mathématiques appliquées aux métriques.

9. (En option), ajoutez des expressions mathématiques supplémentaires, en utilisant à la fois des métriques et des résultats d'autres expressions mathématiques dans les formules des nouvelles expressions mathématiques.
10. Lorsque vous disposez de l'expression pour l'alerte, désactivez les cases à cocher à gauche de chaque autre expression et de chaque métrique sur la page. Seule la case à cocher en regard de l'expression à utiliser pour l'alerte doit être activée. L'expression que vous choisissez pour l'alerte doit produire une seule série temporelle, et afficher une seule ligne sur le graphique. Ensuite, choisissez Select metric (Sélectionner une métrique).

    La page Specify metric and conditions (Spécifier les métriques et les conditions) apparaît, présentant un graphique et d'autres informations sur l'expression mathématique que vous avez sélectionnée.
11. Pour Whenever ***`expression`*** is (À chaque fois que l'expression est), spécifiez si la métrique doit être supérieure à, inférieure à ou égale au seuil. Dans than... (à...), spécifiez la valeur de seuil.
12. 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 pour la première valeur un nombre inférieur à celui de la seconde valeur. Pour plus d'informations, consultez . Évaluation d'une alerte.
13. 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,

consultez . Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.

14. Choisissez Next (Suivant).
15. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alerte passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Pour que l'alerte envoie plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, choisissez Add notification (Ajouter une notification).

    Pour que l'alerte n'envoie pas de notifications, choisissez Remove (Supprimer).
16. Pour que l'alarme exécute des actions Auto Scaling, EC2, Lambda ou Systems Manager, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer. Si vous choisissez une fonction Lambda comme action d'alarme, vous spécifiez le nom de la fonction ou l'ARN, et vous pouvez éventuellement choisir une version spécifique de la fonction.

    Les alertes peuvent effectuer des actions du Systems Manager uniquement lorsqu'elles passent à l'état ALARM. Pour plus d'informations sur les actions de Systems Manager, voir Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

    > **Note**
    >
    > Pour créer une alerte qui exécute une action SSM Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager.
17. Lorsque vous avez terminé, choisissez Next (Suivant).
18. Saisissez un nom et une description pour l'alerte. Ensuite, sélectionnez Suivant.

    Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.
19. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

Vous pouvez également ajouter des alertes à un tableau de bord. Pour plus d'informations, consultez Ajouter ou supprimer un widget d'alarme dans un CloudWatch tableau de bord .

Création d'une CloudWatch alarme basée sur une requête Metrics Insights

Vous pouvez créer une alarme sur toute requête Metrics Insights qui renvoie une seule série temporelle. Cela peut être particulièrement utile pour créer des alarmes dynamiques qui surveillent des métriques agrégées sur une flotte de votre infrastructure ou de vos applications. Créez l'alarme une fois, et elle s'ajuste au fur et à mesure que des ressources sont ajoutées ou retirées de la flotte. Par exemple, vous pouvez créer une alarme qui surveille l'ulitisation du CPU de toutes vos instances, et l'alarme s'ajuste dynamiquement lorsque vous ajoutez ou supprimez des instances.

Pour obtenir des instructions complètes, veuillez consulter Création d'alarmes sur les requêtes Metrics Insights.

Création d’une alarme basée sur une source de données connectée

Vous pouvez créer des alarmes qui surveillent les métriques provenant de sources de données absentes CloudWatch. Pour plus d’informations sur la création de connexions avec ces autres sources de données, veuillez consulter Interrogation de métriques d’autres sources de données.

Pour créer une alarme sur les métriques d’une source de données à laquelle vous êtes connecté

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques, Toutes les métriques.
3. Choisissez l’onglet Requête multisource.
4. Pour Sources de données, choisissez le nom de la source de données à utiliser.
5. Le générateur de requêtes vous invite à saisir les informations nécessaires à la requête afin de récupérer les métriques à utiliser pour l’alarme. Le flux de travail est différent pour chaque source de données et est adapté à la source de données. Par exemple, pour les sources de données Amazon Managed Service for Prometheus et Prometheus, un éditeur de requêtes PromQL avec un assistant de requête apparaît.
6. Lorsque vous avez terminé de créer la requête, choisissez Requête graphique.
7. Si l’exemple de graphique se présente comme vous le souhaitez, choisissez Créer une alarme.
8. La page Indiquer les métriques et les conditions s’affiche. Si la requête que vous utilisez produit plus d’une série temporelle, une bannière d’avertissement s’affiche en haut de la page. Si c’est le cas, sélectionnez une fonction à utiliser pour agréger les séries temporelles dans la fonction d’agrégation.
9. (Facultatif) Ajoutez une étiquette pour l’alarme.

10. 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.
11. 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 Évaluation d'une alerte.
12. 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, consultez Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.
13. Choisissez Suivant.
14. Sous Notification, spécifiez une rubrique Amazon SNS qui doit recevoir une notification lorsque l'alarme passe à l'état `ALARM`, `OK` 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.
15. Pour que l'alarme exécute des actions Auto Scaling, EC2, Lambda ou Systems Manager, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer. Si vous choisissez une fonction Lambda comme action d'alarme, vous spécifiez le nom de la fonction ou l'ARN, et vous pouvez éventuellement choisir une version spécifique de la fonction.

Les alertes peuvent effectuer des actions du Systems Manager uniquement lorsqu'elles passent à l'état ALARM. Pour plus d'informations sur les actions de Systems Manager, voir Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

> **Note**
>
> Pour créer une alerte qui exécute une action SSM Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager.

16. Choisissez Suivant.
17. Sous Add a description (Ajouter une description), saisissez un nom et une description pour l'alerte et choisissez Next (Suivant). Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

> **Tip**
>
> Le nom de l’alarme ne peut contenir que des caractères UTF-8. Il ne peut pas contenir de caractères de contrôle ASCII.

18. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont correctes, et choisissez Create alarm (Créer une alerte).

Informations sur les alarmes pour les sources de données connectées

- Lorsqu'il CloudWatch évalue une alarme, il le fait toutes les minutes, même si la durée de l'alarme est supérieure à une minute. Pour que l’alarme fonctionne, la fonction Lambda doit pouvoir renvoyer une liste d’horodatages commençant à n’importe quelle minute, et pas seulement à des multiples de la durée de la période. Ces horodatages doivent être espacés d’une longueur de période.

  Par conséquent, si la source de données interrogée par le Lambda ne peut renvoyer que des horodatages multiples de la durée de la période, la fonction doit « rééchantillonner » les données extraites pour qu’elles correspondent aux horodatages attendus par la requête `GetMetricData`.

Par exemple, une alarme avec une période de cinq minutes est évaluée toutes les minutes à l'aide de fenêtres de cinq minutes décalées d'une minute à chaque fois. Dans ce cas :

- Pour l'évaluation de l'alarme à 12 h 15, des points de CloudWatch données sont attendus avec des horodatages de`12:00:00`, et. `12:05:00` `12:10:00`
- Ensuite, pour l'évaluation de l'alarme à 12 h 16, on CloudWatch attend des points de données horodatés de`12:01:00`, et. `12:06:00` `12:11:00`

- Lors de CloudWatch l'évaluation d'une alarme, tous les points de données renvoyés par la fonction Lambda qui ne correspondent pas aux horodatages attendus sont supprimés et l'alarme est évaluée en utilisant les points de données attendus restants. Par exemple, lorsque l'alarme est évaluée à `12:15:00`, il attend des données horodatées `12:00:00`, `12:05:00` et `12:10:00`. S'il reçoit des données horodatées de`12:00:00`,, et `12:05:00` `12:06:00``12:10:00`, les données sont supprimées `12:06:00` et CloudWatch évalue l'alarme en utilisant les autres horodatages.

  Ensuite, pour la prochaine évaluation à `12:16:00`, il attend des données horodatées `12:01:00`, `12:06:00` et `12:11:00`. S'il n'a que les données horodatées `12:00:00`, `12:05:00` et `12:10:00`, tous ces points de données sont ignorés à 12 h 16 et l'alarme passe à l'état correspondant à celui que vous avez spécifié pour l'alarme en ce qui concerne le traitement des données manquantes. Pour plus d'informations, consultez Évaluation d'une alerte.
- Nous vous recommandons de créer ces alarmes pour prendre des mesures lorsqu'elles passent à l'état `INSUFFICIENT_DATA`, car plusieurs cas d'utilisation d'une défaillance de la fonction Lambda feront passer l'alarme à `INSUFFICIENT_DATA` quelle que soit la manière dont vous l'avez configurée pour traiter les données manquantes.
- Si la fonction Lambda renvoie une erreur ou renvoie des données partielles :
  - En cas de problème d'autorisation lors de l'appel de la fonction Lambda, l'alarme commence à présenter des transitions de données manquantes conformément à la façon dont vous avez spécifié l'alarme pour traiter les données manquantes lors de sa création.
  - Si la fonction Lambda retourne `'StatusCode' = 'PartialData'`, l'évaluation de l'alarme échoue et l'alarme passe à `INSUFFICIENT_DATA` au bout de trois tentatives, ce qui prend environ trois minutes.
  - Toute autre erreur provenant de la fonction Lambda entraîne le passage de l'alarme à `INSUFFICIENT_DATA`.
- Si la métrique demandée par la fonction Lambda présente un retard tel que le dernier point de données est toujours manquant, vous devez utiliser une solution de contournement. Vous

pouvez créer une alarme M sur N ou augmenter la période d'évaluation de l'alarme. Pour plus d'informations sur les alarmes M sur N, veuillez consulter Évaluation d'une alerte.

Création d'une CloudWatch alarme basée sur la détection d'anomalies

Vous pouvez créer une alarme basée sur la détection d' CloudWatch anomalies, qui analyse les données métriques passées et crée un modèle des valeurs attendues. Les valeurs attendues prennent en compte les modèles classiques horaires, quotidiens et hebdomadaires dans la métrique.

Vous définissez une valeur pour le seuil de détection des anomalies et vous CloudWatch utilisez ce seuil avec le modèle pour déterminer la plage de valeurs « normale » de la métrique. Une valeur plus élevée pour le seuil produit une bande plus épaisse de valeurs « normales ».

Vous pouvez choisir si l'alerte est déclenchée lorsque la valeur de la métrique dépasse le groupe de valeurs attendues, se situe sous le groupe ou se trouve être supérieure ou inférieure au groupe.

Vous pouvez également créer des alertes de détection d'anomalies sur des métriques uniques et les sorties des expressions mathématiques de métrique. Vous pouvez utiliser ces expressions pour créer des graphiques qui visualisent les canaux de détection d'anomalies.

Dans un compte configuré en tant que compte de surveillance pour l'observabilité CloudWatch entre comptes, vous pouvez créer des détecteurs d'anomalies sur les métriques des comptes sources en plus des métriques du compte de surveillance.

Pour plus d'informations, consultez Utilisation de la détection des CloudWatch anomalies.

> **Note**
>
> Si vous utilisez déjà la détection d'anomalies à des fins de visualisation pour une métrique dans la console Metrics et que vous créez une alerte de détection d'anomalies pour cette même métrique, le seuil que vous définissez pour l'alerte ne change pas le seuil que vous avez déjà défini pour la visualisation. Pour plus d'informations, consultez Création d'un graphique.

Pour créer une alerte basée sur la détection d'anomalies

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation, choisissez Alarms (alertes), All alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Select Metric (Sélectionner une métrique).
5. Effectuez l'une des actions suivantes :
   - Choisissez l'espace de noms de service qui contient votre métrique, puis continuez à choisir les options au fur et à mesure qu'elles apparaissent pour réduire vos possibilités. Lorsqu'une liste de métriques apparaît, cochez la case qui se trouve à côté de votre métrique.
   - Dans la zone de recherche, tapez le nom d'une métrique, d'une dimension ou d'un ID de ressource. Sélectionnez l'un des résultats, puis continuez à choisir les options au fur et à mesure qu'elles apparaissent jusqu'à ce qu'une liste de métriques apparaisse. Cochez la case qui se trouve à côté de votre métrique.
6. Choisissez Metric graphié.

   a. (Facultatif) Pour Statistiques, choisissez le menu déroulant, puis sélectionnez l'une des statistiques ou percentiles prédéfinis. Vous pouvez utiliser le champ de recherche de la liste déroulante pour spécifier un centile personnalisé, tel que **`p95.45`**.

   b. (Facultatif) Pour Période, choisissez le menu déroulant, puis sélectionnez l'une des périodes d'évaluation prédéfinies.

   > **Note**
   >
   > Lors de CloudWatch l'évaluation de votre alarme, il agrège la période en un seul point de données. Pour une alerte de détection d'anomalie, la période d'évaluation doit être d'une minute ou plus.

7. Choisissez Suivant.
8. Sous Conditions, spécifiez les éléments suivants :

   a. Choisissez la Anomaly detection (détection d'anomalies).

   Si le modèle de cette métrique et de cette statistique existe déjà, CloudWatch affiche un aperçu de la bande de détection des anomalies dans le graphique en haut de l'écran. Une fois que vous avez créé votre alarme, l'affichage du groupe de détection d'anomalies réelle peut prendre jusqu'à 15 minutes. Avant cela, le groupe que vous voyez est une approximation du groupe de détection d'anomalies.

> **Tip**
>
> Pour afficher le graphique pour une période plus longue, choisissez Edit (Modifier) en haut à droite de la page.

Si le modèle de cette métrique et de cette statistique n'existe pas déjà, CloudWatch génère la bande de détection des anomalies une fois que vous avez fini de créer votre alarme. Pour les nouveaux modèles, l'affichage du groupe de détection d'anomalies réel peut prendre jusqu'à 3 heures. L'entraînement du nouveau modèle peut prendre jusqu'à deux semaines, de sorte que le groupe de détection des anomalies affiche des valeurs attendues plus précises.

b. Quel que soit l'emplacement de la ***métrique***, précisez l'heure de déclenchement de l'alerte. Par exemple, lorsque la métrique est supérieure, inférieure ou se trouve hors de la bande (dans chacune des directions).

c. Pour Anomaly detection threshold (Seuil de détection des anomalies), choisissez le nombre à utiliser pour le seuil de détection des anomalies. Un nombre plus élevé crée un groupe plus épais de valeurs « normales » qui est plus tolérant aux changements de métrique. Un nombre moins élevé crée un groupe plus fin qui passera à l'état `ALARM` avec des déviations de métriques plus petites. Le nombre ne doit pas nécessairement être un nombre entier.

d. 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 Évaluation d'une alerte.

e. 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, consultez Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.

f. Si l'alerte utilise un centile comme statistique surveillée, une zone Percentiles with low samples (Centiles avec exemples de bas niveau) s'affiche. Utilisez-la pour choisir si vous souhaitez évaluer ou ignorer les cas avec des taux d'échantillons faibles. Si vous

sélectionnez Ignore (ignorer : conserver l'état d'alerte), l'état actuel de l'alerte est toujours conservé lorsque la taille de l'échantillon est trop réduite. Pour plus d'informations, consultez . CloudWatch Alarmes basées sur les percentiles et échantillons de données faibles.

9. Choisissez Next (Suivant).
10. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alerte passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Pour envoyer plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, sélectionnez Add notification (Ajouter une notification).

    Sélectionnez Remove (Supprimer)si vous ne voulez pas que l'alerte envoie de notifications.
11. Vous pouvez configurer l'alarme pour effectuer des actions EC2 ou appeler une fonction Lambda lorsqu'elle change d'état, ou pour créer un Systems OpsItem Manager ou un incident lorsqu'elle passe en état ALARM. Pour ce faire, sélectionnez le bouton approprié, puis sélectionnez l'état d'alerte et l'action à effectuer.

    Si vous choisissez une fonction Lambda comme action d'alarme, vous spécifiez le nom de la fonction ou l'ARN, et vous pouvez éventuellement choisir une version spécifique de la fonction.

    Pour plus d'informations sur les actions de Systems Manager, consultez les sections Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

    > **Note**
    >
    > Pour créer une alerte qui exécute une action AWS Systems Manager Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager.
12. Choisissez Suivant.
13. Sous Add a description (Ajouter une description), saisissez un nom et une description pour l'alerte et choisissez Next (Suivant). Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

> **Tip**
>
> Le nom de l'alerte ne doit contenir que des caractères UTF-8 et 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).

Modification d'un modèle de détection d'anomalies

Après avoir créé une alerte, vous pouvez ajuster le modèle de détection des anomalies. Vous pouvez exclure certaines périodes d'utilisation dans la création du modèle. Il est essentiel d'exclure des données d'apprentissage les événements inhabituels tels que les pannes du système, les déploiements et les périodes de congés. Vous pouvez également spécifier si le modèle doit être ajusté aux modifications relatives à l'heure d'été.

Pour ajuster le modèle de détection d'anomalies pour une alerte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All alarms (Toutes les alertes).
3. Sélectionnez le nom de l'alerte. Si nécessaire, utilisez la zone de recherche pour trouver l'alerte.
4. Choisissez Analyser, Dans les métriques.
5. Dans la colonne Détails, choisissez ANOMALY_DETECTION_BAND, puis Modifier le modèle de détection d'anomalies.
6. Pour exclure une période dans la production du modèle, sélectionnez l'icône de calendrier par Date de fin. Ensuite, sélectionnez ou saisissez les jours et heures à exclure de la formation, puis sélectionnez Apply (Appliquer).
7. Si la métrique est sensible au changement d'heure, sélectionnez le fuseau horaire approprié dans la zone Metric timezone (Fuseau horaire de la métrique).
8. Sélectionnez Update (Mettre à jour).

Suppression d'un modèle de détection d'anomalies

L'utilisation de la détection d'anomalies pour une alerte augmente les frais . Selon les bonnes pratiques, si votre alarme n’a plus besoin d’un modèle de détection des anomalies, supprimez

d’abord l’alarme, puis le modèle. Lorsque les alarmes de détection d’anomalies sont évaluées, les détecteurs d’anomalies manquants sont créés en votre nom. Si vous supprimez le modèle sans supprimer l'alerte, l'alerte recrée automatiquement le modèle.

Pour supprimer une alerte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All Alarms (Toutes les alertes).
3. Sélectionnez le nom de l'alerte.
4. Sélectionnez Actions, Supprimer.
5. Dans la boîte de confirmation, choisissez Supprimer.

Pour supprimer un modèle de détection des anomalies qui a été utilisé pour une alarme

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Metrics (Métriques), puis choisissez All metrics (Toutes les métriques).
3. Choisissez Browse (Parcourir), puis sélectionnez la métrique qui inclut le modèle de détection des anomalies. Vous pouvez rechercher la métrique dans le champ de recherche ou la sélectionner parmi les options.
   - (Facultatif) Si vous utilisez l'interface d'origine, choisissez All metrics (Toutes les métriques), puis sélectionnez la métrique qui inclut le modèle de détection des anomalies. Vous pouvez rechercher la métrique dans le champ de recherche ou la sélectionner parmi les options.
4. Choisissez Graphed metrics (Graphique des métriques).
5. Dans l'onglet Graphed metrics (Métriques graphiques), sélectionnez le nom du modèle de détection des anomalies que vous voulez supprimer, puis cliquez sur Delete anomaly detection model (Supprimer le modèle de détection des anomalies).
   - (Facultatif) Si vous utilisez l'interface d'origine, sélectionnez Edit model (Modifier le modèle). Vous êtes redirigé vers un nouvel écran. Sur le nouvel écran, sélectionnez Delete model (Supprimer le modèle), puis cliquez sur Delete (Supprimer).

Créer des alertes sur les journaux

Les étapes décrites dans les sections suivantes expliquent comment créer des CloudWatch alarmes dans les journaux.

Création d'une CloudWatch alarme basée sur un filtre métrique de groupes de logs

Cette section décrit comment créer une alarme basée sur un filtre de métrique d'un groupe de journaux. Grâce aux filtres métriques, vous pouvez rechercher des termes et des modèles dans les données du journal au fur et à mesure de leur envoi CloudWatch. Pour plus d'informations, consultez la section Créer des métriques à partir d'événements de journal à l'aide de filtres dans le guide de l'utilisateur Amazon CloudWatch Logs. Avant de créer une alarme basée sur un filtre de métrique d'un groupe de journaux, vous devez effectuer les actions suivantes :

- Créez un groupe de journaux . Pour plus d'informations, consultez la section Utilisation des groupes de journaux et des flux de CloudWatch journaux dans le guide de l'utilisateur Amazon Logs.
- Créez un filtre de métrique. Pour plus d'informations, consultez la section Création d'un filtre métrique pour un groupe de CloudWatch journaux dans le guide de l'utilisateur Amazon Logs.

Pour créer une alarme basée sur un filtre de métrique d'un groupe de journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Logs (Journaux), puis Log groups (Groupes de journaux).
3. Sélectionnez le groupe de journaux qui comprend votre filtre de métrique.
4. Sélectionnez Metric filters (Filtres de métrique).
5. Dans l'onglet Metric filters (Filtres de métrique), cochez la case du filtre de métrique sur lequel vous voulez baser votre alarme.
6. Sélectionnez Créer une alerte.
7. (Facultatif) Sous Metric (Métrique), modifiez le champ Metric name (Nom de la métrique), Statistic (Statistique) et Period (Période).
8. Sous Conditions, spécifiez les éléments suivants :

    a. Pour Threshold type (Type de seuil), choisissez Static (Statique) ou Anomaly detection (Détection des anomalies).

    b. Pour Whenever, ***your-metric-name***c'est... , choisissez Plus grand, Plus grand/égal, inférieur/égal ou inférieur.

    c. Pour than . . . (à . . .), spécifiez un nombre pour votre valeur de seuil.

9. Sélectionnez Additional configuration (Configuration supplémentaire).

    a. Pour Data points to alarm (Points de données pour l'alarme), indiquez combien de points de données déclenchent le passage de votre alarme à l'état `ALARM`. Si vous spécifiez des valeurs correspondantes, votre alarme passe à l'état `ALARM` si ce nombre de périodes consécutives est dépassé. Pour créer une alarme M-sur-N, spécifiez un nombre pour la première valeur qui est inférieur au nombre que vous spécifiez pour la deuxième valeur. Pour plus d'informations, consultez la section Utilisation des CloudWatch alarmes Amazon.

    b. Pour Missing data treatment (Traitement des données manquantes), sélectionnez une option pour spécifier comment traiter les données manquantes lors de l'évaluation de votre alarme.

10. Choisissez Suivant.

11. Sous Notification, spécifiez une rubrique Amazon SNS qui doit recevoir une notification lorsque l'alarme passe à l'état `ALARM`, `OK` 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).

    b. (Facultatif) Pour ne pas envoyer de notifications, choisissez Remove (Supprimer).

12. Pour que l'alarme exécute des actions Auto Scaling, EC2, Lambda ou Systems Manager, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer. Si vous choisissez une fonction Lambda comme action d'alarme, vous spécifiez le nom de la fonction ou l'ARN, et vous pouvez éventuellement choisir une version spécifique de la fonction.

    Les alertes peuvent effectuer des actions du Systems Manager uniquement lorsqu'elles passent à l'état ALARM. Pour plus d'informations sur les actions de Systems Manager, voir Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

    > **Note**
    >
    > Pour créer une alerte qui exécute une action SSM Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager.

13. Choisissez Suivant.

14. Pour Name and description (Nom et description), saisissez un nom et une description pour votre alarme. Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est

affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

15. Pour Preview and create (Prévisualiser et créer), vérifiez que votre configuration est correcte, puis sélectionnez Create alarm (Créer l'alarme).

Combinaison d'alarmes

Vous pouvez ainsi combiner plusieurs alarmes en une seule alarme composite afin de créer un indicateur de santé résumé et agrégé pour l'ensemble d'une application ou d'un groupe de ressources. CloudWatch Les alarmes composites sont des alarmes qui déterminent leur état en surveillant les états des autres alarmes. Vous définissez des règles pour combiner l'état de ces alarmes surveillées en utilisant la logique booléenne.

Vous pouvez utiliser des alarmes composites pour réduire le bruit d'alarme en prenant des mesures uniquement à un niveau agrégé. Par exemple, vous pouvez créer une alarme composite pour envoyer une notification à l'équipe de votre serveur Web si une alarme liée à votre serveur Web se déclenche. Lorsque l'une de ces alarmes passe à l'état ALARM, l'alarme composite passe elle-même à l'état ALARM et envoie une notification à votre équipe. Si d'autres alarmes liées à votre serveur Web passent également à l'état ALARM, votre équipe ne sera pas surchargée de nouvelles notifications puisque l'alarme composite l'a déjà informée de la situation existante.

Vous pouvez également utiliser des alarmes composites pour créer des conditions d'alarme complexes et prendre des mesures uniquement lorsque de nombreuses conditions différentes sont remplies. Par exemple, vous pouvez créer une alarme composite qui combine une alarme de processeur et une alarme de mémoire, et qui n'avertirait votre équipe que si les alarmes du processeur et de la mémoire se sont déclenchées.

Utilisation d'alarmes composites

Lorsque vous utilisez des alarmes composites, deux options s'offrent à vous :

- configurer les actions que vous souhaitez effectuer uniquement au niveau des alarmes composites et créer les alarmes surveillées sous-jacentes sans actions ;
- configurer un ensemble d'actions différent au niveau de l'alarme composite. Par exemple, les actions d'alarme composites pourraient impliquer une autre équipe en cas de problème généralisé.

Les alarmes composites ne peuvent effectuer que les actions suivantes :

- Notifier une rubrique Amazon SNS
- Invoquer des fonctions Lambda
- Création OpsItems dans le Systems Manager Ops Center
- Création d'incidents dans Systems Manager Incident Manager

> **Note**
>
> Toutes les alarmes sous-jacentes de votre alarme composite doivent être dans le même compte et la même région que votre alarme composite. Toutefois, si vous configurez une alarme composite dans un CloudWatch compte de surveillance de l'observabilité entre comptes, les alarmes sous-jacentes peuvent surveiller les métriques dans différents comptes sources et dans le compte de surveillance lui-même. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.
> Une alerte composite unique peut surveiller 100 alertes sous-jacentes, et 150 alertes composites peuvent surveiller une seule alerte sous-jacente.

Expressions de règle

Toutes les alertes composites contiennent des expressions normées. Les expressions normées indiquent aux alertes composites les autres alertes à surveiller et à partir desquelles elles doivent déterminer leur état. Une expression normée peut faire référence à la fois aux alertes de métrique et à d'autres alertes composites. Lorsque vous référencez une alerte dans une expression normée, vous désignez une fonction de l'alerte qui détermine dans quel état l'alerte se trouvera parmi les trois états suivants :

- ALARM

  ALARM ("alarm-name or alarm-ARN") est défini sur TRUE si l'alarme est à l'état ALARM.
- OK

  OK ("alarm-name or alarm-ARN") est défini sur TRUE si l'alarme est à l'état OK.
- INSUFFICIENT_DATA

  INSUFFICIENT_DATA (“alarm-name or alarm-ARN") est défini sur TRUE si l'alarme nommée est à l'état INSUFFICIENT_DATA.

> **Note**
>
> TRUE a toujours pour valeur TRUE, et FALSE a toujours pour valeur FALSE.

Exemples d'expressions

Le paramètre de la demande `AlarmRule` prend en charge l'utilisation des opérateurs logiques `AND`, `OR`, et `NOT`, afin que vous puissiez combiner plusieurs fonctions en une seule expression. Les exemples d'expressions suivants montrent comment configurer les alertes sous-jacentes dans votre alerte composite :

- `ALARM(CPUUtilizationTooHigh) AND ALARM(DiskReadOpsTooHigh)`

  L'expression indique que l'alerte composite passe à l'état `ALARM` uniquement si `CPUUtilizationTooHigh` et`DiskReadOpsTooHigh` sont à l'état `ALARM`.

- `ALARM(CPUUtilizationTooHigh) AND NOT ALARM(DeploymentInProgress)`

  L'expression indique que l'alerte composite passe à l'état `ALARM` si `CPUUtilizationTooHigh` est à l'état `ALARM` et si `DeploymentInProgress` n'est pas à l'état `ALARM`. Il s'agit d'un exemple d'alerte composite qui réduit le bruit d'alerte pendant une fenêtre de déploiement.

- `(ALARM(CPUUtilizationTooHigh) OR ALARM(DiskReadOpsTooHigh)) AND OK(NetworkOutTooHigh)`

  L'expression indique que l'alerte composite passe à l'état `ALARM` si `(ALARM(CPUUtilizationTooHigh)` ou `(DiskReadOpsTooHigh)` est à l'état `ALARM` et si `(NetworkOutTooHigh)` est à l'état `OK`. Voici un exemple d'alerte composite qui réduit le bruit d'alerte en ne vous envoyant pas de notifications lorsque l'une des alertes sous-jacentes n'est pas à l'état `ALARM` pendant qu'un problème de réseau se produit.

Rubriques

- Créer une alerte composite
- Supprimer des actions d'alarme composites

Créer une alerte composite

Les étapes décrites dans cette section expliquent comment utiliser la CloudWatch console pour créer une alarme composite. Vous pouvez également utiliser l'API ou AWS CLI créer une alarme composite. Pour plus d'informations, consultez PutCompositeAlarmou put-composite-alarm

Pour créer une alerte composite

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), puis All alarms (Toutes les alertes).
3. Dans la liste des alertes, activez les cases à cocher en regard de chacune des alertes existantes que vous souhaitez référencer dans votre expression normée, ensuite Créer une alerte composite.
4. Sous Specify composite alarm conditions (Spécifier des conditions d'alerte composite), spécifiez l'expression de règle pour votre nouvelle alerte composite.

   > **Note**
   >
   > Automatiquement, les alertes que vous avez sélectionnées dans la liste des alertes sont répertoriées dans la zone Conditions. Par défaut, la fonction `ALARM` a été désignée pour chacune de vos alertes, et chacune de vos alertes est associée par l'opérateur logique `OR`.

   Vous pouvez utiliser les sous-étapes suivantes pour modifier l'expression de votre règle :

   a. Vous pouvez modifier l'état requis pour chacune de vos alertes de `ALARM` à `OK` ou `INSUFFICENT_DATA`.

   b. Vous pouvez modifier l'opérateur logique dans l'expression de votre règle de `OR` à `AND` ou `NOT`, et vous pouvez ajouter des parenthèses pour regrouper vos fonctions.

   c. Vous pouvez inclure d'autres alertes dans votre expression de règle ou supprimer des alertes de votre expression de règle.

   Exemple : expression de règle avec conditions

```
(ALARM("CPUUtilizationTooHigh") OR
ALARM("DiskReadOpsTooHigh")) AND
```

```
OK("NetworkOutTooHigh")
```

Dans l'exemple d'expression de règle où l'alarme composite se déclenche `ALARM` lorsque ALARM (UtilizationTooHigh« CPU ») ou ALARM (» DiskReadOpsTooHigh «) est activé en `ALARM` même temps que OK (» NetworkOutTooHigh «)`OK`.

5. Lorsque vous avez terminé, choisissez Next (Suivant).
6. Sous Configuration d'actions, vous pouvez choisir parmi les éléments suivants :

   Pour Notification

   - Sélectionnez une rubrique SNS existante,Création d'une nouvelle rubrique SNS, ou Utiliser un ARN de rubrique pour définir la rubrique SNS qui recevra la notification.
   - Choisissez Add notification (Ajouter une notification) pour envoyer plusieurs notifications pour le même état d'alerte ou pour différents états d'alerte.
   - Choisissez Remove (Supprimer) pour empêcher votre alerte d'envoyer des notifications ou de prendre des mesures.

   (Facultatif) Pour que l'alarme invoque une fonction Lambda lorsqu'elle change d'état, choisissez Ajouter une action Lambda. Spécifiez ensuite le nom de la fonction ou l'ARN, et choisissez éventuellement une version spécifique de la fonction.

   Pour Systems Manager action (Action du gestionnaire de systèmes)

   - Choisissez Add Systems Manager action (Ajouter une action Systems Manager), afin que votre alerte puisse effectuer une action SSM quand elle passe à l'état ALARM.

   Pour en savoir plus sur les actions de Systems Manager, consultez les sections Configuration CloudWatch pour créer à OpsItems partir d'alarmes dans le Guide de AWS Systems Manager l'utilisateur et Création d'incidents dans le Guide de l'utilisateur d'Incident Manager. Pour créer une alerte qui exécute une action SSM Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager dans le guide de l'utilisateur d'Incident Manager.
7. Lorsque vous avez terminé, choisissez Next (Suivant).
8. Sous Add name and description (Ajoutez un nom et une description), entrez un nom d'alerte et en option une description de votre nouvelle alerte composite. Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut

inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

9. Lorsque vous avez terminé, choisissez Next (Suivant).
10. Sous Aperçu et création, confirmez vos informations, puis choisissez Création d'une alerte composite.

> **Note**
>
> Vous pouvez créer un cycle d'alertes composites, dans lequel deux alertes dépendent l'une de l'autre. Si vous vous trouvez dans ce scénario, vos alertes composites cessent d'être évaluées et vous ne pouvez pas les supprimer car elles sont dépendantes les unes des autres. Le moyen le plus simple de rompre le cycle de dépendance entre vos alertes composites est de modifier la fonction `AlarmRule` dans l'une d'elles pour `False`.

Supprimer des actions d'alarme composites

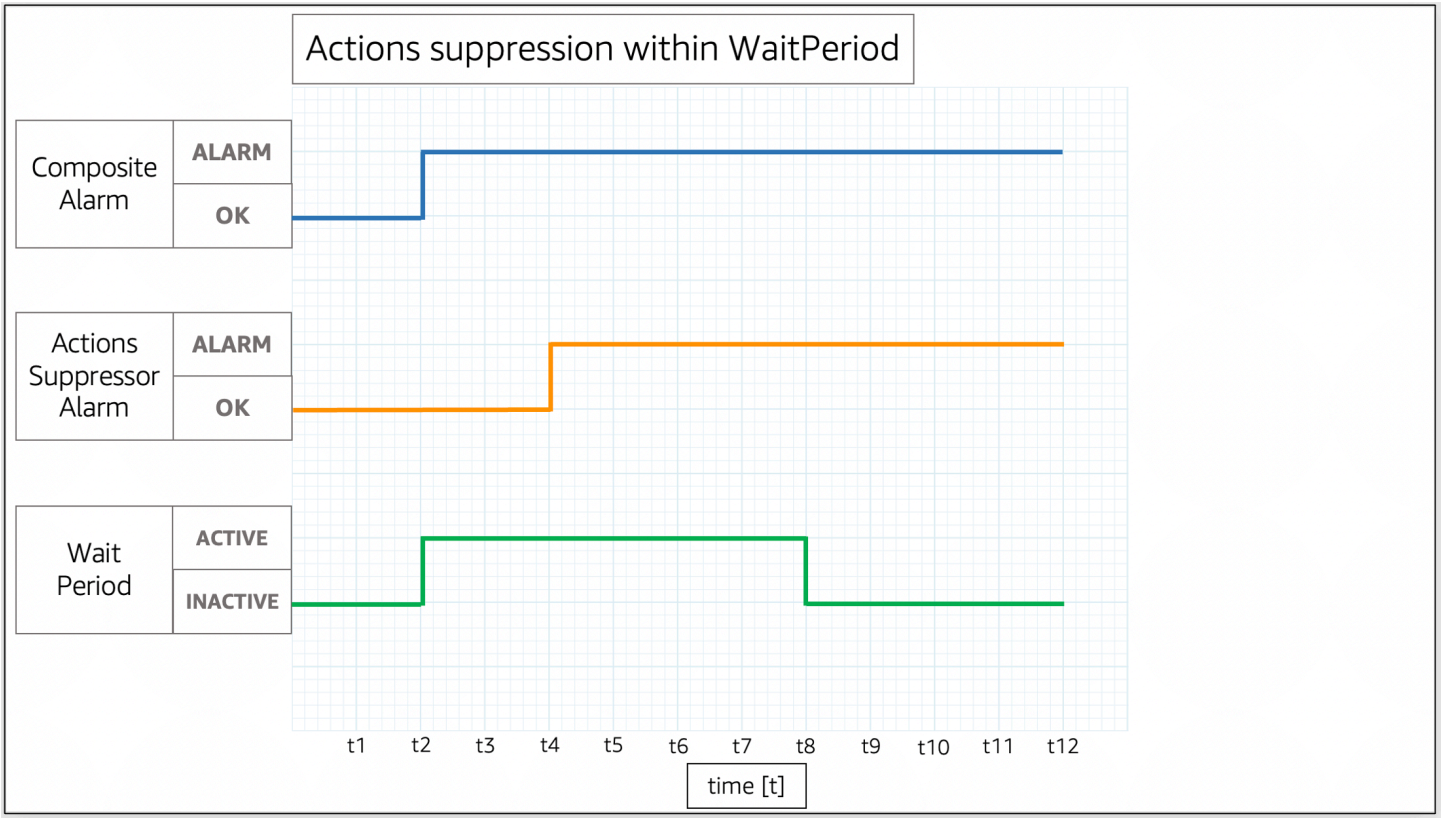
Étant donné que les alarmes composites vous permettent d'obtenir une vue agrégée de votre état de santé sur plusieurs alarmes, il existe des situations courantes où l'on s'attend à ce que ces alarmes se déclenchent. Par exemple, pendant une période de maintenance de votre application ou lorsque vous enquêtez sur un incident en cours. Dans de telles situations, vous souhaiterez peut-être supprimer les actions de vos alarmes composites, afin d'éviter les notifications indésirables ou la création de nouveaux tickets d'incident

Avec la suppression d'action d'alerte composite, vous définissez les alertes comme des alertes de suppression. Les alertes de suppression empêchent les alertes composites de prendre des mesures. Par exemple, vous pouvez spécifier une alerte de suppression qui représente l'état d'une ressource de prise en charge. Si la ressource de prise en charge est indisponible, l'alerte de suppression empêche l'alerte composite d'envoyer des notifications. La suppression de l'action d'alerte composite vous aide à réduire le bruit des alertes, de sorte que vous passez moins de temps à gérer vos alertes et plus de temps à vous concentrer sur vos opérations.

Vous spécifiez des alertes de suppression lorsque vous configurez des alertes composites. Toute alerte peut fonctionner comme une alerte de suppression. Lorsqu'une alerte de suppression passe de l'état `OK` à `ALARM`, son alerte composite cesse de prendre des mesures. Lorsqu'une alerte de suppression passe de l'état `ALARM` à `OK`, son alerte composite reprend ses actions.

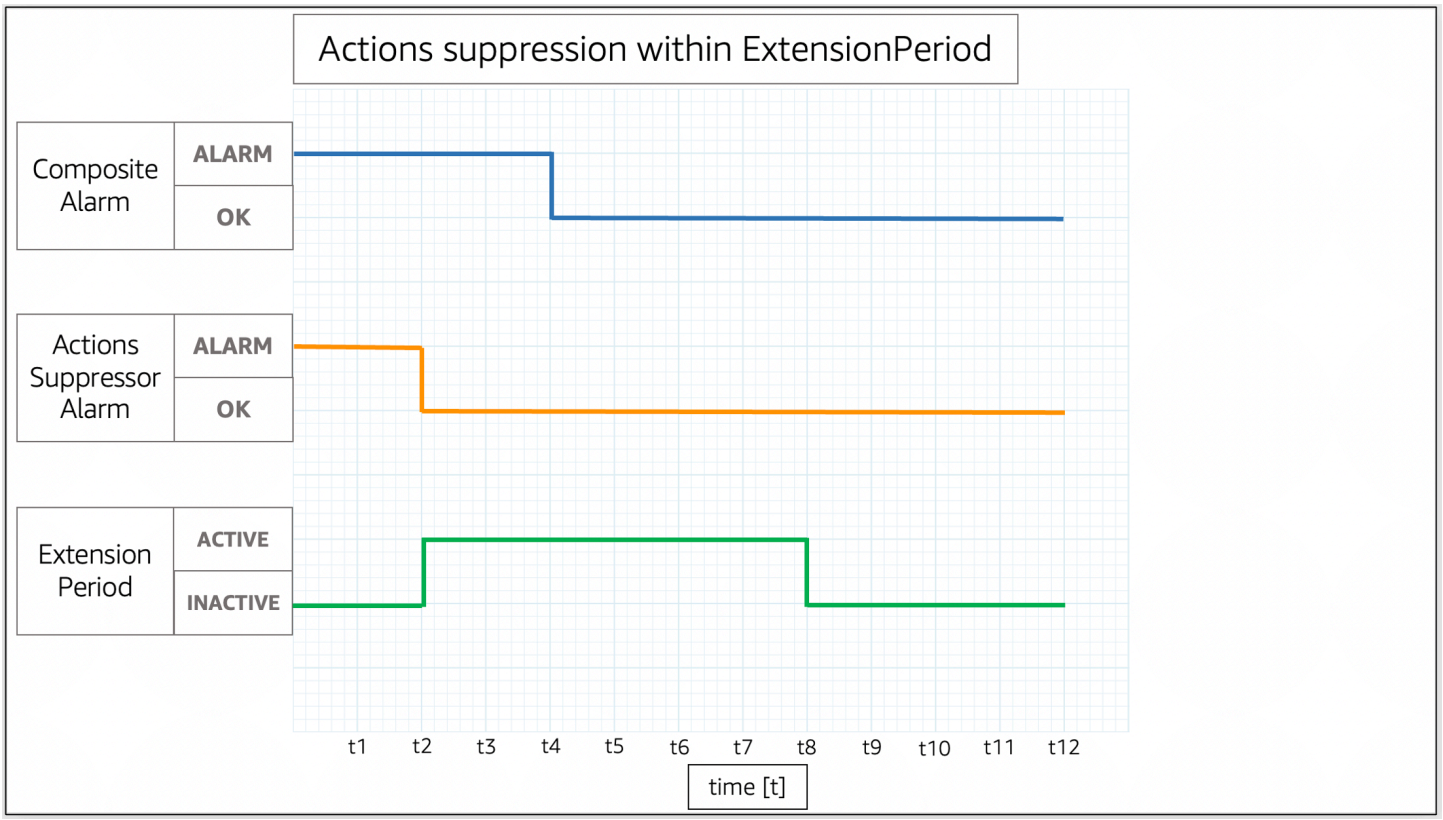
`WaitPeriod` et `ExtensionPeriod`

Lorsque vous spécifiez une alerte de suppression, vous définissez les paramètres `WaitPeriod` et `ExtensionPeriod`. Ces paramètres empêchent les alertes composites de prendre des mesures inattendues lorsque les alertes de suppression changent d'état. Utiliser `WaitPeriod` pour compenser les retards qui peuvent survenir lorsqu'une alerte de suppression passe de `OK` à `ALARM`. Par exemple, si l'alerte de suppression passe de `OK` à `ALARM` dans les 60 secondes, réglez `WaitPeriod` sur 60 secondes.



Dans l'image, l'alerte composite passe de `OK` à `ALARM` à t2. Une `WaitPeriod` commence à t2 et se termine à t8. Cela donne à l'alerte de suppression le temps de passer de`OK` à `ALARM` à t4 avant de supprimer les actions de l'alerte composite lorsque la `WaitPeriod` expire à t8.

Utilisez `ExtensionPeriod` pour compenser les retards pouvant survenir lorsqu'une alerte composite passe à `OK` suite à une alerte de suppression passant à `OK`. Par exemple, si une alerte composite passe à `OK` dans les 60 secondes suivant le passage de l'alerte de suppression à `OK`, réglez `ExtensionPeriod` à 60 secondes.



Dans l'image, l'alerte de suppression passe de ALARM à OK à t2. Une `ExtensionPeriod` commence à t2 et se termine à t8. Cela donne à l'alerte composite l'heure de changement de ALARM à OK avant que la`ExtensionPeriod` expire à t8.

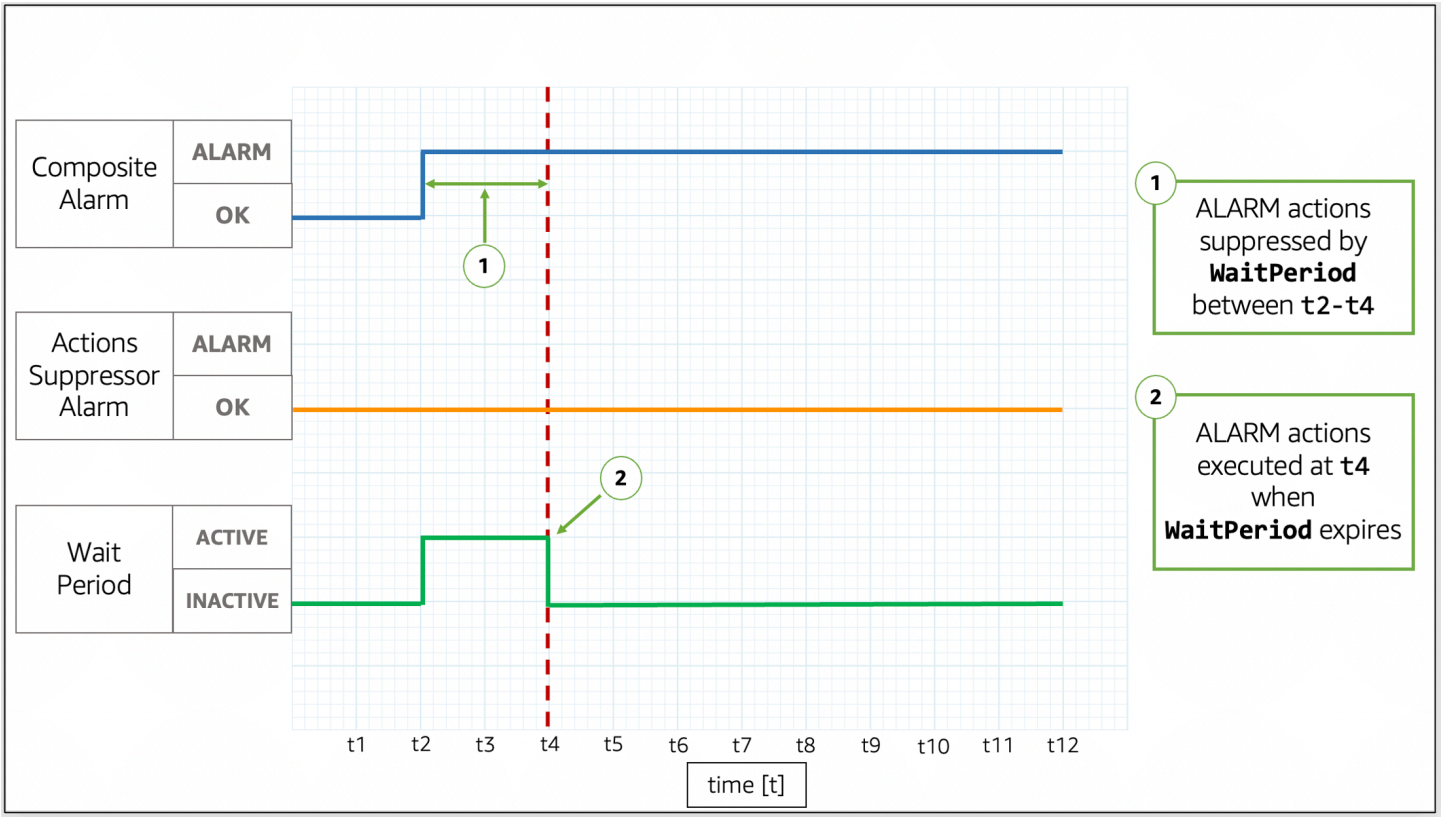
Les alertes composites ne prennent aucune mesure lorsque `WaitPeriod` et `ExtensionPeriod` deviennent actives. Les alertes composites prennent des mesures en fonction de leur état actuel lorsque `ExtensionPeriod` et `WaitPeriod` deviennent inactives. Nous vous recommandons de définir la valeur de chaque paramètre sur 60 secondes, car les alarmes métriques CloudWatch sont évaluées toutes les minutes. Vous pouvez définir les paramètres sur n'importe quel entier en quelques secondes.

Les exemples suivants décrivent plus en détail comment `WaitPeriod` et `ExtensionPeriod` empêchent les alertes composites de prendre des mesures inattendues.

> **Note**
>
> Dans les exemples suivants, `WaitPeriod` est configuré comme 2 unités de temps, et `ExtensionPeriod` est configuré comme 3 unités de temps.

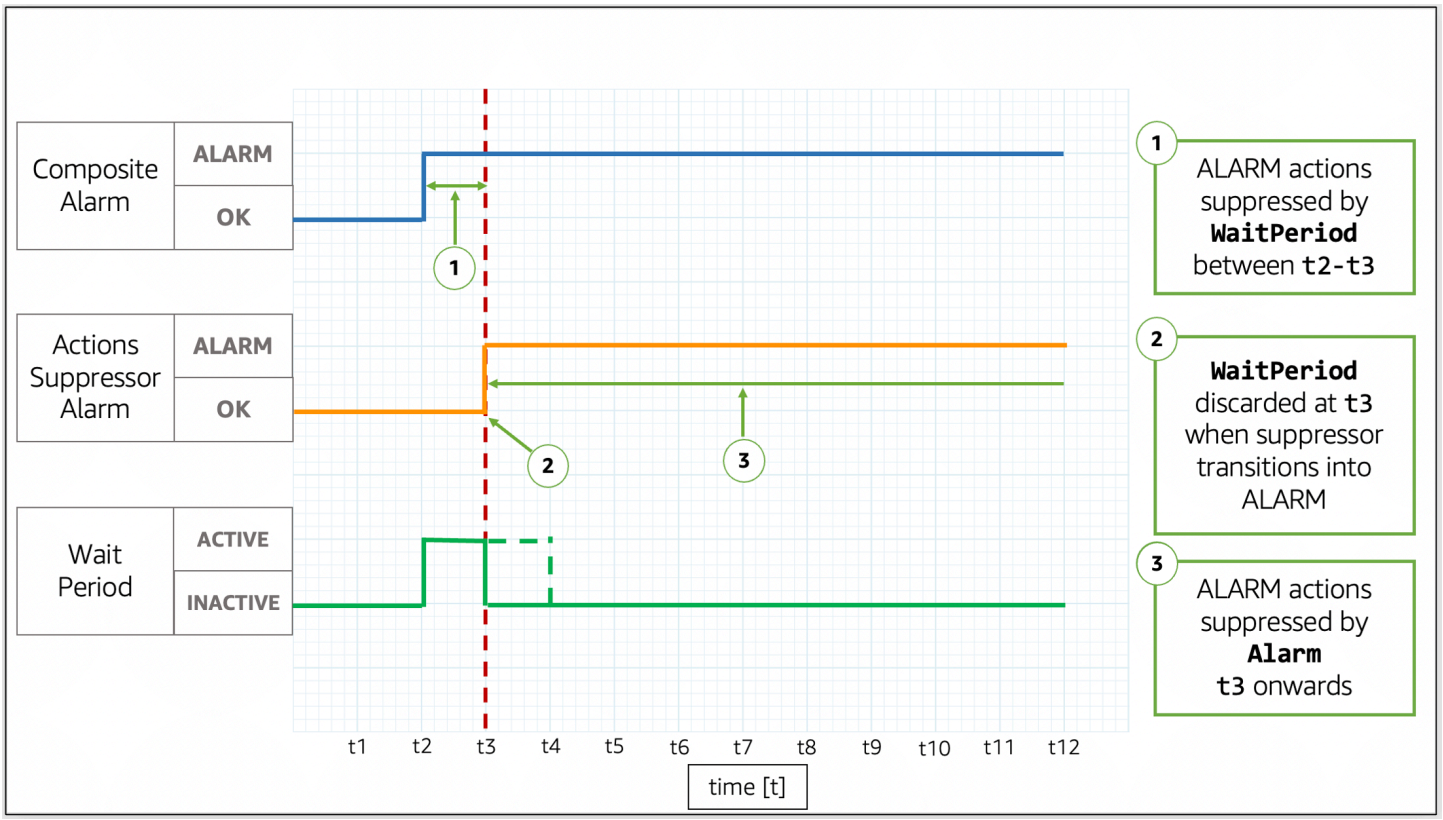
Exemples

Exemple 1 : Les actions ne sont pas supprimées après **`WaitPeriod`**



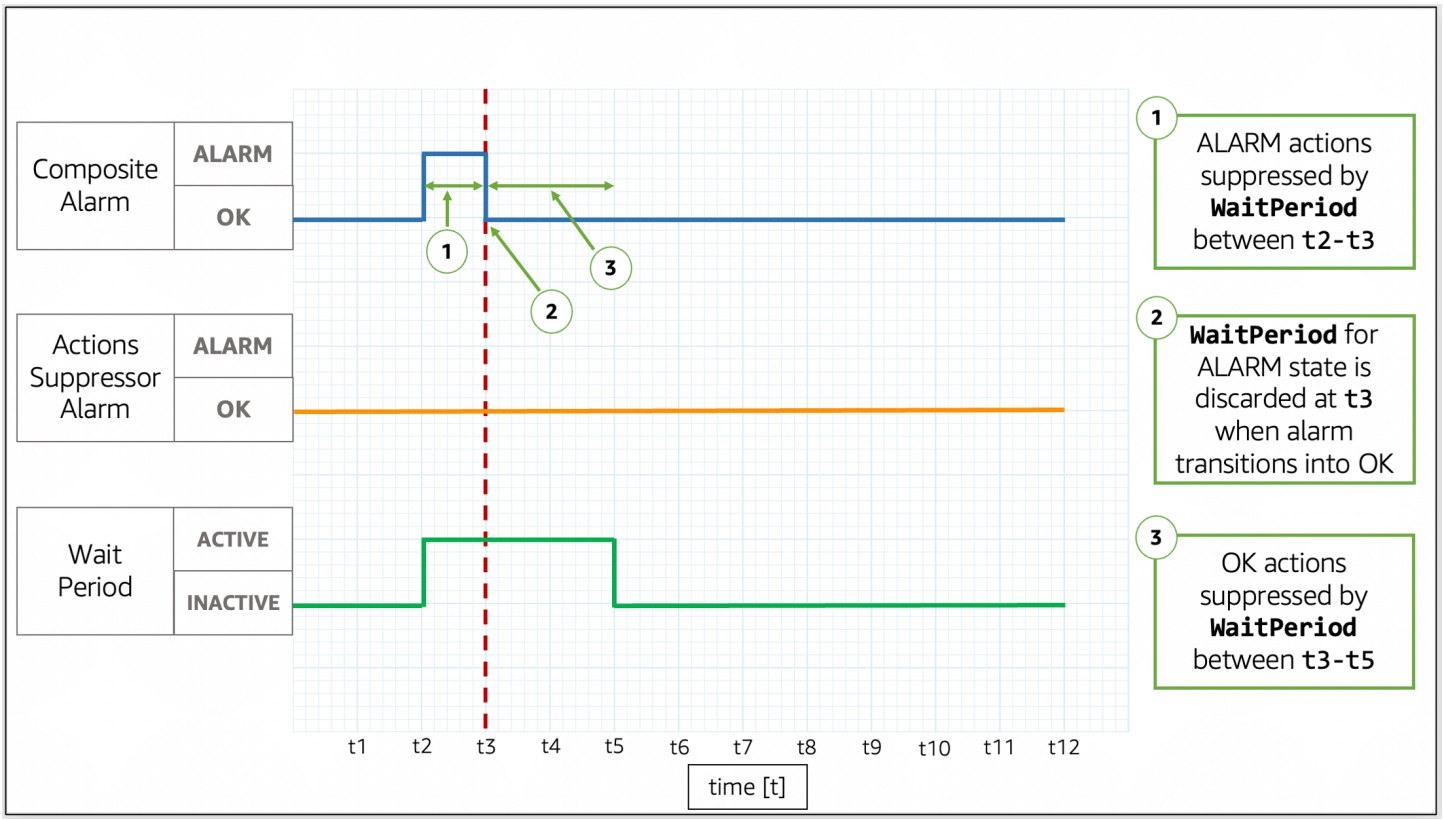
Dans l'image, l'état de l'alerte composite passe de `OK` à `ALARM` à t2. Une `WaitPeriod` commence à t2 et se termine à t4, ce qui peut empêcher l'alerte composite de prendre des mesures. Après que `WaitPeriod` expire à t4, l'alerte composite prend des mesures car l'alerte de suppression est toujours `OK`.

Exemple 2 : Les actions sont supprimées par une alerte avant que **`WaitPeriod`** expire



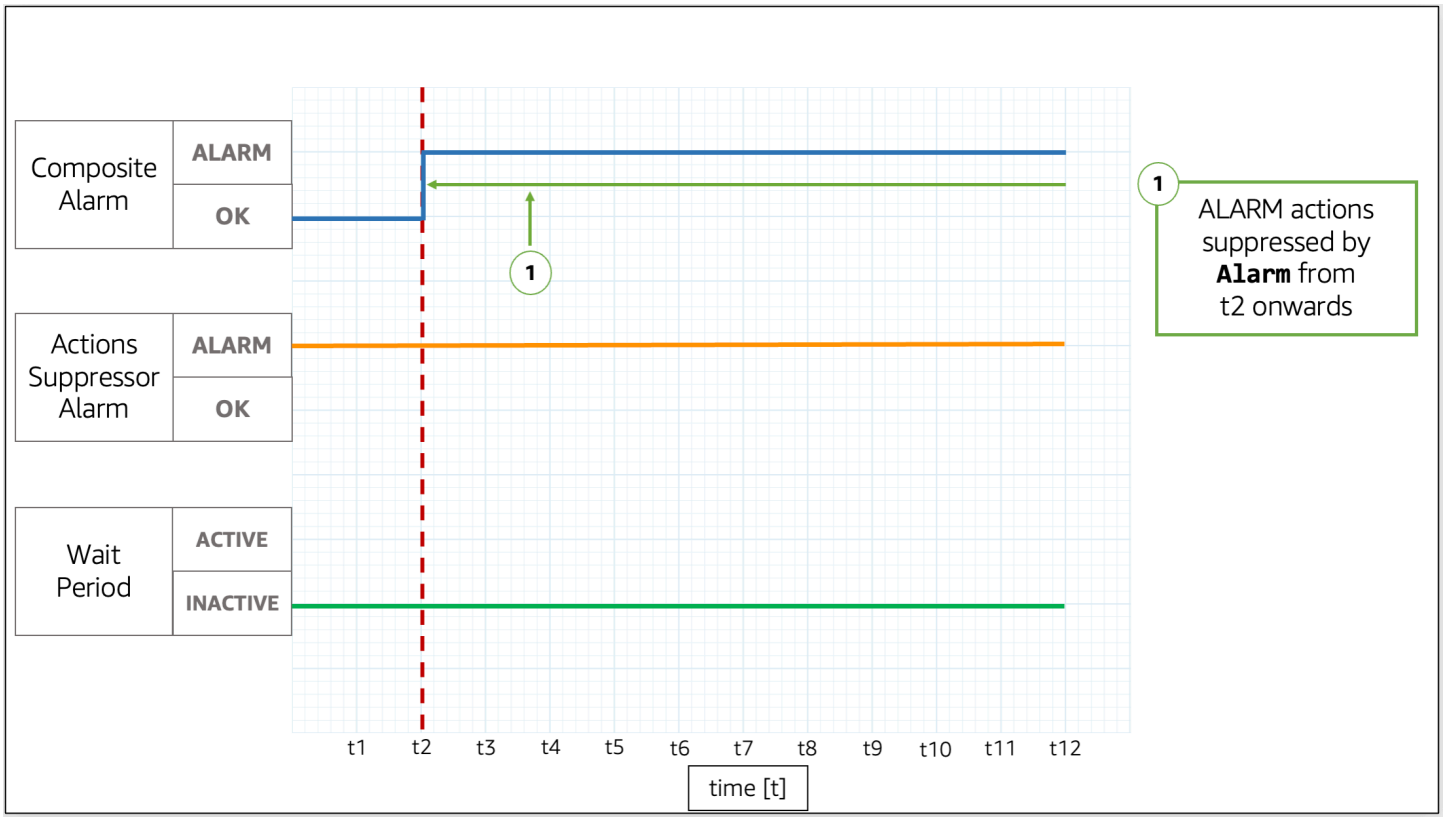
Dans l'image, l'état de l'alerte composite passe de`OK` à `ALARM` à t2. Une `WaitPeriod` commence à t2 et se termine à t4. Cela donne à l'alerte de suppression le temps de passer de `OK` à `ALARM` à t3. Parce que l'alerte de suppression passe de `OK` à `ALARM` à t3, la `WaitPeriod` qui a commencé à t2 est ignorée, et l'alerte de suppression empêche désormais l'alerte composite de prendre des mesures.

Exemple 3 : Transition d'état lorsque des actions sont supprimées par **`WaitPeriod`**



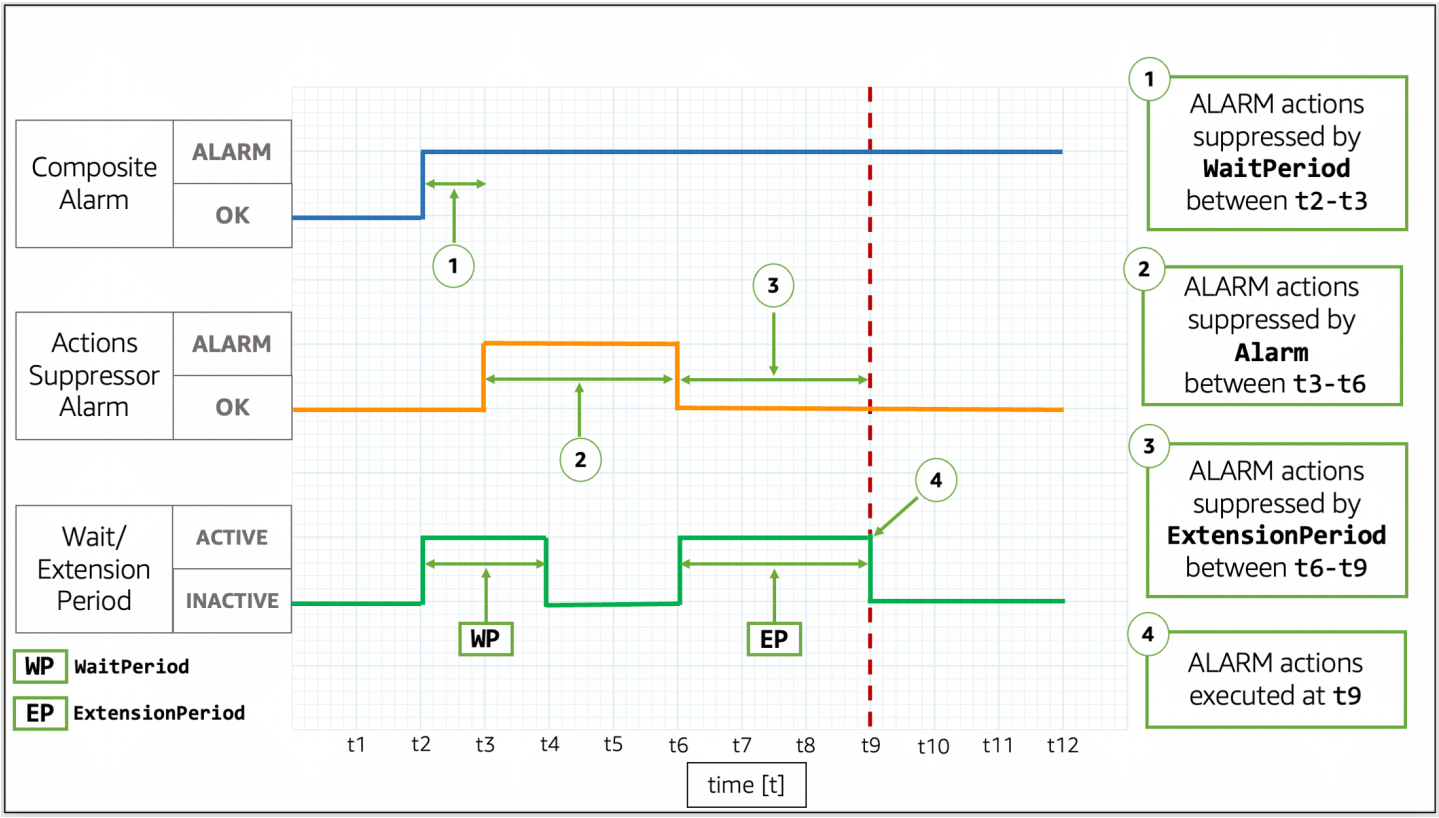
Dans l'image, l'état de l'alerte composite passe de`OK` à `ALARM` à t2. Une `WaitPeriod` commence à t2 et se termine à t4. Cela donne à l'alerte de suppression le temps de changer d'état. L'alerte composite revient à `OK` à t3, donc `WaitPeriod` qui a commencé à t2 est écartée. Une nouvelle `WaitPeriod` commence à t3 et se termine à t5. Lorsque la nouvelle `WaitPeriod` expire à t5, l'alerte composite prend ses mesures.

Exemple 4 : transition d'état lorsque des actions sont supprimées par une alerte



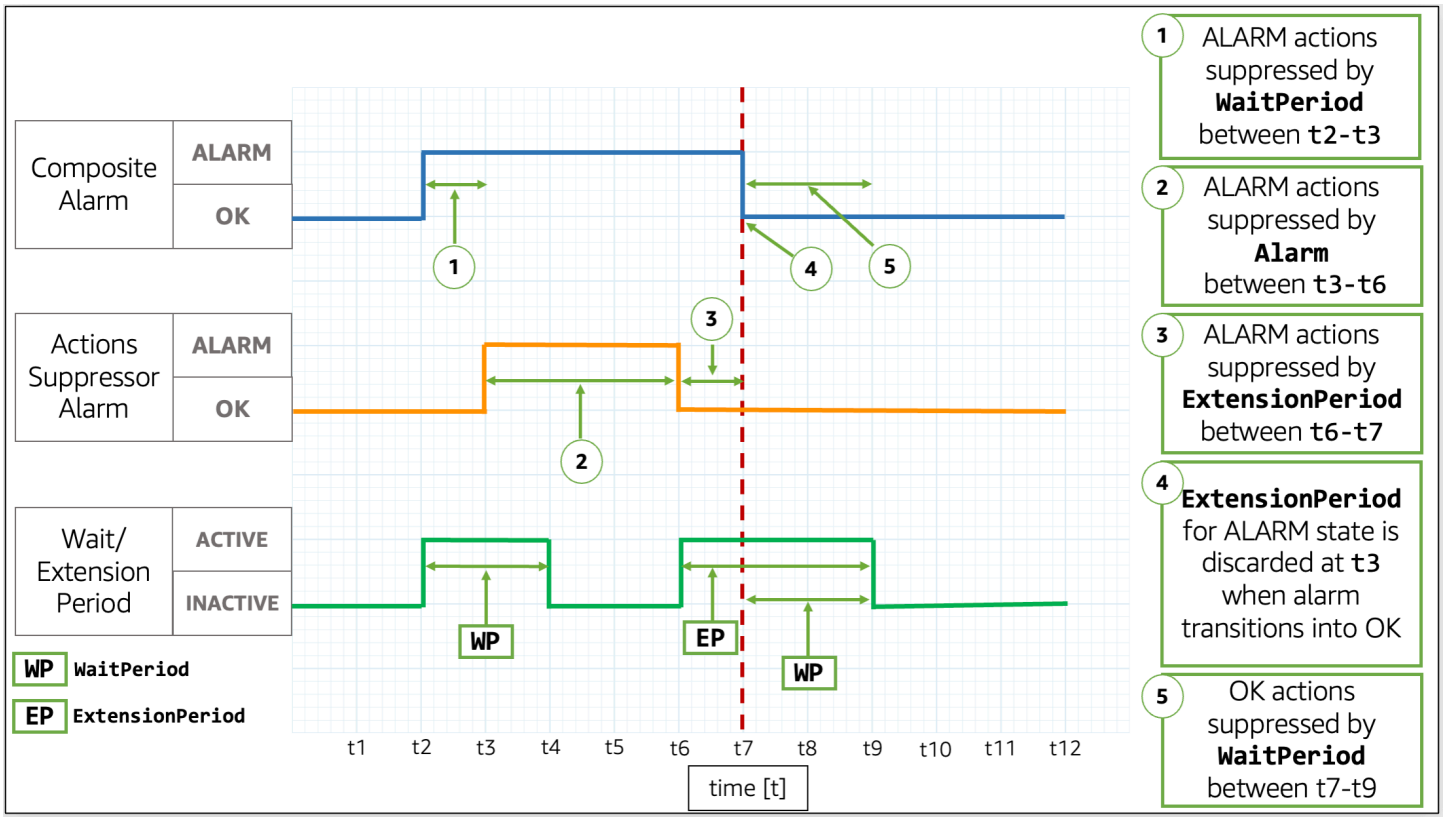
Dans l'image, l'état de l'alerte composite passe de `OK` à `ALARM` à t2. L'alerte de suppression est déjà activée `ALARM`. L'alerte de suppression empêche l'alerte composite de prendre des mesures.

Exemple 5 : Les actions ne sont pas supprimées après **`ExtensionPeriod`**



Dans l'image, l'état de l'alerte composite passe de`OK` à `ALARM` à t2. Une `WaitPeriod` commence à t2 et se termine à t4. Cela donne à l'alerte de suppression le temps de passer de `OK` à `ALARM` à t3 avant de supprimer les actions de l'alerte composite jusqu'à t6. Parce que l'alerte de suppression passe de `OK` à `ALARM` à t3, la `WaitPeriod` qui a commencé à t2 est écartée. À t6, l'alerte de suppression passe à `OK`. Une `ExtensionPeriod` commence à t6 et se termine à t9. Lorsque la nouvelle `ExtensionPeriod` expire, l'alerte composite prend ses mesures.

Exemple 6 : Transition d'état lorsque des actions sont supprimées par **`ExtensionPeriod`**



Dans l'image, l'état de l'alerte composite passe de`OK` à `ALARM` à t2. Une `WaitPeriod` commence à t2 et se termine à t4. Cela donne à l'alerte de suppression le temps de passer de `OK` à `ALARM` à t3 avant de supprimer les actions de l'alerte composite jusqu'à t6. Parce que l'alerte de suppression passe de `OK` à `ALARM` à t3, la `WaitPeriod` qui a commencé à t2 est écartée. À t6, l'alerte de suppression revient à `OK`. Une `ExtensionPeriod` commence à t6 et se termine à t9. Lorsque l'alerte composite redevient `OK` à t7, la `ExtensionPeriod` est supprimée, et une nouvelle `WaitPeriod` commence à t7 et se termine à t9.

> **Tip**
>
> Si vous remplacez l'alerte de suppression d'action, toute `WaitPeriod` ou `ExtensionPeriod` active est éliminée.

Agir en cas de changement d'alerte

CloudWatch peut informer les utilisateurs de deux types de changements d'alarme : lorsqu'une alarme change d'état et lorsque la configuration d'une alarme est mise à jour.

Lorsqu'une alerte est évaluée, elle peut passer d'un état à un autre, comme ALARM, OK ou INSUFICIENT_DATA. Ces changements d'état d'alerte peuvent signaler un éventuel incident, un retour à la normale ou l'indisponibilité d'une métrique. Dans de tels cas, vous devrez peut-être contacter ou informer les utilisateurs à l'aide de l'une des options suivantes :

- Vous pouvez configurer l'alerte pour envoyer une notification à une rubrique SNS dans le cadre des actions de l'alerte. Une rubrique SNS peut ensuite être configurée pour la messagerie application-to-application (A2A) ainsi que pour les notifications application-to-person (A2P), y compris les canaux tels que les notifications par e-mail et les SMS. Toutes les destinations que vous définissez pour votre rubrique SNS reçoivent la notification d'alarme. Pour plus d'informations, consultez la rubrique Destinations des évènements Amazon SNS.
- Vous pouvez configurer des notifications pour les événements de changement d'état des alarmes. AWS Les notifications utilisateur offrent un moyen natif de configurer de telles notifications et constituent l'approche recommandée.

En outre, CloudWatch envoie des événements à Amazon EventBridge chaque fois que les alarmes changent d'état et lorsque des alarmes sont créées, supprimées ou mises à jour. Vous pouvez écrire EventBridge des règles pour prendre des mesures ou être averti lorsque EventBridge vous recevez ces événements.

Rubriques

Notifier les utilisateurs en cas de changements d'alertes

Cette section explique comment vous pouvez utiliser les notifications AWS utilisateur ou Amazon Simple Notification Service pour informer les utilisateurs des modifications apportées aux alarmes.

Configuration des notifications AWS utilisateur

Vous pouvez utiliser les notifications AWS utilisateur pour configurer des canaux de diffusion afin d'être informé des changements CloudWatch d'état d'alarme et des événements de modification de configuration. Vous recevez une notification lorsqu'un événement correspond à une règle que vous avez spécifiée. Vous pouvez recevoir des notifications relatives à des événements via plusieurs canaux, notamment les notifications de discussion AWS Chatbot ou les notifications push AWS Console Mobile Application. Vous pouvez également consulter les notifications dans le Centre de

notifications de la console. Notifications utilisateur prend en charge l'agrégation, ce qui peut réduire le nombre de notifications que vous recevez lors d'événements spécifiques.

Les configurations de notification que vous créez avec les notifications AWS utilisateur ne sont pas prises en compte dans la limite du nombre d'actions que vous pouvez configurer par état d'alarme cible. Lorsque les notifications AWS utilisateur correspondent aux événements envoyés à Amazon EventBridge, celui-ci envoie des notifications pour toutes les alarmes de votre compte et des régions sélectionnées, sauf si vous spécifiez un filtre avancé pour autoriser ou refuser des alarmes ou des modèles spécifiques.

L'exemple suivant de filtre avancé correspond à un changement d'état d'alerte de OK à ALARM pour l'alerte nommée `ServerCpuTooHigh`.

```
{
"detail": {
    "alarmName": ["ServerCpuTooHigh"],
    "previousState": { "value": ["OK"] },
    "state": { "value": ["ALARM"] }
  }
}
```

Vous pouvez utiliser n'importe laquelle des propriétés publiées par une alarme lors d' EventBridge événements pour créer un filtre. Pour plus d'informations, consultez Événements d'alarme et EventBridge.

Configuration des notifications Amazon SNS

Vous pouvez utiliser Amazon Simple Notification Service pour envoyer à la fois des messages application-to-application (A2A) et des messages application-to-person (A2P), y compris des messages texte (SMS) mobiles et des e-mails. Pour plus d'informations, consultez la rubrique Destinations des évènements Amazon SNS.

Pour chaque état qu'une alerte peut prendre, vous pouvez configurer l'alerte pour envoyer un message à une rubrique SNS. Chaque rubrique Amazon SNS que vous configurez pour un état associé à une alerte donnée est prise en compte dans la limite du nombre d'actions que vous pouvez configurer pour cette alerte et cet état. Vous pouvez envoyer des messages à la même rubrique Amazon SNS à partir de n'importe quelle alerte de votre compte, et utiliser la même rubrique Amazon SNS pour les utilisateurs d'application à application (A2A) et d'application à personne (A2P). Étant donné que cette configuration est effectuée au niveau des alertes, seules les alertes que vous avez configurées envoient des messages à la rubrique Amazon SNS sélectionnée.

Pour commencer, créez une rubrique à laquelle vous vous abonnez. Vous pouvez, le cas échéant, publier un message test dans cette rubrique. Pour obtenir un exemple, consultez Configuration d'une rubrique Amazon SNS à l'aide du AWS Management Console. Pour plus d'informations, consultez Démarrage avec Amazon SNS.

Sinon, si vous prévoyez d'utiliser le AWS Management Console pour créer votre CloudWatch alarme, vous pouvez ignorer cette procédure car vous pouvez créer le sujet lors de la création de l'alarme.

Lorsque vous créez une CloudWatch alarme, vous pouvez ajouter des actions pour chaque état cible dans lequel l'alarme entre. Ajoutez une notification Amazon SNS pour l'état dont vous souhaitez être informé et sélectionnez la rubrique Amazon SNS que vous avez créée à l'étape précédente pour envoyer une notification par e-mail lorsque l'alerte passe à l'état sélectionné.

> **Note**
>
> Lorsque vous créez une rubrique Amazon SNS, vous choisissez d'en faire une rubrique standard ou une rubrique FIFO. CloudWatch garantit la publication de toutes les notifications d'alarme pour les deux types de sujets. Cependant, même si vous utilisez un sujet FIFO, dans de rares cas, CloudWatch envoie les notifications au sujet dans le désordre. Si vous utilisez une rubrique FIFO, l'alerte définit l'ID du groupe de messages des notifications d'alerte comme un hachage de l'ARN de l'alerte.

Prévention des problèmes du député confus

Pour éviter les problèmes de sécurité liés à la confusion entre les services, nous vous recommandons d'utiliser les clés de condition `aws:SourceArn` et les clés de condition `aws:SourceAccount` globales figurant dans la politique de ressources Amazon SNS qui autorise l'accès CloudWatch à vos ressources Amazon SNS.

L'exemple de politique de ressources suivant utilise la clé de `aws:SourceArn` condition pour restreindre l'`SNS:Publish`autorisation à utiliser uniquement par les CloudWatch alarmes du compte spécifié.

```
{
    "Statement": [{
        "Effect": "Allow",
        "Principal": {
            "Service": "cloudwatch.amazonaws.com"
        },
```

```
        "Action": "SNS:Publish",
        "Resource": "arn:aws:sns:us-east-2:444455556666:MyTopic",
        "Condition": {
            "ArnLike": {
                "aws:SourceArn": "arn:aws:cloudwatch:us-east-2:111122223333:alarm:*"
            },
            "StringEquals": {
                "aws:SourceAccount": "111122223333"
            }
        }
    }]
}
```

Si un ARN d'alerte contient des caractères non ASCII, utilisez uniquement l'option `aws:SourceAccount`clé de condition globale pour limiter les autorisations.

Configuration d'une rubrique Amazon SNS à l'aide du AWS Management Console

Pour commencer, créez une rubrique à laquelle vous vous abonnez. Vous pouvez, le cas échéant, publier un message test dans cette rubrique.

Pour créer une rubrique SNS

1. Ouvrez la console Amazon SNS à partir de l'adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le tableau de bord Amazon SNS, sous Common actions (Actions courantes), choisissez Create topic (Créer une rubrique).
3. Dans la boîte de dialogue Create new topic (Créer une rubrique), pour Topic name (Nom de rubrique), saisissez un nom de rubrique (par exemple **`my-topic`**).
4. Choisissez Create topic (Créer une rubrique).
5. Copiez Topic ARN (ARN de la rubrique) pour la tâche suivante (par exemple, arn:aws:sns:us-east-1:111122223333:my-topic).

Pour s'abonner à une rubrique SNS

1. Ouvrez la console Amazon SNS à partir de l'adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le panneau de navigation, choisissez Subscriptions (Abonnements), puis Create a subscription (Créer un abonnement).

3. Dans la boîte de dialogue Create subscription (Créer un abonnement), sous Topic ARN (ARN de la rubrique), collez l'ARN de la rubrique que vous avez créé à l'étape précédente.
4. Pour Protocole, choisissez E-mail.
5. Pour Endpoint (Point de terminaison), saisissez une adresse e-mail que vous pouvez utiliser pour recevoir la notification, puis choisissez Create subscription (Créer un abonnement).
6. Depuis votre application de messagerie, ouvrez le message dans AWS Notifications et confirmez votre abonnement.

   Votre navigateur Web affiche une réponse de confirmation provenant de Amazon SNS.

Pour publier un message test dans une rubrique SNS

1. Ouvrez la console Amazon SNS à partir de l'adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le volet de navigation, choisissez Rubriques.
3. Sur la page Topics (Rubriques), sélectionnez une rubrique, puis choisissez Publish to topic (Publier dans la rubrique).
4. Sur la page Publish a message (Publier un message), tapez l'objet de votre message dans Subject (Sujet), puis saisissez un court message dans Message.
5. Choisissez Publish Message (Publier un message).
6. Vérifiez votre messagerie électronique afin de confirmer que vous avez reçu le message en provenance de la rubrique.

Configuration d'une rubrique SNS à l'aide du AWS CLI

Pour commencer, créez une rubrique SNS, puis publiez-y un message directement afin de vérifier que vous l'avez configurée correctement.

Pour configurer une rubrique SNS

1. Créez la rubrique à l'aide de la commande create-topic, comme suit.

   ```
   aws sns create-topic --name my-topic
   ```

   Amazon SNS renvoie l'ARN d'une rubrique avec le format suivant :

```
{
    "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic"
}
```

2. Abonnez votre adresse e-mail à la rubrique à l'aide de la commande subscribe. Si la demande d'abonnement aboutit, vous recevrez un e-mail de confirmation.

```
aws sns subscribe --topic-arn arn:aws:sns:us-east-1:111122223333:my-topic --
protocol email --notification-endpoint my-email-address
```

Amazon SNS retourne les informations suivantes :

```
{
    "SubscriptionArn": "pending confirmation"
}
```

3. Depuis votre application de messagerie, ouvrez le message dans AWS Notifications et confirmez votre abonnement.

Votre navigateur Web affiche une réponse de confirmation provenant de Amazon Simple Notification Service.

4. Vérifiez l'abonnement à l'aide de la list-subscriptions-by-topiccommande.

```
aws sns list-subscriptions-by-topic --topic-arn arn:aws:sns:us-
east-1:111122223333:my-topic
```

Amazon SNS retourne les informations suivantes :

```
{
  "Subscriptions": [
    {
        "Owner": "111122223333",
        "Endpoint": "me@mycompany.com",
        "Protocol": "email",
        "TopicArn": "arn:aws:sns:us-east-1:111122223333:my-topic",
        "SubscriptionArn": "arn:aws:sns:us-east-1:111122223333:my-topic:64886986-
bf10-48fb-a2f1-dab033aa67a3"
    }
  ]
```

```
}
```

5. (En option) Publiez un message test dans la rubrique à l'aide de la commande publish.

```
aws sns publish --message "Verification" --topic arn:aws:sns:us-
east-1:111122223333:my-topic
```

Amazon SNS retourne les informations suivantes.

```
{
    "MessageId": "42f189a0-3094-5cf6-8fd7-c2dde61a4d7d"
}
```

6. Vérifiez votre messagerie électronique afin de confirmer que vous avez reçu le message en provenance de la rubrique.

Événements d'alarme et EventBridge

CloudWatch envoie des événements à Amazon EventBridge chaque fois qu'une CloudWatch alarme est créée, mise à jour, supprimée ou change d'état d'alarme. Vous pouvez utiliser EventBridge ces événements pour écrire des règles qui prennent des mesures, par exemple pour vous avertir lorsqu'une alarme change d'état. Pour plus d'informations, consultez Qu'est-ce qu'Amazon EventBridge ?

CloudWatch garantit la transmission des événements de changement d'état de l'alarme à EventBridge.

Exemples d'événements provenant de CloudWatch

Cette section inclut des exemples d'événements de CloudWatch.

Changement d'état pour une alerte à métrique unique

```
{
    "version": "0",
    "id": "c4c1c1c9-6542-e61b-6ef0-8c4d36933a92",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:04:40Z",
    "region": "us-east-1",
```

```
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "configuration": {
            "description": "Goes into alarm when server CPU utilization is too high!",
            "metrics": [
                {
                    "id": "30b6c6b2-a864-43a2-4877-c09a1afc3b87",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "CPUUtilization",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Average"
                    },
                    "returnData": true
                }
            ]
        },
        "previousState": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [0.0666851903306472 (01/10/19 13:46:00)] was not greater than the threshold (50.0)
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-01T13:56:40.985+0000\",\"startDate\":\"2019-10-01T13:46:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[0.0666851903306472],
\"threshold\":50.0}",
            "timestamp": "2019-10-01T13:56:40.987+0000",
            "value": "OK"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints
 [99.50160229693434 (02/10/19 16:59:00)] was greater than the threshold (50.0) (minimum
 1 datapoint for OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:04:40.985+0000\",\"startDate\":\"2019-10-02T16:59:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[99.50160229693434],
\"threshold\":50.0}",
```

```
            "timestamp": "2019-10-02T17:04:40.989+0000",
            "value": "ALARM"
        }
    }
}
```

Changement d'état pour une alerte de mathématiques appliquées aux métriques

```
{
    "version": "0",
    "id": "2dde0eb1-528b-d2d5-9ca6-6d590caf2329",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-02T17:20:48Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "configuration": {
            "description": "Goes into alarm if total network traffic exceeds 10Kb",
            "metrics": [
                {
                    "expression": "SUM(METRICS())",
                    "id": "e1",
                    "label": "Total Network Traffic",
                    "returnData": true
                },
                {
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkIn",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
```

```
                    "returnData": false
                },
                {
                    "id": "m2",
                    "metricStat": {
                        "metric": {
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            },
                            "name": "NetworkOut",
                            "namespace": "AWS/EC2"
                        },
                        "period": 300,
                        "stat": "Maximum"
                    },
                    "returnData": false
                }
            ]
        },
        "previousState": {
            "reason": "Unchecked: Initial alarm creation",
            "timestamp": "2019-10-02T17:20:03.642+0000",
            "value": "INSUFFICIENT_DATA"
        },
        "state": {
            "reason": "Threshold Crossed: 1 out of the last 1 datapoints [45628.0
 (02/10/19 17:10:00)] was greater than the threshold (10000.0) (minimum 1 datapoint for
 OK -> ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-02T17:20:48.551+0000\",\"startDate\":\"2019-10-02T17:10:00.000+0000\",
\"period\":300,\"recentDatapoints\":[45628.0],\"threshold\":10000.0}",
            "timestamp": "2019-10-02T17:20:48.554+0000",
            "value": "ALARM"
        }
    }
}
```

Changement d'état pour une alerte de détection d'anomalies

```
{
    "version": "0",
    "id": "daafc9f1-bddd-c6c9-83af-74971fcfc4ef",
    "detail-type": "CloudWatch Alarm State Change",
```

```
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2019-10-03T16:00:04Z",
    "region": "us-east-1",
    "resources": ["arn:aws:cloudwatch:us-east-1:123456789012:alarm:EC2 CPU Utilization
 Anomaly"],
    "detail": {
        "alarmName": "EC2 CPU Utilization Anomaly",
        "state": {
            "value": "ALARM",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints [0.0
 (03/10/19 15:58:00)] was less than the lower thresholds [0.020599444741798756] or
 greater than the upper thresholds [0.3006915352732461] (minimum 1 datapoint for OK ->
 ALARM transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T16:00:04.650+0000\",\"startDate\":\"2019-10-03T15:58:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.0],\"recentLowerThresholds\":
[0.020599444741798756],\"recentUpperThresholds\":[0.3006915352732461]}",
            "timestamp": "2019-10-03T16:00:04.653+0000"
        },
        "previousState": {
            "value": "OK",
            "reason": "Thresholds Crossed: 1 out of the last 1 datapoints
 [0.166666666664241 (03/10/19 15:57:00)] was not less than the lower thresholds
 [0.0206719426210418] or not greater than the upper thresholds [0.30076870222143803]
 (minimum 1 datapoint for ALARM -> OK transition).",
            "reasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2019-10-03T15:59:04.670+0000\",\"startDate\":\"2019-10-03T15:57:00.000+0000\",
\"period\":60,\"recentDatapoints\":[0.166666666664241],\"recentLowerThresholds\":
[0.0206719426210418],\"recentUpperThresholds\":[0.30076870222143803]}",
            "timestamp": "2019-10-03T15:59:04.672+0000"
        },
        "configuration": {
            "description": "Goes into alarm if CPU Utilization is out of band",
            "metrics": [{
                "id": "m1",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
```

```
                    "period": 60,
                    "stat": "Average"
                },
                "returnData": true
            }, {
                "id": "ad1",
                "expression": "ANOMALY_DETECTION_BAND(m1, 0.8)",
                "label": "CPUUtilization (expected)",
                "returnData": true
            }]
        }
    }
}
```

Changement d'état pour une alerte composite avec alerte de suppression

```
{
    "version": "0",
    "id": "d3dfc86d-384d-24c8-0345-9f7986db0b80",
    "detail-type": "CloudWatch Alarm State Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-22T15:57:45Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "actionsSuppressedReason": "Actions suppressed by WaitPeriod",
            "value": "ALARM",
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.FirstChild transitioned to ALARM
 at Friday 22 July, 2022 15:57:45 UTC",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"ALARM\",\"timestamp
\":\"2022-07-22T15:57:45.394+0000\"}}]}",
            "timestamp": "2022-07-22T15:57:45.394+0000"
        },
        "previousState": {
            "value": "OK",
```

```
            "reason": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:SuppressionDemo.EventBridge.Main was created and its alarm
 rule evaluates to OK",
            "reasonData": "{\"triggeringAlarms\":[{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:TotalNetworkTrafficTooHigh\",\"state\":{\"value\":\"OK\",
\"timestamp\":\"2022-07-14T16:28:57.770+0000\"}},{\"arn\":\"arn:aws:cloudwatch:us-
east-1:123456789012:alarm:ServerCpuTooHigh\",\"state\":{\"value\":\"OK\",\"timestamp\":
\"2022-07-14T16:28:54.191+0000\"}}]}",
            "timestamp": "2022-07-22T15:56:14.552+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180
        }
    }
}
```

Création d'une alerte composite

```
{

    "version": "0",
    "id": "91535fdd-1e9c-849d-624b-9a9f2b1d09d0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:22Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:22.289+0000"
        },
        "configuration": {
```

```
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "alarmName": "ServiceAggregatedAlarm",
            "description": "Aggregated monitor for instance",
            "actionsEnabled": true,
            "timestamp": "2022-03-03T17:06:22.289+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Création d'une alerte composite avec alerte de suppression

```
{
    "version": "0",
    "id": "454773e1-09f7-945b-aa2c-590af1c3f8e0",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:46Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "create",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
```

```
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Mise à jour d'une alerte de métrique

```
{

    "version": "0",
    "id": "bc7d3391-47f8-ae47-f457-1b4d06118d50",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:06:34Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServerCpuTooHigh"
    ],
    "detail": {
        "alarmName": "ServerCpuTooHigh",
        "operation": "update",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:13.757+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 80,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [
                {
                    "id": "86bfa85f-b14c-ebf7-8916-7da014ce23c0",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "CPUUtilization",
                            "dimensions": {
                                "InstanceId": "i-12345678901234567"
                            }
                        },
```

```
                    "period": 300,
                    "stat": "Average"
                },
                "returnData": true
            }
        ],
        "alarmName": "ServerCpuTooHigh",
        "description": "Goes into alarm when server CPU utilization is too high!",
        "actionsEnabled": true,
        "timestamp": "2022-03-03T17:06:34.267+0000",
        "okActions": [],
        "alarmActions": [],
        "insufficientDataActions": []
    },
    "previousConfiguration": {
        "evaluationPeriods": 1,
        "threshold": 70,
        "comparisonOperator": "GreaterThanThreshold",
        "treatMissingData": "ignore",
        "metrics": [
            {
                "id": "d6bfa85f-893e-b052-a58b-4f9295c9111a",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "CPUUtilization",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 300,
                    "stat": "Average"
                },
                "returnData": true
            }
        ],
        "alarmName": "ServerCpuTooHigh",
        "description": "Goes into alarm when server CPU utilization is too high!",
        "actionsEnabled": true,
        "timestamp": "2022-03-03T17:06:13.757+0000",
        "okActions": [],
        "alarmActions": [],
        "insufficientDataActions": []
    }
```

```
    }
}
```

Mise à jour d'une alerte composite avec une alerte de suppression

```
{
    "version": "0",
    "id": "4c6f4177-6bd5-c0ca-9f05-b4151c54568b",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T13:59:56Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "update",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        },
        "previousConfiguration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 180,
```

```
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:46.425+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```

Suppression d'une alerte de mathématiques appliquées aux métriques

```
{

    "version": "0",
    "id": "f171d220-9e1c-c252-5042-2677347a83ed",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-03-03T17:07:13Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:TotalNetworkTrafficTooHigh"
    ],
    "detail": {
        "alarmName": "TotalNetworkTrafficTooHigh",
        "operation": "delete",
        "state": {
            "value": "INSUFFICIENT_DATA",
            "timestamp": "2022-03-03T17:06:17.672+0000"
        },
        "configuration": {
            "evaluationPeriods": 1,
            "threshold": 10000,
            "comparisonOperator": "GreaterThanThreshold",
            "treatMissingData": "ignore",
            "metrics": [{
                    "id": "m1",
                    "metricStat": {
                        "metric": {
                            "namespace": "AWS/EC2",
                            "name": "NetworkIn",
                            "dimensions": {
```

```
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 300,
                    "stat": "Maximum"
                },
                "returnData": false
            },
            {
                "id": "m2",
                "metricStat": {
                    "metric": {
                        "namespace": "AWS/EC2",
                        "name": "NetworkOut",
                        "dimensions": {
                            "InstanceId": "i-12345678901234567"
                        }
                    },
                    "period": 300,
                    "stat": "Maximum"
                },
                "returnData": false
            },
            {
                "id": "e1",
                "expression": "SUM(METRICS())",
                "label": "Total Network Traffic",
                "returnData": true
            }
        ],
        "alarmName": "TotalNetworkTrafficTooHigh",
        "description": "Goes into alarm if total network traffic exceeds 10Kb",
        "actionsEnabled": true,
        "timestamp": "2022-03-03T17:06:17.672+0000",
        "okActions": [],
        "alarmActions": [],
        "insufficientDataActions": []

    }
  }
}
```

Suppression d'une alerte composite avec une alerte de suppression

```
{
    "version": "0",
    "id": "e34592a1-46c0-b316-f614-1b17a87be9dc",
    "detail-type": "CloudWatch Alarm Configuration Change",
    "source": "aws.cloudwatch",
    "account": "123456789012",
    "time": "2022-07-14T14:00:01Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:cloudwatch:us-east-1:123456789012:alarm:ServiceAggregatedAlarm"
    ],
    "detail": {
        "alarmName": "ServiceAggregatedAlarm",
        "operation": "delete",
        "state": {
            "actionsSuppressedBy": "WaitPeriod",
            "value": "ALARM",
            "timestamp": "2022-07-14T13:59:46.425+0000"
        },
        "configuration": {
            "alarmRule": "ALARM(ServerCpuTooHigh) OR
 ALARM(TotalNetworkTrafficTooHigh)",
            "actionsSuppressor": "ServiceMaintenanceAlarm",
            "actionsSuppressorWaitPeriod": 120,
            "actionsSuppressorExtensionPeriod": 360,
            "alarmName": "ServiceAggregatedAlarm",
            "actionsEnabled": true,
            "timestamp": "2022-07-14T13:59:56.290+0000",
            "okActions": [],
            "alarmActions": [],
            "insufficientDataActions": []
        }
    }
}
```


Gérer les alarmes

Modifier ou supprimer une CloudWatch alarme

Vous pouvez modifier ou supprimer une alerte existante.

Vous ne pouvez pas modifier le nom d'une alerte existante. Vous pouvez copier une alerte et attribuer un nom différent à la nouvelle alerte. Pour copier une alerte, cochez la case en regard de l'alerte dans la liste des alertes et choisissez Action, Copy (Copier).

Pour modifier une alerte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All Alarms (Toutes les alertes).
3. Sélectionnez le nom de l'alerte.
4. Pour rechercher des balises, choisissez l'onglet Balises, puis sélectionnez Gérer les balises.
5. Pour modifier d'autres parties de l'alerte, choisissez Actions, Modifier.

   La page Specify metric and conditions (Spécifier les métriques et les conditions) apparaît, présentant un graphique et d'autres informations sur la métrique et la statistique que vous avez sélectionnées.
6. Pour modifier la métrique, choisissez Edit (Modifier), choisissez l'onglet All metrics (Toutes les métriques), puis effectuez l'une des actions suivantes :
   - Choisissez l'espace de noms du service qui contient la métrique de votre choix. Ensuite, choisissez des options au fur et à mesure qu'elles s'affichent pour affiner les choix. Lorsqu'une liste des métriques apparaît, sélectionnez la case à cocher en regard de la métrique voulue.
   - Dans la zone de recherche, tapez le nom d'une métrique, d'une dimension ou d'un ID de ressource, puis appuyez sur Entrée. Ensuite, choisissez l'un des résultats et continuez jusqu'à ce qu'une liste des métriques s'affiche. Cochez la case en regard de la métrique voulue.

   Choisissez Select metric (Sélectionner une métrique).
7. Pour modifier d'autres aspects de l'alerte, sélectionnez les options appropriées. Pour modifier le nombre de points de données devant se situer au-delà du seuil pour que l'alerte passe à l'état `ALARM` ou pour modifier la façon dont les données manquantes sont traitées, choisissez Additional configuration (Configuration supplémentaire).
8. Choisissez Suivant.
9. Sous Notification, Auto Scaling action, et EC2 action, vous pouvez modifier les actions réalisées lorsque l'alerte est déclenchée. Ensuite, sélectionnez Suivant.
10. Vous pouvez modifier la description de l'alerte.

Vous ne pouvez pas modifier le nom d'une alerte existante. Vous pouvez copier une alerte et attribuer un nom différent à la nouvelle alerte. Pour copier une alerte, cochez la case en regard de l'alerte dans la liste des alertes et choisissez Action, Copy (Copier).

11. Choisissez Suivant.
12. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Update alarm (Mettre à jour une alerte).

Mettre à jour une liste de notifications par e-mail ayant été créée à l'aide de la console Amazon SNS

1. Ouvrez la console Amazon SNS à partir de l’adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le panneau de navigation, choisissez Topics (Rubriques), puis sélectionnez l'ARN de votre liste de notifications (rubrique).
3. Effectuez l’une des actions suivantes :
   - Pour ajouter une adresse e-mail, choisissez Create subscription (Créer un abonnement). Pour Protocole, choisissez E-mail. Indiquez l'adresse e-mail du nouveau destinataire dans Endpoint (Point de terminaison). Choisissez Créer un abonnement.
   - Pour supprimer une adresse e-mail, choisissez l'ID d'abonnement (Subscription ID). Choisissez Other subscription actions (Autres actions d'abonnement), Delete subscriptions (Supprimer des abonnements).
4. Choisissez Publish to topic (Publier dans la rubrique.

Pour supprimer une alarme

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, cliquez sur Alarms (alertes).
3. Cochez la case située à gauche du nom de l'alerte, puis choisissez Actions, Delete (Supprimer).
4. Sélectionnez Delete (Supprimer).

Masquer les alarmes Auto Scaling

Lorsque vous visualisez vos alarmes dans le AWS Management Console, vous pouvez masquer les alarmes liées à Amazon EC2 Auto Scaling et à Application Auto Scaling. Cette fonction est disponible uniquement dans AWS Management Console.

Pour masquer temporairement les alertes Auto Scaling

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All alarms (Toutes les alertes), puis sélectionnez Hide Auto Scaling alarms (Masquer les alertes à scalabilité automatique).

Cas d'utilisation et exemples d'alertes

Les sections suivantes fournissent des exemples et des didacticiels pour les alertes pour les cas d'utilisation courants.

Créez une alarme de facturation pour surveiller vos AWS frais estimés

Vous pouvez suivre l'estimation de vos AWS frais en utilisant Amazon CloudWatch. Lorsque vous activez le suivi des frais estimés pour votre AWS compte, les frais estimés sont calculés et envoyés plusieurs fois par jour CloudWatch sous forme de données métriques.

Les données de métriques de facturation sont stockées dans la région USA Est (Virginie du Nord) et regroupent des frais du monde entier. Ces données incluent les frais estimés pour chaque service AWS que vous utilisez, en plus du total estimatif de vos AWS frais.

L'alerte se déclenche lorsque votre compte de facturation dépasse le seuil que vous spécifiez. Elle se déclenche uniquement lorsque la facturation actuelle est supérieure au seuil. Elle n'utilise pas de projections en fonction de votre utilisation jusqu'à présent dans le mois.

Si vous créez une alerte de facturation lorsque vos frais ont déjà dépassé le seuil, l'alerte passe à l'état `ALARM` immédiatement.

> **Note**
>
> Pour plus d'informations sur l'analyse des CloudWatch frais qui vous ont déjà été facturés, consultezCloudWatch facturation et coût.

Tâches

- Activation des alertes de facturation
- Création d'une alarme de facturation
- Suppression d'une alerte de facturation

Activation des alertes de facturation

Avant de créer une alarme pour vos frais estimés, vous devez activer les alertes de facturation afin de pouvoir surveiller vos AWS frais estimés et créer une alarme à l'aide des données métriques de facturation. Après avoir activé les alertes de facturation, vous ne pouvez pas désactiver la collecte de données, mais vous pouvez supprimer toute alerte de facturation que vous avez créée.

Après avoir activé les alertes de facturation pour la première fois, il faut environ 15 minutes avant de pouvoir afficher des données de facturation et de configurer des alertes de facturation.

Prérequis

- Vous devez être connecté à l'aide des informations d'identification utilisateur racine du compte ou en tant qu'utilisateur IAM disposant de l'autorisation d'afficher les informations de facturation.
- Pour les comptes à facturation Consolidée, les données de facturation de chaque compte lié sont disponibles en vous connectant en tant que le compte de règlement. Vous pouvez afficher les données de facturation pour le montant total des coûts estimés et pour les coûts estimés par service pour chaque compte lié, ainsi que pour le compte de regroupement.
- Dans un compte de facturation consolidé, les métriques de compte lié aux membres ne sont capturées que si le compte payeur active la préférence Recevoir des alertes de facturation. Si vous modifiez votre compte gestion/payeur, vous devez activer les alertes de facturation dans le nouveau compte gestion/payeur.
- Le compte ne doit pas faire partie du réseau de partenaires Amazon (APN) car les statistiques de facturation ne sont pas publiées CloudWatch pour les comptes APN. Pour plus d'informations, consultez Réseau de partenaires AWS .

Pour activer la surveillance des coûts estimés

1. Ouvrez la AWS Billing console à l'adresse https://console.aws.amazon.com/billing/.
2. Dans le panneau de navigation, sélectionnez Billing Preferences (Préférences de facturation).
3. Pour Préférences des alertes, choisissez Modifier.

4. Choisissez Recevoir des alertes CloudWatch de facturation.
5. Choisissez Save preferences (Enregistrer des préférences).

Création d'une alarme de facturation

> **Important**
>
> Avant de créer une alerte de facturation, vous devez définir votre région sur USA Est (Virginie du Nord). Les données de métriques de facturation sont stockées dans cette région et regroupent des frais du monde entier. Vous devez également activer les alertes de facturation pour votre compte ou dans le compte de gestion/payeur (si vous utilisez la facturation consolidée). Pour plus d'informations, consultez Activation des alertes de facturation.

Dans cette procédure, vous créez une alarme qui envoie une notification lorsque vos frais estimés AWS dépassent un seuil défini.

Pour créer une alarme de facturation 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 Alarms (alertes), puis All alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Sélectionner une métrique. Dans Browse (Parcourir), sélectionnez Billing (Facturation), puis Total Estimated Charge (Frais estimés totaux).

   > **Note**
   >
   > Si vous ne voyez pas la métrique Facturation/Total des frais estimés, activez les alertes de facturation et changez votre région pour USA Est (Virginie du Nord). Pour plus d'informations, consultez Activation des alertes de facturation.

5. Cochez la case correspondant à la EstimatedChargesmétrique, puis sélectionnez Select metric.
6. Pour Statistique, choisissez Maximum.
7. Pour Period (Période), choisissez 6 hours (6 heures).
8. Pour Threshold type (Type de seuil), choisissez Static (Statique).
9. Pour Whenever, EstimatedCharges c'est... , choisissez Greater.

10. Pour que... , définissez la valeur à laquelle vous souhaitez déclencher votre alerte. Par exemple, **`200`** USD.

    Les valeurs EstimatedChargesmétriques sont uniquement en dollars américains (USD) et la conversion des devises est assurée par Amazon Services LLC. Pour plus d'informations, voir Qu'est-ce que c'est AWS Billing ? .

    > **Note**
    >
    > Après avoir défini une valeur seuil, le graphique d'aperçu affiche vos frais estimés pour le mois en cours.

11. Choisissez Configuration supplémentaire et procédez comme suit :

    - Pour Datapoints to alarm (Points de données à alarmer), indiquez 1 out of 1 (1 sur 1).
    - Pour Missing data treatment (Traitement des données manquantes), sélectionnez Treat missing data as missing (Traiter les données manquantes comme manquantes).

12. Choisissez Suivant.
13. Sous Notification, assurez-vous que l'option En alarme est sélectionnée. Spécifiez une rubrique Amazon SNS à notifier lorsque l'alerte passe à l'état `ALARM`. La rubrique Amazon SNS peut inclure votre adresse e-mail afin que vous receviez un e-mail lorsque le montant de facturation dépasse le seuil que vous avez spécifié.

    Vous pouvez sélectionner une rubrique Amazon SNS existante, créer une nouvelle rubrique Amazon SNS ou utiliser l'ARN d'une rubrique pour notifier un autre compte. Si vous voulez que votre alarme envoie plusieurs notifications pour le même état de l'alarme ou pour des états de l'alarme différents, sélectionnez Add notification (Ajouter une notification).

14. Choisissez Suivant.
15. Sous Name and description (Nom et description), saisissez un nom pour votre alarme. Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII.

    - (Facultatif) Saisissez une description de votre alarme. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

16. Choisissez Suivant.

17. Sous Preview and create (Prévisualiser et créer), assurez-vous que votre configuration est correcte, puis sélectionnez Create alarm (Créer l'alarme).

Suppression d'une alerte de facturation

Lorsque vous n'avez plus besoin d'une alerte de facturation, vous pouvez la supprimer.

Pour supprimer une alerte de facturation

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Si nécessaire, changez la région en USA Est (Virginie du Nord). Les données de métriques de facturation sont stockées dans cette région et reflètent des frais du monde entier.
3. Dans le panneau de navigation, choisissez Alarms (alertes), All alarms (Toutes les alertes).
4. Cochez la case en regard de l'alerte, et choisissez Actions puis Supprimer.
5. Lorsque vous êtes invité à confirmer l’opération, choisissez Yes, Delete.

Créer une alerte d'utilisation du processeur

Vous pouvez créer une CloudWatch alarme qui envoie une notification à l'aide d'Amazon SNS lorsque l'état de l'alarme passe de à`OK`. `ALARM`

L'alerte passe à l'état `ALARM` lorsque l'utilisation d'UC moyenne d'une instance EC2 dépasse un seuil défini pendant le nombre spécifié de périodes consécutives.

Configuration d'une alarme d'utilisation du processeur à l'aide du AWS Management Console

Suivez ces étapes pour utiliser le AWS Management Console afin de créer une alarme d'utilisation du processeur.

Pour créer une alerte basée sur l'utilisation de l'UC

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All Alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Sélectionner une métrique.

5. Dans l'onglet Toutes les métriques, choisissez Métriques EC2.
6. Choisissez une catégorie de métrique (par exemple, Per-Instance Metrics(métriques par instance)).
7. Recherchez la ligne contenant l'instance que vous souhaitez répertorier dans la InstanceIdcolonne et CPUUtilization dans la colonne Nom de la métrique. Cochez la case en regard de cette ligne, puis choisissez Select metric (Sélectionner la métrique).
8. Sous Spécifier la métrique et les conditions, pour Statistique, choisissez Moyenne, choisissez l'un des percentiles prédéfinis ou spécifiez un percentile personnalisé (par exemple, **`p95.45`**).
9. Choisissez une période (par exemple, **`5 minutes`**).
10. Sous Conditions, spécifiez les éléments suivants :

    a. Pour Threshold type (Type de seuil), choisissez Static (Statique).

    b. Pour Chaque fois que CPUUtilization est, spécifiez Supérieure. Sous à..., spécifiez le seuil qui déclenche l'alerte pour passer à l'état alerte si l'utilisation de l'UC dépasse ce pourcentage. Par exemple, 70.

    c. 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 pour la première valeur un nombre inférieur à celui de la seconde valeur. Pour plus d'informations, consultez . Évaluation d'une alerte.

    d. 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, consultez . Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.

    e. Si l'alerte utilise un centile comme statistique surveillée, une zone Percentiles with low samples (Centiles avec exemples de bas niveau) s'affiche. Utilisez-la pour choisir si vous souhaitez évaluer ou ignorer les cas avec des taux d'échantillons faibles. Si vous sélectionnez ignore (ignorer : conserver l'état d'alerte), l'état actuel de l'alerte est toujours conservé lorsque la taille de l'échantillon est trop réduite. Pour plus d'informations, consultez . CloudWatch Alarmes basées sur les percentiles et échantillons de données faibles.

11. Choisissez Suivant.

12. Sous Notification, choisissez In alarm (Dans l'alerte) et sélectionnez une rubrique SNS à notifier lorsque l'alerte est en état `ALARM`

    Pour que l'alerte envoie plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, choisissez Add notification (Ajouter une notification).

    Pour que l'alerte n'envoie pas de notifications, choisissez Remove (Supprimer).
13. Lorsque vous avez terminé, choisissez Next (Suivant).
14. Saisissez un nom et une description pour l'alerte. Ensuite, sélectionnez Suivant.

    Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.
15. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

Configuration d'une alarme d'utilisation du processeur à l'aide du AWS CLI

Suivez ces étapes pour utiliser le AWS CLI afin de créer une alarme d'utilisation du processeur.

Pour créer une alerte basée sur l'utilisation de l'UC

1. Configurez une rubrique SNS. Pour plus d'informations, consultez Configuration des notifications Amazon SNS.
2. Créez une alarme à l'aide de la put-metric-alarmcommande suivante.

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70%" --metric-name CPUUtilization --namespace AWS/EC2 --statistic
 Average --period 300 --threshold 70 --comparison-operator GreaterThanThreshold --
dimensions  Name=InstanceId,Value=i-12345678 --evaluation-periods 2 --alarm-actions
 arn:aws:sns:us-east-1:111122223333:my-topic --unit Percent
```

3. Testez l'alarme en forçant un changement d'état de l'alarme à l'aide de la set-alarm-statecommande.

    a. Remplacez la valeur `INSUFFICIENT_DATA` de l'état de l'alerte par `OK`.

```
aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
 "initializing" --state-value OK
```

b. Remplacez la valeur OK de l'état de l'alerte par ALARM.

```
aws cloudwatch set-alarm-state --alarm-name cpu-mon --state-reason
 "initializing" --state-value ALARM
```

c. Vérifiez si vous avez reçu une notification concernant l'alerte.

Créer une alerte de latence d'équilibreur de charge qui envoie un e-mail

Vous pouvez configurer une notification Amazon SNS et configurer une alerte qui surveille une latence qui excède 100 ms pour votre Classic Load Balancer.

Configuration d'une alarme de latence à l'aide du AWS Management Console

Suivez ces étapes pour utiliser le pour créer une alarme AWS Management Console de latence de l'équilibreur de charge.

Pour créer une alerte de latence d'équilibreur de charge

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All Alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Sous CloudWatch Métriques par catégorie, choisissez la catégorie ELB Metrics.
5. Sélectionnez la ligne avec le Classic Load Balancer et la métrique Latency (Latence).
6. Pour les statistiques, choisissez Average (Moyenne), l'un des centiles prédéfinis, ou spécifiez un centile personnalisé (par exemple, **`p95.45`**).
7. Pour la période, choisissez 1 Minute.
8. Choisissez Suivant.
9. Sous Alarm Threshold (Seuil d'alerte), saisissez un nom unique pour l'alerte (par exemple : **`myHighCpuAlarm`**) et une description de l'alerte (par exemple :**`Alarm when Latency exceeds 100s`**). Les noms des alertes ne doivent contenir que des caractères UTF-8 et ne peuvent pas contenir de caractères de contrôle ASCII.

Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

10. Sous Whenever (À chaque fois que), pour is (est), choisissez **>** et tapez **`0.1`**. Pour for (pour), entrez **`3`**.
11. Sous Paramètres supplémentaires, dans Traiter les données manquantes comme, choisissez à ignorer (conserver l'état d'alerte) de sorte que les points de données manquants ne déclenchent pas de changement d'état de l'alerte.

    Dans Percentiles avec exemples de bas niveau, choisissez à ignorer (conserver l'état d'alerte), de sorte que l'alerte évalue uniquement les situations comptant des nombres d'échantillons de données adéquats.
12. Sous Actions, pour Whenever this alarm (Chaque fois que cette alerte), choisissez State is ALARM (L'état est alerte). Pour Send notification to (Envoyer une notification à), choisissez une rubrique SNS existante ou créez-en une.

    Pour créer une rubrique SNS, choisissez New list (Nouvelle liste). Pour Send notification to (Envoyer une notification à), entrez le nom de la rubrique SNS (**`myHighCpuAlarm`**, par exemple) et, pour Email list, entrez une liste d'adresses e-mail, séparées par une virgule, à avertir quand l'alerte passe à l'état `ALARM`. Chaque adresse e-mail reçoit un e-mail de confirmation d'abonnement à la rubrique. Vous devrez confirmer l'abonnement avant de pouvoir recevoir des notifications.
13. Choisissez Create Alarm (Créer l'alerte).

Configuration d'une alarme de latence à l'aide du AWS CLI

Suivez ces étapes pour utiliser le pour créer une alarme AWS CLI de latence de l'équilibreur de charge.

Pour créer une alerte de latence d'équilibreur de charge

1. Configurez une rubrique SNS. Pour plus d’informations, consultez Configuration des notifications Amazon SNS.
2. Créez l'alarme à l'aide de la put-metric-alarmcommande suivante :

```
aws cloudwatch put-metric-alarm --alarm-name lb-mon --alarm-description "Alarm
 when Latency exceeds 100s" --metric-name Latency --namespace AWS/ELB --statistic
 Average --period 60 --threshold 100 --comparison-operator GreaterThanThreshold --
dimensions Name=LoadBalancerName,Value=my-server --evaluation-periods 3 --alarm-
actions arn:aws:sns:us-east-1:111122223333:my-topic --unit Seconds
```

3. Testez l'alarme en forçant un changement d'état de l'alarme à l'aide de la set-alarm-statecommande.

   a. Remplacez la valeur `INSUFFICIENT_DATA` de l'état de l'alerte par `OK`.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Remplacez la valeur `OK` de l'état de l'alerte par `ALARM`.

   ```
   aws cloudwatch set-alarm-state --alarm-name lb-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Vérifiez si vous avez reçu une notification par e-mail concernant l'alerte.

Créer une alerte de débit du stockage qui envoie un e-mail

Vous pouvez configurer une notification SNS et une alerte qui se déclenche quand Amazon EBS dépasse un débit de 100 Mo.

Configuration d'une alerte de débit du stockage via la AWS Management Console

Suivez ces étapes pour AWS Management Console créer une alarme basée sur le débit d'Amazon EBS.

Pour créer une alerte de débit du stockage

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All Alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Sous EBS Metrics, choisissez une catégorie de métrique.
5. Sélectionnez la ligne contenant le volume et la VolumeWriteBytesmétrique.

6. Pour les statistiques, choisissez Average (Moyenne). Pour la période, choisissez 5 Minutes. Choisissez Suivant.
7. Sous Alarm Threshold (Seuil d'alerte), saisissez un nom unique pour l'alerte (par exemple : **`myHighWriteAlarm`**) et une description de l'alerte (par exemple :**`VolumeWriteBytes exceeds 100,000 KiB/s`**). Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.
8. Sous Whenever (À chaque fois que), pour is (est), choisissez **>** et tapez **`100000`**. Pour for, tapez **`15`** périodes consécutives.

   Une représentation graphique du seuil apparaît sous Alarm Preview (Aperçu de l'alerte).
9. Sous Paramètres supplémentaires, dans Traiter les données manquantes comme, choisissez à ignorer (conserver l'état d'alerte) de sorte que les points de données manquants ne déclenchent pas de changement d'état de l'alerte.
10. Sous Actions, pour Whenever this alarm (Chaque fois que cette alerte), choisissez State is ALARM (L'état est alerte). Pour Send notification to, choisissez une rubrique SNS existante ou créez-en une.

    Pour créer une rubrique SNS, choisissez New list (Nouvelle liste). Pour Send notification to (Envoyer une notification à), entrez le nom de la rubrique SNS (**`myHighCpuAlarm`**, par exemple) et, pour Email list, entrez une liste d'adresses e-mail, séparées par une virgule, à avertir quand l'alerte passe à l'état `ALARM`. Chaque adresse e-mail reçoit un e-mail de confirmation d'abonnement à la rubrique. Vous devez confirmer l'abonnement avant de pouvoir recevoir des notifications à une adresse e-mail.
11. Choisissez Create Alarm (Créer l'alerte).

Configuration d'une alarme de débit de stockage à l'aide du AWS CLI

Suivez ces étapes pour AWS CLI créer une alarme basée sur le débit d'Amazon EBS.

Pour créer une alerte de débit du stockage

1. Créez une rubrique SNS. Pour plus d'informations, consultez Configuration des notifications Amazon SNS.
2. Créez l'alerte.

```
aws cloudwatch put-metric-alarm --alarm-name ebs-mon --alarm-description "Alarm
 when EBS volume exceeds 100MB throughput" --metric-name VolumeReadBytes --
namespace AWS/EBS --statistic Average --period 300 --threshold 100000000 --
comparison-operator GreaterThanThreshold --dimensions Name=VolumeId,Value=my-
volume-id --evaluation-periods 3 --alarm-actions arn:aws:sns:us-
east-1:111122223333:my-alarm-topic --insufficient-data-actions arn:aws:sns:us-
east-1:111122223333:my-insufficient-data-topic
```

3. Testez l'alarme en forçant un changement d'état de l'alarme à l'aide de la set-alarm-statecommande.

   a. Remplacez la valeur `INSUFFICIENT_DATA` de l'état de l'alerte par `OK`.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
    "initializing" --state-value OK
   ```

   b. Remplacez la valeur `OK` de l'état de l'alerte par `ALARM`.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
    "initializing" --state-value ALARM
   ```

   c. Remplacez la valeur `ALARM` de l'état de l'alerte par `INSUFFICIENT_DATA`.

   ```
   aws cloudwatch set-alarm-state --alarm-name ebs-mon --state-reason
    "initializing" --state-value INSUFFICIENT_DATA
   ```

   d. Vérifiez si vous avez reçu une notification par e-mail concernant l'alerte.

Création d'une alarme sur les indicateurs de compteur Performance Insights à partir d'une AWS base de données

CloudWatch inclut une fonction mathématique métrique DB_PERF_INSIGHTS que vous pouvez utiliser pour intégrer les indicateurs de compteur Performance Insights CloudWatch depuis Amazon Relational Database Service et Amazon DocumentDB (avec compatibilité avec MongoDB). DB_PERF_INSIGHTS introduit également la métrique `DBLoad` à des intervalles inférieurs à la minute. Vous pouvez définir des CloudWatch alarmes sur ces métriques.

Pour obtenir plus d'informations sur l'Analyse des performances d'Amazon RDS, consultez Surveillance de la charge de la base de données avec Performance Insights sur Amazon RDS.

Pour obtenir plus d'informations sur l'Analyse des performances d'Amazon DocumentDB, consultez Monitoring with Performance Insights.

La détection des anomalies n'est pas prise en charge pour les alertes basées sur la fonction DB_PERF_INSIGHTS.

> **Note**
>
> Les métriques haute résolution avec une granularité inférieure à la minute récupérées par DB_PERF_INSIGHTS ne s'appliquent qu'à la métrique DBLoad, ou aux métriques du système d'exploitation si vous avez activé la Surveillance améliorée à une résolution supérieure. Pour de plus amples informations sur la Surveillance améliorée d'Amazon RDS, consultez Surveillance des métriques du système d'exploitation à l'aide de la Surveillance améliorée.
>
> Vous pouvez créer une alarme haute résolution à l'aide de la fonction DB_PERF_INSIGHTS. La plage d'évaluation maximale pour une alarme haute résolution est de trois heures. Vous pouvez utiliser la CloudWatch console pour représenter graphiquement les métriques récupérées à l'aide de la fonction DB_PERF_INSIGHTS pour n'importe quel intervalle de temps.

Pour créer une alerte basée sur les métriques d'Analyse des performances

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), puis All alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Select Metric (Sélectionner une métrique).
5. Choisissez le menu déroulant Ajouter des mathématiques, puis sélectionnez Métriques de performance des bases de données, DB_PERF_INSIGHTS dans la liste.

   Après avoir sélectionné DB_PERF_INSIGHTS, une zone d'expression mathématique apparaît à l'endroit où vous appliquez ou modifiez des expressions mathématiques.
6. Dans la zone d'expression mathématique, saisissez votre expression mathématique DB_PERF_INSIGHTS, puis sélectionnez Appliquer.

   Par exemple, **`DB_PERF_INSIGHTS('RDS', 'db-ABCDEFGHIJKLMNOPQRSTUVWXY1', 'os.cpuUtilization.user.avg')`**

> **Important**
>
> Lorsque vous utilisez l'expression mathématique DB_PERF_INSIGHTS, vous devez spécifier l'ID de ressource de base de données unique de la base de données. Il est différent de l'identifiant de base de données. Pour trouver un identifiant de ressource de base de données dans la console Amazon RDS, choisissez l'instance de base de données pour en afficher les détails. Choisissez ensuite l'onglet Configuration. L'ID de ressource est indiqué dans la section Configuration.

Pour plus d'informations sur la fonction DB_PERF_INSIGHTS et les autres fonctions disponibles avec les mathématiques de métrique, consultez Syntaxe et fonctions des mathématiques appliquées aux métriques.

7. Choisissez Select metric (Sélectionner une métrique).

   La page Specify metric and conditions (Spécifier les métriques et les conditions) apparaît, présentant un graphique et d'autres informations sur l'expression mathématique que vous avez sélectionnée.

8. Pour Whenever ***`expression`*** is (À chaque fois que l'expression est), spécifiez si la métrique doit être supérieure à, inférieure à ou égale au seuil. Dans than... (à...), spécifiez la valeur de seuil.

9. 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 pour la première valeur un nombre inférieur à celui de la seconde valeur. Pour plus d'informations, consultez . Évaluation d'une alerte.

10. 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, consultez . Configuration de la façon dont les CloudWatch alarmes traitent les données manquantes.

11. Choisissez Next (Suivant).

12. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alerte passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

    Pour que l'alerte envoie plusieurs notifications pour le même état d'alerte ou pour les différents états d'alerte, choisissez Add notification (Ajouter une notification).

    Pour que l'alerte n'envoie pas de notifications, choisissez Remove (Supprimer).

13. Pour que l'alarme exécute des actions Auto Scaling, EC2, Lambda ou Systems Manager, cliquez sur le bouton approprié, puis choisissez l'état de l'alerte et l'action à effectuer. Si vous choisissez une fonction Lambda comme action d'alarme, vous spécifiez le nom de la fonction ou l'ARN, et vous pouvez éventuellement choisir une version spécifique de la fonction.

    Les alertes peuvent effectuer des actions du Systems Manager uniquement lorsqu'elles passent à l'état ALARM. Pour plus d'informations sur les actions de Systems Manager, voir Configuration CloudWatch pour créer à OpsItems partir d'alarmes et Création d'incidents.

    > **Note**
    >
    > Pour créer une alerte qui exécute une action SSM Incident Manager, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez les exemples de politiques basées sur l'identité pour AWS Systems Manager Incident Manager.

14. Lorsque vous avez terminé, choisissez Next (Suivant).
15. Saisissez un nom et une description pour l'alerte. Ensuite, sélectionnez Suivant.

    Le nom ne doit contenir que des caractères UTF-8 et ne peut pas contenir de caractères de contrôle ASCII. La description peut inclure le formatage du markdown, qui est affiché uniquement dans l'onglet Détails de l'alarme de la CloudWatch console. Le markdown peut être utile pour ajouter des liens vers des runbooks ou d'autres ressources internes.

16. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

Création d'alarmes qui arrêtent, mettent hors service, redémarrent ou récupèrent une instance EC2

À l'aide des actions CloudWatch d'alarme Amazon, vous pouvez créer des alarmes qui arrêtent, mettent fin, redémarrent ou restaurent automatiquement vos instances EC2. Vous pouvez utiliser les

actions d'arrêt ou de terminaison pour vous permettre d'économiser de l'argent quand vous n'avez plus besoin qu'une instance s'exécute. De même, les actions de redémarrage et de récupération vous permettent de redémarrer automatiquement ces instances ou de les récupérer sur un nouveau matériel en cas de déficience du nouveau matériel.

Il existe un certain nombre de scénarios dans lesquels vous pourriez vouloir arrêter ou terminer automatiquement votre instance. Par exemple, vous pourriez avoir des instances dédiées aux tâches de traitement différé de la paie ou de calcul scientifique qui s'exécutent pendant une durée, puis achèvent leur travail. Plutôt que de laisser ces instances demeurer inactives (et d'accumuler les frais), vous pouvez les arrêter ou les terminer, ce qui vous aide à économiser de l'argent. La principale différence entre l'utilisation des actions d'alarme d'arrêt ou de fin est que vous pouvez facilement redémarrer une instance arrêtée si vous devez l'exécuter à nouveau ultérieurement. Vous pouvez conserver également les mêmes ID d'instance et volume racine. Cependant, vous ne pouvez pas redémarrer une instance terminée. Vous devez à la place lancer une nouvelle instance.

Vous pouvez ajouter les actions d'arrêt, de résiliation ou de redémarrage à toute alarme définie sur une métrique Amazon EC2 par instance, y compris les mesures de surveillance de base et détaillées fournies par Amazon CloudWatch (dans l'espace de noms AWS/EC2), en plus des mesures personnalisées incluant la dimension « InstanceId = », à condition que la InstanceId valeur fasse référence à une instance Amazon EC2 en cours d'exécution valide. Vous pouvez également ajouter l'action de restauration à des alarmes définies sur n'importe quelle mesure Amazon EC2 par instance, à l'exception de `StatusCheckFailed_Instance`.

Pour configurer une action CloudWatch d'alarme capable de redémarrer, d'arrêter ou de mettre fin à une instance, vous devez utiliser un rôle IAM lié à un service,. AWSServiceRoleForCloudWatchEvents Le rôle AWSServiceRoleForCloudWatchEvents IAM permet d' AWS effectuer des actions d'alarme en votre nom.

Pour créer le rôle lié au service pour CloudWatch Events, utilisez la commande suivante :

```
aws iam create-service-linked-role --aws-service-name events.amazonaws.com
```

Prise en charge de la console

Vous pouvez créer des alarmes à l'aide de la CloudWatch console ou de la console Amazon EC2. Les procédures décrites dans cette documentation utilisent la CloudWatch console. Pour voir les procédures qui utilisent la console Amazon EC2, consultez Créer des alarmes qui arrêtent, mettent hors service, redémarrent ou récupèrent une instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Autorisations

Si vous utilisez un compte AWS Identity and Access Management (IAM) pour créer ou modifier une alarme qui exécute des actions EC2 ou des actions Systems Manager OpsItem , vous devez en avoir l'`iam:CreateServiceLinkedRole`autorisation.

Table des matières

Ajouter des actions d'arrêt aux CloudWatch alarmes Amazon

Vous pouvez créer une alarme qui arrête une instance Amazon EC2 quand un certain seuil a été atteint. Par exemple, vous pouvez exécuter des instances de développement ou de test, et, à l'occasion, oublier de les fermer. Vous pouvez créer une alarme qui est déclenchée quand le pourcentage moyen d'utilisation de l'UC a été inférieur à 10 % pendant 24 heures, indiquant que l'instance est inactive et n'est plus en cours d'utilisation. Vous pouvez ajuster le seuil, la durée et la période en fonction de vos besoins. De plus, vous pouvez ajouter une notification SNS de façon à recevoir un e-mail quand l'alarme est déclenchée.

Les instances Amazon EC2 qui utilisent un volume Amazon Elastic Block Store comme périphérique racine peuvent être arrêtées ou résiliées, tandis que celles qui recourent au stockage d'instance comme périphérique racine peuvent uniquement être résiliées.

Pour créer une alarme afin d'arrêter une instance inactive à l'aide de la CloudWatch console Amazon

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (alertes), All alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Select Metric (Sélectionner une métrique).
5. Pour les AWS namespaces (Espaces de nom), choisissez EC2.
6. Procédez comme suit :

a. Sélectionnez Per-instance Metrics (Métriques par instance).

b. Cochez la case de la ligne avec la bonne instance et la métrique CPUUtilization.

c. Sélectionnez l'onglet Graphed metrics (Graphiques des métriques).

d. Pour les statistiques, choisissez Average (Moyenne).

e. Choisissez une période (par exemple, **`1 Hour`**).

f. Choisissez Select metric (Sélectionner une métrique).

7. Pour l'étape Define Alarm (Définir une alarme), procédez comme suit :

a. Sous Conditions, choisissez Static (Statique).

b. Sous Whenever CPUUtilization is (chaque fois que CPUUtilization est), sélectionnez Lower (Plus bas).

c. Pour than (que), saisissez **`10`**.

d. Choisissez Suivant.

e. Sous Notification, pour Send notification to (Envoyer une notification à), choisissez une rubrique SNS existante ou créez-en une.

Pour créer une rubrique SNS, choisissez New list (Nouvelle liste). Pour Send a notification to (Envoyer une notification à), saisissez un nom pour la rubrique SNS (par exemple, Stop_EC2_Instance). Pour Email list (Liste des adresses e-mail), tapez une liste séparée par des virgules des adresses e-mail pour être informé lorsque l'alarme passe à l'état `ALARM`. Chaque adresse e-mail reçoit un e-mail de confirmation d'abonnement à la rubrique. Vous devez confirmer l'abonnement avant de pouvoir recevoir des notifications à une adresse e-mail.

f. Choisissez Add EC2 Action (Ajouter une action EC2).

g. Pour Alarm state trigger (Déclencheur de l'état d'alarme), choisissez In alarm (En alarme). Pour Take the following action (Effectuer l'action suivante), choisissez Stop this instance (Arrêter cette instance).

h. Choisissez Suivant.

i. Saisissez un nom et une description pour l'alerte. Le nom ne doit contenir que des caractères ASCII. Sélectionnez ensuite Next (Suivant).

j. Dans Preview and create (Prévisualiser et créer), confirmez que les informations et les conditions sont telles que vous les voulez, puis choisissez Create alarm (Créer une alerte).

Ajouter des actions de résiliation aux CloudWatch alarmes Amazon

Vous pouvez créer une alarme qui finit automatiquement une instance EC2 quand un certain seuil a été atteint (aussi longtemps que la protection de fin n'est pas activée pour l'instance). Par exemple, il se peut que vous vouliez mettre hors service une instance quand elle a terminé son travail et que vous n'en avez plus besoin. Si vous souhaitez utiliser l'instance par la suite, vous devez arrêter l'instance, et non y mettre fin. Pour de plus amples informations sur l'activation et la désactivation de la protection de résiliation d'une instance, veuillez consulter Activation de la protection contre la résiliation d'une instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Pour créer une alarme afin de mettre fin à une instance inactive à l'aide de la CloudWatch console Amazon

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (Alarmes), puis Create Alarm (Créer une alarme).
3. Pour l'étape Select Metric (Sélectionner une métrique), procédez comme suit :
   a. Sous EC2 Metrics (Métriques EC2), choisissez Per-Instance Metrics (Métriques par instance).
   b. Sélectionnez la ligne avec l'instance et la métrique CPUUtilization.
   c. Pour les statistiques, choisissez Average (Moyenne).
   d. Choisissez une période (par exemple, **1 Hour**).
   e. Choisissez Next (Suivant).
4. Pour l'étape Define Alarm (Définir une alarme), procédez comme suit :
   a. Sous Alarm Threshold (Seuil de l'alarme), tapez un nom unique pour l'alarme (par exemple, « Mettre hors service l'instance EC2 ») et une description de l'alarme (par exemple, « Mettre hors service l'instance EC2 quand l'UC est inactive trop longtemps »). Les noms d'alarme doivent contenir uniquement des caractères ASCII.
   b. Sous Whenever, pour is, choisissez **<** et entrez **10**. Pour for, tapez **24** périodes consécutives.

      Une représentation graphique du seuil apparaît sous Alarm Preview (Aperçu de l'alarme).
   c. Sous Notification, pour Send notification to (Envoyer une notification à), choisissez une rubrique SNS existante ou créez-en une.

Pour créer une rubrique SNS, choisissez New list (Nouvelle liste). Pour Send a notification to (Envoyer une notification à), saisissez un nom pour la rubrique SNS (par exemple, Terminate_EC2_Instance). Pour Email list (Liste des adresses e-mail), tapez une liste séparée par des virgules des adresses e-mail pour être informé lorsque l'alarme passe à l'état `ALARM`. Chaque adresse e-mail reçoit un e-mail de confirmation d'abonnement à la rubrique. Vous devez confirmer l'abonnement avant de pouvoir recevoir des notifications à une adresse e-mail.

d. Choisissez EC2 Action (Action EC2).

e. Pour Whenever this alarm (Chaque fois que cette alarme), sélectionnez State is ALARM (L'état est ALARME). Pour Take this action (Effectuer cette action), choisissez Terminate this instance (Mettre fin à cette instance).

f. Sélectionnez Create Alarm (Créer une alerte).

Ajouter des actions de redémarrage aux CloudWatch alarmes Amazon

Vous pouvez créer une CloudWatch alarme Amazon qui surveille une instance Amazon EC2 et redémarre automatiquement l'instance. L’action d’alarme de redémarrage est recommandée pour les défaillances de vérification de l’état d’instance (par opposition à l’action d’alarme de récupération, qui convient aux défaillances de la vérification de l’état du système). Le redémarrage d’une instance est similaire à celui d’un système d’exploitation. Dans la plupart des cas, il suffit de quelques minutes pour redémarrer votre instance. Lorsque vous redémarrez une instance, elle reste sur le même hôte physique, ce qui signifie qu’elle conserve son nom DNS public, son adresse IP privée et toutes les données se trouvant sur ses volumes de stockage d’instance.

Le redémarrage d'une instance ne déclenche pas de nouvelle heure de facturation d'instance, contrairement à l'arrêt, puis au redémarrage d'une instance. Pour plus d'informations sur le redémarrage d'une instance, consultez Reboot Your Instance (Redémarrage de votre instance) dans le Guide de l'utilisateur Amazon EC2 pour instances Linux.

> **Important**
>
> Pour prévenir toute condition de concurrence entre les actions de redémarrage et de récupération, évitez de définir la même période d'évaluation pour une alarme de redémarrage et une alarme de récupération. Nous vous recommandons de définir des alarmes de redémarrage sur trois périodes d’évaluation d’une minute chacune.

Pour créer une alarme afin de redémarrer une instance à l'aide de la CloudWatch console Amazon

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (Alarmes), puis Create Alarm (Créer une alarme).
3. Pour l'étape Select Metric (Sélectionner une métrique), procédez comme suit :
   a. Sous EC2 Metrics (Métriques EC2), choisissez Per-Instance Metrics (Métriques par instance).
   b. Sélectionnez la ligne contenant l'instance et la métrique StatusCheckFailed_Instance.
   c. Pour la statistique, choisissez Minimum.
   d. Choisissez une période (par exemple, **`1 Minute`**).
   e. Choisissez Next (Suivant).
4. Pour l'étape Define Alarm (Définir une alarme), procédez comme suit :
   a. Sous Alarm Threshold (Seuil d'alarme), tapez un nom unique pour l'alarme (par exemple, « Redémarrer l'instance EC2 ») et une description de l'alarme (par exemple, « Redémarrer instance EC2 en cas d'échec de la vérification de l'état »). Les noms d'alarme doivent contenir uniquement des caractères ASCII.
   b. Sous Whenever, pour is, choisissez **>** et tapez **`0`**. Pour for, tapez **`3`** périodes consécutives.

      Une représentation graphique du seuil apparaît sous Alarm Preview (Aperçu de l'alarme).
   c. Sous Notification, pour Send notification to (Envoyer une notification à), choisissez une rubrique SNS existante ou créez-en une.

      Pour créer une rubrique SNS, choisissez New list (Nouvelle liste). Pour Send a notification to (Envoyer une notification à), saisissez un nom pour la rubrique SNS (par exemple, Reboot_EC2_Instance). Pour Email list (Liste des adresses e-mail), tapez une liste séparée par des virgules des adresses e-mail pour être informé lorsque l'alarme passe à l'état `ALARM`. Chaque adresse e-mail reçoit un e-mail de confirmation d'abonnement à la rubrique. Vous devez confirmer l'abonnement avant de pouvoir recevoir des notifications à une adresse e-mail.
   d. Choisissez EC2 Action (Action EC2).
   e. Pour Whenever this alarm (Chaque fois que cette alarme), sélectionnez State is ALARM (L'état est ALARME). Pour Take this action (Effectuer cette action), choisissez Reboot this instance (Redémarrer cette instance).

f. Sélectionnez Create Alarm (Créer une alerte).

Ajouter des actions de restauration aux CloudWatch alarmes Amazon

Vous pouvez créer une CloudWatch alarme Amazon qui surveille une instance Amazon EC2 et la récupère automatiquement si elle est endommagée en raison d'une défaillance matérielle sous-jacente ou d'un problème nécessitant une AWS intervention pour être réparée. Les instances mises hors service ne peuvent pas être récupérées. Une instance récupérée est identique à l'instance d'origine, y compris pour l'ID d'instance, les adresses IP privées, les adresses IP Elastic et toutes les métadonnées de l'instance.

Lorsque l'alarme `StatusCheckFailed_System` est déclenchée et que l'action de récupération est initiée, vous en êtes averti par la rubrique Amazon SNS que vous avez choisie quand vous avez créé l'alarme et associé l'action de récupération. Lors de la récupération d'instance, l'instance est migrée pendant un redémarrage d'instance, et toutes les données en mémoire sont perdues. Lorsque le processus est terminé, les informations sont publiées dans la rubrique SNS que vous avez configurée pour l'alarme. Toutes les personnes abonnées à cette rubrique SNS recevront une notification par e-mail qui inclut le statut de la tentative de récupération et les éventuelles instructions supplémentaires. Vous remarquerez un redémarrage d'instance sur l'instance récupérée.

L'action de récupération ne peut être utilisée qu'avec `StatusCheckFailed_System`, pas avec `StatusCheckFailed_Instance`.

Voici quelques exemples de problèmes entraînant l'échec des contrôles d'état du système :

- Perte de connectivité réseau
- Perte d'alimentation système
- Problèmes logiciels sur un hôte physique
- Problèmes matériels sur un hôte physique ayant un impact sur l'accessibilité du réseau

L'action de récupération n'est prise en charge que sur certains types d'instance. Pour plus d'informations sur les types d'instance pris en charge et les autres exigences, consultez Récupérer votre instance et Exigences.

> **Important**
>
> Pour prévenir toute condition de concurrence entre les actions de redémarrage et de récupération, évitez de définir la même période d'évaluation pour une alarme de redémarrage

et une alarme de récupération. Nous vous recommandons de définir des alarmes de récupération sur deux périodes d'évaluation d'une minute chacune et des alarmes de redémarrage sur trois périodes d'évaluation d'une minute chacune.

Pour créer une alarme afin de récupérer une instance à l'aide de la CloudWatch console Amazon

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Alarms (Alarmes), puis Create Alarm (Créer une alarme).
3. Pour l'étape Select Metric (Sélectionner une métrique), procédez comme suit :
   a. Sous EC2 Metrics (Métriques EC2), choisissez Per-Instance Metrics (Métriques par instance).
   b. Sélectionnez la ligne contenant l'instance et la métrique StatusCheckFailed_System.
   c. Pour la statistique, choisissez Minimum.
   d. Choisissez une période (par exemple, **`1 Minute`**).

      ⚠ Important

      Pour prévenir toute condition de concurrence entre les actions de redémarrage et de récupération, évitez de définir la même période d'évaluation pour une alarme de redémarrage et une alarme de récupération. Nous vous recommandons de définir des alarmes de récupération sur deux périodes d'évaluation d'une minute chacune.

   e. Choisissez Next (Suivant).
4. Pour l'étape Define Alarm (Définir une alarme), procédez comme suit :
   a. Sous Alarm Threshold (Seuil d'alarme), tapez un nom unique pour l'alarme (par exemple, « Récupérer l'instance EC2 ») et une description de l'alarme (par exemple, « Récupérer l'instance EC2 en cas d'échec de la vérification de l'état »). Les noms d'alarme doivent contenir uniquement des caractères ASCII.
   b. Sous Whenever, pour is, choisissez **>** et tapez **`0`**. Pour for, tapez **2** périodes consécutives.
   c. Sous Notification, pour Send notification to (Envoyer une notification à), choisissez une rubrique SNS existante ou créez-en une.

Pour créer une rubrique SNS, choisissez New list (Nouvelle liste). Pour Send a notification to (Envoyer une notification à), saisissez un nom pour la rubrique SNS (par exemple, Recover_EC2_Instance). Pour Email list (Liste des adresses e-mail), tapez une liste séparée par des virgules des adresses e-mail pour être informé lorsque l'alarme passe à l'état `ALARM`. Chaque adresse e-mail reçoit un e-mail de confirmation d'abonnement à la rubrique. Vous devez confirmer l'abonnement avant de pouvoir recevoir des notifications à une adresse e-mail.

d. Choisissez EC2 Action (Action EC2).

e. Pour Whenever this alarm (Chaque fois que cette alarme), sélectionnez State is ALARM (L'état est ALARME). Pour Take this action (Effectuer cette action), choisissez Recover this instance (Récupérer cette instance).

f. Sélectionnez Create Alarm (Créer une alerte).

Affichage de l'historique des actions et des alarmes déclenchées

Vous pouvez consulter l'historique des alarmes et des actions dans la CloudWatch console Amazon. Amazon CloudWatch conserve l'historique des alertes et des actions des 30 derniers jours.

Pour afficher l'historique des actions et des alarmes déclenchées

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Alarms (Alarmes), puis sélectionnez une alarme.
3. Pour afficher la transition d'état la plus récente, ainsi que les valeurs de date et de métrique, choisissez Details (Détails).
4. Pour afficher les entrées les plus récentes de l'historique, choisissez History (Historique).

Alarmes et marquage

Les balises sont des paires clé-valeur qui peuvent vous aider à organiser et à classer vos ressources. Vous pouvez également les utiliser pour définir des autorisations utilisateur, en accordant à un utilisateur l'autorisation d'accéder ou de modifier uniquement les ressources avec certaines valeurs de balise. Pour des informations plus générales sur le balisage des ressources, consultez la section Marquage de vos ressources AWS

La liste suivante explique en détail le fonctionnement du balisage avec les CloudWatch alarmes.

- Pour pouvoir définir ou mettre à jour les balises d'une CloudWatch ressource, vous devez être connecté à un compte `cloudwatch:TagResource` autorisé. Par exemple, pour créer une alarme et définir des balises pour celle-ci, vous devez disposer de l'`cloudwatch:TagResource`autorisation en plus de l'`the cloudwatch:PutMetricAlarm` autorisation. Nous vous recommandons de vous assurer que tous les membres de votre organisation qui créeront ou mettront à jour CloudWatch des ressources disposent des `cloudwatch:TagResource` autorisations nécessaires.
- Les balises peuvent être utilisées pour le contrôle d'autorisation basé sur les balises. Par exemple, les autorisations d'utilisateur ou de rôle IAM peuvent inclure des conditions visant à limiter les CloudWatch appels à des ressources spécifiques en fonction de leurs balises. Cependant, gardez à l'esprit les points suivants
  - Les balises dont le nom commence par ne `aws:` peuvent pas être utilisées pour le contrôle d'autorisation basé sur les balises.
  - Les alarmes composites ne prennent pas en charge le contrôle d'autorisation basé sur des balises.

Application Signals

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Utilisez les signaux d' CloudWatch application pour piloter automatiquement vos applications AWS afin de surveiller l'état actuel des applications et de suivre les performances des applications à long terme par rapport à vos objectifs commerciaux. Application Signals vous offre une vue unifiée et orientée application de vos services, dépendances et applications, et vous aide à surveiller et à trier l'état des applications.

- Activez Application Signals pour collecter automatiquement les métriques et les suivis de vos applications, et afficher les métriques clés telles que le volume des appels, la disponibilité, la latence, les pannes et les erreurs. Visualisez et triez rapidement l'état de fonctionnement actuel, et vérifiez si vos applications atteignent leurs objectifs de performance à long terme, sans écrire de code personnalisé ni créer de tableaux de bord.
- Créez et surveillez les objectifs de niveau de service (SLO) avec Application Signals. Créez et suivez facilement l'état des SLO liés aux CloudWatch métriques, y compris les nouvelles métriques d'application standard collectées par Application Signals. Consultez et suivez l'état de l'indicateur de niveau de service (SLI) de vos services d'application dans une liste de services et une carte topologique. Créez des alarmes pour suivre vos SLO et suivez les nouvelles métriques d'application standard collectées par Application Signals.
- Consultez une carte de la topologie de votre application découverte automatiquement par Application Signals, qui vous donne une représentation visuelle de vos applications, de leurs dépendances et de leur connectivité.
- Application Signals fonctionne avec CloudWatch RUM, CloudWatchSynthetics canaries Amazon EC2 Auto Scaling et pour afficher les pages de vos clients AWS Service Catalog AppRegistry, les canaris Synthetics et les noms des applications dans les tableaux de bord et les cartes.

Utilisez Application Signals pour la surveillance quotidienne des applications

Utilisez les signaux d'application dans la CloudWatch console, dans le cadre de la surveillance quotidienne des applications :

1. Si vous avez créé des objectifs de niveau de service (SLO) pour vos services, commencez par la page Objectifs de niveau de service (SLO). Cela vous donne un aperçu immédiat de l'état de vos services et opérations les plus critiques. Choisissez le nom du service ou de l'opération d'un SLO pour ouvrir la page Détails du service et consulter des informations détaillées sur le service lors de la résolution des problèmes.
2. Ouvrez la page Services pour voir un résumé de tous vos services et voir rapidement les services présentant le taux de défaillance ou le temps de latence le plus élevé. Si vous avez créé des SLO, consultez le tableau des services pour voir quels services présentent des indicateurs de niveau de service (SLI) non sains. Si un service particulier est dans un état non sain, sélectionnez-le pour ouvrir la page Détail du service et voir les opérations du service, les dépendances, les scripts canary Synthetics et les demandes des clients. Sélectionnez un point dans un graphique pour voir les suivis corrélés afin de pouvoir résoudre et identifier la cause première des problèmes opérationnels.
3. Si de nouveaux services ont été déployés ou si les dépendances ont changé, ouvrez la Carte des services pour inspecter la topologie de votre application. Consultez une carte de vos applications qui montre la relation entre les clients, les scripts canary Synthetics, les services et les dépendances. Consultez rapidement l'état du SLI, les métriques clés telles que le volume d'appels, le taux de défaillance et la latence, et effectuez une analyse approfondie pour obtenir des informations plus détaillées dans la page de Détails du service.

L'utilisation d'Application Signals entraîne des frais. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

> **Note**
>
> Il n'est pas nécessaire d'activer Application Signals pour utiliser CloudWatch Synthetics CloudWatch , RUM ou Eviently. CloudWatch Cependant, Synthetics CloudWatch et RUM fonctionnent avec Application Signals pour offrir des avantages lorsque vous utilisez ces fonctionnalités ensemble.

Langages et architectures pris en charge

Actuellement, Application Signals prend en charge les applications Java et Python.

Application Signals est pris en charge et testé sur Amazon EKS, Amazon ECS et Amazon EC2. Sur les clusters Amazon EKS, il découvre automatiquement les noms de vos services et clusters. Sur les

autres architectures, vous devez fournir les noms des services et des environnements lorsque vous activez ces services pour Application Signals.

Les instructions d'activation des signaux d'application sur Amazon EC2 doivent fonctionner sur toute architecture prenant en charge l' CloudWatch agent et AWS la distribution pour. OpenTelemetry Toutefois, les instructions n’ont pas été testées sur des architectures autres qu’Amazon ECS et Amazon EC2.

Régions prises en charge

Pour cette version préliminaire, Application Signals est pris en charge dans les régions suivantes.

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Oregon)
- Asie-Pacifique (Sydney)
- Asia Pacific (Tokyo)
- Europe (Ireland)

Aperçu du kit SDK

Une version préliminaire du SDK est disponible en téléchargement.

> **Warning**
>
> Les opérations et les paramètres de l’API sont susceptibles d’être modifiés avant que Application Signals ne soit disponible de manière globale. Ces changements pourraient être majeurs. N’utilisez pas la version préliminaire du SDK à des fins de production.

Pour installer le SDK de prévisualisation, installez ou mettez à jour d'abord la dernière version de la AWS CLI version 2. Pour plus d’informations, veuillez consulter la rubrique Install or update the latest version of the AWS CLI.

Utilisez ensuite les commandes suivantes pour télécharger le fichier zip du SDK depuis le compartiment Amazon S3, puis extrayez son contenu. Chaque fichier zip du kit SDK contient les instructions du kit SDK et la documentation de l’API.

> **Note**
>
> Le SDK est fourni dans plusieurs langages de programmation afin que vous puissiez utiliser les API Application Signals avec n'importe lequel de ces langages de programmation. Toutefois, l'instrumentation automatique de votre application pour envoyer des données à Application Signals n'est prise en charge que pour les applications Java et Python.

- Kit SDK Java V2 : `aws s3 cp s3://application-signals-preview-sdk/awsJavaSdkV2.zip ./`
- JavaScript SDK V3 : `aws s3 cp s3://application-signals-preview-sdk/jsSdkV3.zip ./`
- JavaScript SDK V2 : `aws s3 cp s3://application-signals-preview-sdk/jsSdkV2.zip ./`
- Kit SDK Python : `aws s3 cp s3://application-signals-preview-sdk/pythonSdk.zip ./`
- Kit SDK Kotlin : `aws s3 cp s3://application-signals-preview-sdk/kotlin.zip ./`
- Kit SDK Android : `aws s3 cp s3://application-signals-preview-sdk/andriod.zip ./`
- Kit SDK C++ : `aws s3 cp s3://application-signals-preview-sdk/awsCppSdk.zip ./`
- Kit SDK PHP : `aws s3 cp s3://application-signals-preview-sdk/awsSdkPhp.zip ./`
- Kit SDK Ruby : `aws s3 cp s3://application-signals-preview-sdk/awsSdkRuby.zip ./`
- Kit SDK Go V2 : `aws s3 cp s3://application-signals-preview-sdk/awsSdkGoV2.zip ./`
- Kit SDK Go V1 : `aws s3 cp s3://application-signals-preview-sdk/go.zip ./`
- Kit SDK iOS : `aws s3 cp s3://application-signals-preview-sdk/iOS.zip ./`

Rubriques

- Objectifs de niveau de service (SLO)
- Surveillez l'état de fonctionnement de vos applications avec Application Signals
- Métriques d'application standard collectées
- Utiliser une surveillance synthétique
- Réalisez des lancements et des expériences A/B avec Evidently CloudWatch
- Utiliser du CloudWatch rhum

Autorisations requises pour Application Signals

⚠ Application Signals est en version préliminaire pour Amazon CloudWatch et est susceptible d'être modifiée.

Cette section explique les autorisations nécessaires pour activer, gérer et utiliser Application Signals.

Autorisations pour activer et gérer Application Signals

Pour gérer les signaux d'application, vous devez être connecté avec les autorisations suivantes :

```
{
 "Version": "2012-10-17",
 "Statement": [
    {
        "Sid": "CloudWatchApplicationSignalsFullAccessPermissions",
        "Effect": "Allow",
        "Action": "application-signals:*",
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsAlarmsPermissions",
        "Effect": "Allow",
        "Action": [
            "cloudwatch:DescribeAlarms"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsMetricsPermissions",
```

```
        "Effect": "Allow",
        "Action": [
            "cloudwatch:GetMetricData",
            "cloudwatch:ListMetrics"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSyntheticsPermissions",
        "Effect": "Allow",
        "Action": [
            "synthetics:DescribeCanaries",
            "synthetics:DescribeCanariesLastRun",
            "synthetics:GetCanaryRuns"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsRumPermissions",
        "Effect": "Allow",
        "Action": [
            "rum:BatchCreateRumMetricDefinitions",
            "rum:BatchDeleteRumMetricDefinitions",
            "rum:BatchGetRumMetricDefinitions",
            "rum:GetAppMonitor",
            "rum:GetAppMonitorData",
```

```
            "rum:ListAppMonitors",
            "rum:PutRumMetricsDestination",
            "rum:UpdateRumMetricDefinition"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsXrayPermissions",
        "Effect": "Allow",
        "Action": [
            "xray:GetTraceSummaries"
        ],
        "Resource": "*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsPutMetricAlarmPermissions",
        "Effect": "Allow",
        "Action": "cloudwatch:PutMetricAlarm",
        "Resource": [
            "arn:aws:cloudwatch:*:*:alarm:SLO-AttainmentGoalAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLO-WarningAlarm-*",
            "arn:aws:cloudwatch:*:*:alarm:SLI-HealthAlarm-*"
        ]
    },
    {
        "Sid": "CloudWatchApplicationSignalsCreateServiceLinkedRolePermissions",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
        "Condition": {
            "StringLike": {
                "iam:AWSServiceName": "application-signals.cloudwatch.amazonaws.com"
            }
        }
    },
    {
        "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
        "Effect": "Allow",
        "Action": "iam:GetRole",
        "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
    },
    {
```

```
        "Sid": "CloudWatchApplicationSignalsSnsWritePermissions",
        "Effect": "Allow",
        "Action": [
            "sns:CreateTopic",
            "sns:Subscribe"
        ],
        "Resource": "arn:aws:sns:*:*:cloudwatch-application-signals-*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsSnsReadPermissions",
        "Effect": "Allow",
        "Action": "sns:ListTopics",
        "Resource": "*"
    }
 ]
}
```

Pour activer les signaux d'application sur Amazon EC2 ou sur des architectures personnaliséesKubernetes, consultez Activer les signaux d'application sur d'autres plateformes avec une configuration personnalisée. Pour activer et gérer les signaux d'application sur Amazon EKS à l'aide du module complémentaire Amazon CloudWatch Observability EKS, vous devez disposer des autorisations suivantes.

> **Important**
>
> Ces autorisations incluent `iam:PassRole` avec `Resource "*"` et `eks:CreateAddon` avec `Resource "*"`. Ces autorisations sont puissantes et vous devez faire preuve de prudence lorsque vous les accordez.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
"Sid": "CloudWatchApplicationSignalsEksAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:AccessKubernetesApi",
                "eks:CreateAddon",
                "eks:DescribeAddon",
                "eks:DescribeAddonConfiguration",
```

```
                "eks:DescribeAddonVersions",
                "eks:DescribeCluster",
                "eks:DescribeUpdate",
                "eks:ListAddons",
                "eks:ListClusters",
                "eks:ListUpdates",
                "iam:ListRoles",
                "iam:PassRole"
            ],
            "Resource": "*"
        },
        {
"Sid":
 "CloudWatchApplicationSignalsEksCloudWatchObservabilityAddonManagementPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DeleteAddon",
                "eks:UpdateAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
        }
    ]
}
```

Le tableau de bord des signaux d'application affiche les AWS Service Catalog AppRegistry applications auxquelles vos SLO sont associés. Pour voir ces applications dans les pages SLO, vous devez disposer des autorisations suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Exploitation d’Application Signals

Les opérateurs de services qui utilisent les signaux d'application pour surveiller les services et les SLO doivent être connectés à un compte avec les autorisations de lecture seule suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-signals:BatchGet*",
                "application-signals:Get*",
                "application-signals:List*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        },
        {
        "Sid": "CloudWatchApplicationSignalsLogGroupPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:StartQuery",
            "logs:DescribeMetricFilters"
        ],
        "Resource": "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
    },
    {
        "Sid": "CloudWatchApplicationSignalsLogsPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:GetQueryResults",
            "logs:StopQuery"
        ],
        "Resource": "*"
    },
```

```
        {
            "Sid": "CloudWatchApplicationSignalsAlarmsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsMetricsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsSyntheticsReadPermissions",
            "Effect": "Allow",
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:DescribeCanariesLastRun",
                "synthetics:GetCanaryRuns"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsRumReadPermissions",
            "Effect": "Allow",
            "Action": [
                "rum:BatchGetRumMetricDefinitions",
                "rum:GetAppMonitor",
                "rum:GetAppMonitorData",
                "rum:ListAppMonitors"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsXrayReadPermissions",
            "Effect": "Allow",
            "Action": [
                "xray:GetTraceSummaries"
            ],
```

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

Pour voir à quelles AWS Service Catalog AppRegistry applications vos SLO sont associées dans le tableau de bord des signaux d'application, vous devez disposer des autorisations suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsTaggingReadPermissions",
            "Effect": "Allow",
            "Action": "tag:GetResources",
            "Resource": "*"
        }
    ]
}
```

Pour vérifier si les signaux d'application sur Amazon EKS utilisant le module complémentaire Amazon CloudWatch Observability EKS sont activés, vous devez disposer des autorisations suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchApplicationSignalsEksReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:ListAddons",
                "eks:ListClusters"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsEksDescribeAddonReadPermissions",
            "Effect": "Allow",
            "Action": [
                "eks:DescribeAddon"
            ],
            "Resource": "arn:aws:eks:*:*:addon/*/amazon-cloudwatch-observability/*"
```

```
        }
    ]
}
```

Activer Application Signals

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Les rubriques de cette section expliquent comment activer les signaux CloudWatch d'application dans votre environnement. Application Signals est pris en charge sur les clusters Amazon EKS avec un flux de travail de configuration utilisant la console. Il est également pris en charge sur d'autres plateformes, notamment Amazon EC2, avec un processus de configuration personnalisé.

Rubriques

Systèmes compatibles avec Application Signals

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Application Signals est pris en charge et testé sur Amazon EKS, Amazon ECS et Amazon EC2. Les instructions d'activation des signaux d'application sur Amazon EC2 devraient fonctionner sur toutes

les plateformes compatibles avec l' CloudWatch agent et AWS Distro OpenTelemetry, mais elles n'ont pas été testées sur d'autres plateformes.

Java compatibility

Application Signals prend en charge les applications Java et prend en charge les mêmes bibliothèques et frameworks Java que AWS Distro for OpenTelemetry . Pour plus d'informations, veuillez consulter la rubrique Bibliothèques, frameworks, serveurs d'applications et JVM pris en charge.

Les versions 8, 11 et 17 de JVM sont prises en charge.

Compatibilité avec Python

Application Signals prend en charge les mêmes bibliothèques et frameworks que AWS Distro for OpenTelemetry . Pour plus d'informations, consultez la section Packages pris en charge à l'adresse opentelemetry-python-contrib.

Les versions 3.8 et ultérieures de Python sont prises en charge.

Avant d'activer les signaux d'application pour vos applications Python, tenez compte des points suivants.

- Dans certaines applications conteneurisées, `PYTHONPATH` l'absence d'une variable d'environnement peut parfois empêcher l'application de démarrer. Pour résoudre ce problème, veillez à définir la variable d'`PYTHONPATH`environnement à l'emplacement du répertoire de travail de votre application. Cela est dû à un problème connu lié à l' OpenTelemetry auto-instrumentation. Pour plus d'informations sur ce problème, consultez la section Le paramètre d'auto-instrumentation Python de PYTHONPATH n'est pas conforme.
- Pour les applications Django, des configurations supplémentaires sont requises, qui sont décrites dans la documentation OpenTelemetry Python.
  - Utilisez le `--noreload` drapeau pour empêcher le rechargement automatique.
  - Définissez la variable d'`DJANGO_SETTINGS_MODULE`environnement sur l'emplacement du `settings.py` fichier de votre application Django. Cela garantit que OpenTelemetry vous pouvez accéder et intégrer correctement vos paramètres Django.

OpenTelemetry considérations relatives à la compatibilité

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Pour intégrer les signaux d' CloudWatch application à vos applications, nous vous recommandons de supprimer complètement toute solution de surveillance des performances des applications existante de votre application au préalable. Cela inclut la suppression de tout code d'instrumentation et de toute configuration.

Même si Application Signals utilise de l' OpenTelemetry instrumentation, sa compatibilité avec votre OpenTelemetry instrumentation ou configuration existante n'est pas garantie. Dans le meilleur des cas, vous pourrez peut-être conserver certaines de vos OpenTelemetry fonctionnalités, telles que les métriques personnalisées. Veillez toutefois à lire les sections suivantes pour obtenir des détails.

Considérations à prendre en compte si vous utilisez déjà OpenTelemetry

Si vous l'utilisez déjà OpenTelemetry avec votre application, le reste de cette section contient des informations importantes pour garantir la compatibilité avec les signaux d'application.

- Avant d'activer votre application pour Application Signals, vous devez supprimer l'injection de tout autre agent d'auto-instrumentation basé sur OpenTelemetry votre application. Cela permet d'éviter les conflits de configuration. Vous pouvez continuer à utiliser l'instrumentation manuelle à l'aide d' OpenTelemetry API compatibles et de signaux d'application.
- Si vous utilisez une instrumentation manuelle pour générer des intervalles ou des métriques personnalisés à partir de votre application, en fonction de la complexité de l'instrumentation, l'activation d'Application Signals peut les empêcher de générer des données ou avoir un autre comportement indésirable. Vous pourrez peut-être utiliser certaines des configurations disponibles dans OpenTelemetry (à l'exception de celles mentionnées dans le tableau plus loin dans cette section) pour conserver le comportement souhaité de vos métriques ou spans existants. Pour plus d'informations sur ces configurations, consultez la section Configuration du SDK dans la OpenTelemetry documentation.

  Par exemple, en utilisant la `OTEL_EXPORTER_OTLP_METRICS_ENDPOINT` configuration et une instance OpenTelemetry Collector autogérée, vous pourrez peut-être continuer à envoyer vos métriques personnalisées vers la destination souhaitée.

- Certaines variables d’environnement ou propriétés système ne doivent pas être utilisées avec Application Signals, tandis que vous pouvez en utiliser d’autres à condition de suivre les instructions du tableau. Consultez le tableau suivant pour plus d’informations.

Variable d'environnement	Recommandation avec Application Signals
Variables d'environnement générales	
`OTEL_SDK_DISABLED`	Ne doit pas être défini sur `true`.
`OTEL_TRACES_EXPORTER`	Doit avoir la valeur `otlp`.
`OTEL_EXPORTER_OTLP_ENDPOINT`	Ne doit pas être utilisé.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Ne doit pas être utilisé.
`OTEL_ATTRIBUTE_COUNT_LIMIT`	S'il est défini, il doit être défini suffisamment haut pour inclure environ 10 attributs d'étendue supplémentaires ajoutés par CloudWatch Application Signals.
`OTEL_PROPAGATORS`	Si défini, `xray` doit être inclus pour le suivi final.
`OTEL_TRACES_SAMPLER`	Si défini, doit être `xray` pour utiliser l’échantillonnage centralisé X-Ray. Pour utiliser l’échantillonnage local, définissez-le sur `parentbased_traceidratio` et spécifiez le taux d’échantillonnage dans `OTEL_TRACES_SAMPLER_ARG` .
`OTEL_TRACES_SAMPLER_ARG`	Si vous utilisez l’échantillon de suivi centralisé X-Ray par défaut, cette variable ne doit pas être utilisée. Si vous utilisez plutôt l’échantillonnage local, définissez le taux d’échantillonnage dans cette

Variable d'environnement	Recommandation avec Application Signals
	variable. Par exemple, `0.05` pour un taux d'échantillonnage de 5 %.
Variables d'environnement spécifiques à Java	
`OTEL_JAVA_ENABLED_RESOURCE_PROVIDERS`	Si cette option est définie, les détecteurs de AWS ressources doivent être inclus.
Variables d'environnement spécifiques à Python	
`OTEL_PYTHON_CONFIGURATOR`	S'il est utilisé, doit être réglé sur `aws_configurator`
`OTEL_PYTHON_DISTRO`	S'il est utilisé, doit être réglé sur `aws_distro`

Activation d'Application Signals sur les clusters Amazon EKS

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

CloudWatch Application Signals est pris en charge pour les applications Java et Python exécutées dans des clusters Amazon EKS. Pour activer Application Signals pour les applications d'un cluster Amazon EKS, deux options s'offrent à vous :

- Pour activer Application Signals pour vos applications sur un cluster Amazon EKS existant, suivez les étapes décrites dans Activation d'Application Signals sur un cluster Amazon EKS avec vos services.
- Pour tester Application Signals dans un environnement hors production avec un exemple d'application, suivez les instructions fournies dans Activation d'Application Signals sur un nouveau cluster Amazon EKS avec un exemple d'application. Ce flux de travail utilise des scripts fournis par AWS pour créer un nouveau cluster Amazon EKS et installer un exemple d'application activé

pour Application Signals. Cela vous permet de voir et de tester le end-to-end fonctionnement des signaux d'application.

Rubriques

Activation d’Application Signals sur un cluster Amazon EKS avec vos services

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Pour activer les signaux CloudWatch d'application sur vos applications sur un cluster Amazon EKS existant, suivez les instructions de cette section.

⚠ Important

Si vous utilisez OpenTelemetry déjà une application que vous avez l'intention d'activer pour les signaux d'application, consultez la section OpenTelemetry considérations relatives à la compatibilité avant d'activer les signaux d'application.

Pour activer Application Signals pour vos applications sur un cluster Amazon EKS existant

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Services.
3. Si vous n’avez pas encore activé Application Signals dans ce compte, vous devez accorder à Application Signals les autorisations nécessaires pour découvrir vos services. Pour ce faire, procédez comme suit. Cette opération ne doit être effectuée qu’une seule fois par compte.
   a. Choisissez Commencer à découvrir vos services.
   b. Cochez la case et choisissez Commencer à découvrir les services.

Si vous effectuez cette étape pour la première fois dans votre compte, le rôle AWSServiceRoleForCloudWatchApplicationSignalslié au service est créé. Ce rôle accorde à Application Signals les autorisations suivantes :

- `xray:GetServiceGraph`
- `logs:StartQuery`
- `logs:GetQueryResults`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Pour plus d'informations sur ce rôle, consultez Autorisations de rôle liées au service pour les signaux d'application CloudWatch .

4. Choisissez Activer Application Signals.
5. Pour Spécifier la plateforme, choisissez EKS.
6. Pour Sélectionner un cluster EKS, sélectionnez le cluster dans lequel vous souhaitez activer Application Signals.
7. Si le module complémentaire Amazon CloudWatch Observability EKS n'est pas encore activé sur ce cluster, vous êtes invité à l'activer. Dans ce cas, vous pouvez procéder de l'une des façons suivantes :

   a. Choisissez Ajouter le module complémentaire CloudWatch Observability EKS. La console Amazon EKS apparaît.

   b. Cochez la case Amazon CloudWatch Observability et choisissez Next.

      Le module complémentaire CloudWatch Observability EKS active à la fois les signaux d'application et les informations sur les CloudWatch conteneurs avec une observabilité améliorée pour Amazon EKS. Pour plus d'informations sur Container Insights, consultez Container Insights.

   c. Sélectionnez la version la plus récente du module complémentaire à installer.

   d. Sélectionnez un rôle IAM à utiliser pour le module complémentaire. Si vous choisissez Hériter du nœud, attachez les autorisations appropriées au rôle IAM utilisé par vos composants master. *`my-worker-node-role`*Remplacez-le par le rôle IAM utilisé par vos nœuds de travail Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
--policy-arn arn:aws:iam::aws:policy/AWSXRayWriteOnlyAccess
```

e. Si vous souhaitez créer une fonction du service pour utiliser le module complémentaire, veuillez consulter Installez l' CloudWatch agent à l'aide du module complémentaire Amazon CloudWatch Observability EKS.

f. Choisissez Suivant, confirmez les informations affichées à l'écran, puis choisissez Créer.

g. Dans l'écran suivant, choisissez Enable CloudWatch Application Signals pour revenir à la CloudWatch console et terminer le processus.

8. Il existe deux options pour activer vos applications pour les signaux d'application. Pour des raisons de cohérence, nous vous recommandons de choisir une option par cluster.

- L'option Console est plus simple. L'utilisation de cette méthode entraîne le redémarrage immédiat de vos pods.
- La méthode Annotate Manifest File vous permet de mieux contrôler le moment où vos pods redémarrent et peut également vous aider à gérer votre surveillance de manière plus décentralisée si vous ne souhaitez pas la centraliser.

Console

L'option Console utilise la configuration avancée du module complémentaire Amazon CloudWatch Observability EKS pour configurer les signaux d'application pour vos services. Pour plus d'informations sur le module complémentaire, consultez(Facultatif) Configuration supplémentaire.

Si la liste des charges de travail et des espaces de noms ne s'affiche pas, assurez-vous de disposer des autorisations appropriées pour les consulter pour ce cluster. Pour plus d'informations, consultez la section Autorisations requises.

Vous pouvez surveiller des charges de travail uniques ou des espaces de noms entiers.

Pour surveiller une seule charge de travail :

1. Cochez la case correspondant à la charge de travail que vous souhaitez surveiller.

2. Sélectionnez la langue de la charge de travail. Pour les applications Python, assurez-vous que votre application respecte les prérequis requis avant de continuer. Pour plus d'informations, consultez L'application Python ne démarre pas une fois les signaux d'application activés.
3. Sélectionnez Exécuté. Le module complémentaire Amazon CloudWatch Observability EKS injectera immédiatement les SDK AWS Distro for OpenTelemetry Autoinstrumentation (ADOT) dans vos pods et déclenchera le redémarrage des pods pour permettre la collecte de métriques et de traces d'applications.

Pour surveiller l'intégralité d'un espace de noms :

1. Cochez la case correspondant à l'espace de noms que vous souhaitez surveiller.
2. Sélectionnez la langue de la charge de travail. Cela s'applique à toutes les charges de travail de cet espace de noms, qu'elles soient actuellement déployées ou qu'elles le soient dans le futur. Pour les applications Python, assurez-vous que votre application respecte les prérequis requis avant de continuer. Pour plus d'informations, consultez L'application Python ne démarre pas une fois les signaux d'application activés.
3. Sélectionnez Exécuté. Le module complémentaire Amazon CloudWatch Observability EKS injectera immédiatement les SDK AWS Distro for OpenTelemetry Autoinstrumentation (ADOT) dans vos pods et déclenchera le redémarrage des pods pour permettre la collecte de métriques et de traces d'applications.

Pour activer Application Signals dans un autre cluster Amazon EKS, choisissez Activer Application Signals dans l'écran Services.

Annotate manifest file

Dans la CloudWatch console, la section Monitor Services explique que vous devez ajouter une annotation à un manifeste YAML dans le cluster. L'ajout de cette annotation permet à l'application d'envoyer automatiquement des métriques, des suivis et des journaux à Application Signals.

Vous avez deux options pour l'annotation :

- Annoter une charge de travail instrumente automatiquement une seule charge de travail dans le cluster.

- Annoter l’espace de noms permet d’instrumenter automatiquement toutes les charges de travail déployées dans l’espace de noms sélectionné.

Choisissez l’une de ces options, puis suivez les étapes appropriées :

- Pour annoter une seule charge de travail :
  1. Choisissez Annoter la charge de travail.
  2. Collez l'une des lignes suivantes dans la `PodTemplate` section du fichier manifeste de charge de travail.
     - Pour les charges de travail Java : `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
     - Pour les charges de travail en Python : `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

       Pour les applications Python, des configurations supplémentaires sont requises. Pour plus d’informations, consultez L'application Python ne démarre pas une fois les signaux d'application activés.
  3. Dans votre terminal, saisissez `kubectl apply -f your_deployment_yaml` pour appliquer la modification.
- Pour annoter toutes les charges de travail dans un espace de noms :
  1. Choisissez Annoter l’espace de noms.
  2. Collez l'une des lignes suivantes dans la section des métadonnées du fichier manifeste de l'espace de noms. Si l'espace de noms inclut à la fois des charges de travail Java et Python, collez ces deux lignes dans le fichier manifeste de l'espace de noms.
     - S'il existe des charges de travail Java dans l'espace de noms : `annotations: instrumentation.opentelemetry.io/inject-java: "true"`
     - S'il existe des charges de travail Python dans l'espace de noms : `annotations: instrumentation.opentelemetry.io/inject-python: "true"`

       Pour les applications Python, des configurations supplémentaires sont requises. Pour plus d’informations, consultez L'application Python ne démarre pas une fois les signaux d'application activés.
  3. Dans votre terminal, saisissez `kubectl apply -f your_namespace_yaml` pour appliquer la modification.

4. Dans votre terminal, saisissez une commande pour redémarrer tous les pods de l'espace de noms. Voici un exemple de commande pour redémarrer les charges de travail de déploiement : `kubectl rollout restart deployment -n namespace_name`

9. Choisissez Afficher les services lorsque vous avez terminé. Cela vous amène à la vue Services d'Application Signals, où vous pouvez voir les données collectées par Application Signals. Les données peuvent prendre quelques minutes pour s'afficher.

   Pour activer Application Signals dans un autre cluster Amazon EKS, choisissez Activer Application Signals dans l'écran Services.

   Pour plus d'informations sur la vue Services, veuillez consulter Surveillez l'état de fonctionnement de vos applications avec Application Signals.

> **Note**
>
> Nous avons identifié certaines considérations que vous devez garder à l'esprit lorsque vous activez des applications Python pour les signaux d'application. Pour plus d'informations, consultez L'application Python ne démarre pas une fois les signaux d'application activés.

Activation d'Application Signals sur un nouveau cluster Amazon EKS avec un exemple d'application

> **Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.**

Pour tester CloudWatch Application Signals sur un exemple d'application avant de l'utiliser pour vos propres applications, suivez les instructions de cette section. Ces instructions utilisent des scripts pour vous aider à créer un cluster Amazon EKS, à installer un exemple d'application et à équiper l'exemple d'application pour qu'il fonctionne avec Application Signals.

L’exemple d’application est une application Spring « Pet Clinic » composée de quatre microservices. Ces services s'exécutent sur Amazon EKS sur Amazon EC2 et exploitent les scripts d'activation des signaux d'application pour activer le cluster avec l'agent d'auto-instrumentation Java ou Python.

Prérequis

- Actuellement, Application Signals surveille uniquement les applications Java et Python.
- Vous devez l'avoir AWS CLI installé sur l'instance. Nous recommandons AWS CLI la version 2, mais la version 1 devrait également fonctionner. Pour plus d'informations sur l'installation du AWS CLI, voir Installer ou mettre à jour la dernière version du AWS CLI.
- Les scripts de cette section sont destinés à être exécutés dans des environnements Linux et macOS. Pour les instances Windows, nous vous recommandons d'utiliser un AWS Cloud9 environnement pour exécuter ces scripts. Pour plus d'informations AWS Cloud9, voir Qu'est-ce que c'est AWS Cloud9 ? .
- Installez une version prise en charge de `kubectl`. Vous devez utiliser une version de `kubectl` qui se situe à une différence de version mineure près de celle du plan de contrôle de votre cluster Amazon EKS. Par exemple, un client `kubectl` 1.26 fonctionne avec les clusters Kubernetes 1.25, 1.26 et 1.27. Si vous possédez déjà un cluster Amazon EKS, vous devrez peut-être configurer les AWS informations d'identification pour`kubectl`. Pour plus d’informations, veuillez consulter la rubrique Création ou mise à jour d’un fichier `kubeconfig` pour un cluster Amazon EKS.
- Installez`eksctl`. `eksctl`utilise le AWS CLI pour interagir avec AWS, ce qui signifie qu'il utilise les mêmes AWS informations d'identification que le AWS CLI. Pour plus d’informations, veuillez consulter la rubrique Installation ou mise à jour de `eksctl`.
- Installez `jq`. `jq` est nécessaire pour exécuter les scripts d’activation d’Application Signals. Pour plus d’informations, veuillez consulter la rubrique Télécharger jq.

Étape 1 : télécharger les scripts

Pour télécharger les scripts permettant de configurer CloudWatch Application Signals avec un exemple d'application, vous pouvez télécharger et décompresser le fichier de GitHub projet compressé sur un lecteur local, ou vous pouvez cloner le GitHub projet.

Pour cloner le projet, ouvrez une fenêtre de terminal et saisissez la commande Git suivante dans un répertoire de travail donné.

```
git clone https://github.com/aws-observability/application-signals-demo.git
```

Étape 2 : générer et déployer l'exemple d'application

Pour générer et diffuser les exemples d'images d'application, suivez ces instructions.

Étape 3 : déployer et activer Application Signals et l'exemple d'application

Assurez-vous d'avoir rempli les exigences listées dans Activation d'Application Signals sur un nouveau cluster Amazon EKS avec un exemple d'application avant de suivre les étapes suivantes.

Pour déployer et activer Application Signals et l'exemple d'application

1. Saisissez la commande suivante dans le terminal local où vous avez décompressé le script d'intégration. *new-cluster-name*Remplacez-le par le nom que vous souhaitez utiliser pour le nouveau cluster. Remplacez *region-name* par le nom de la AWS région, tel que. `us-west-1`

   Cette commande configure l'exemple d'application s'exécutant dans un nouveau cluster Amazon EKS avec Application Signals activés.

   ```
   # assuming the current working directory is 'onboarding'
   # this script sets up a new cluster, enables Application Signals, and deploys the
   # sample application
   cd application-signals-demo/scripts/eks/appsignals/one-step && ./setup.sh new-
   cluster-name region-name
   ```

   L'exécution du script d'installation prend environ 30 minutes et effectue les opérations suivantes :

   - création d'un cluster Amazon EKS dans la région spécifiée ;
   - création des autorisations IAM nécessaires pour Application Signals (`arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess` et `arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy`) ;
   - Active les signaux d'application en installant l' CloudWatch agent et en instrumentant automatiquement l'exemple d'application pour les CloudWatch métriques et les traces X-Ray.
   - Déploie l'exemple d'application PetClinic Spring dans le même cluster Amazon EKS.
   - Crée cinq CloudWatch canaris Synthetics, `pc-add-vist` nomméspc-create-owners,,,,. `pc-visit-pet` `pc-visit-vet` `pc-clinic-traffic` Ces scripts canary fonctionneront à une fréquence d'une minute afin de générer du trafic synthétique pour l'exemple d'application et de montrer comment les scripts canary Synthetics apparaissent dans Application Signals ;
   - Crée quatre objectifs de niveau de service (SLO) pour l' PetClinic application portant les noms suivants :

    - Disponibilité pour la recherche d’un propriétaire
    - Latence lors de la recherche d’un propriétaire
    - Disponibilité pour l’enregistrement d’un propriétaire
    - Latence lors de l’enregistrement d’un propriétaire
  - création du rôle IAM requis avec une politique de confiance personnalisée accordant à Application Signals les autorisations suivantes :
    - `cloudwatch:PutMetricData`
    - `cloudwatch:GetMetricData`
    - `xray:GetServiceGraph`
    - `logs:StartQuery`
    - `logs:GetQueryResults`
2. (Facultatif) Si vous souhaitez consulter le code source de l' PetClinic exemple d'application, vous pouvez le trouver dans le dossier racine.

```
- application-signals-demo
  - spring-petclinic-admin-server
  - spring-petclinic-api-gateway
  - spring-petclinic-config-server
  - spring-petclinic-customers-service
  - spring-petclinic-discovery-server
  - spring-petclinic-vets-service
  - spring-petclinic-visits-service
```

3. Pour afficher l' PetClinic exemple d'application déployé, exécutez la commande suivante pour trouver l'URL :

```
kubectl get ingress
```

Étape 4 : surveiller l’exemple d’application

Après avoir effectué les étapes décrites dans la section précédente pour créer le cluster Amazon EKS et déployer l’exemple d’application, vous pouvez utiliser Application Signals pour surveiller l’application.

> **Note**
>
> Pour que la console Application Signals commence à se remplir, une partie du trafic doit atteindre l'exemple d'application. Les étapes précédentes ont notamment permis de créer des canaris CloudWatch Synthetics qui génèrent du trafic vers l'exemple d'application.

Surveillance de l'intégrité du service

Une fois activé, CloudWatch Application Signals découvre et remplit automatiquement une liste de services sans nécessiter de configuration supplémentaire.

Pour consulter la liste des services découverts et surveiller leur état

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation du service, choisissez Application Signals, Services.
3. Pour afficher un service, ses opérations et ses dépendances, choisissez le nom de l'un des services de la liste.

   Cette vue unifiée centrée sur les applications permet d'avoir une vision complète de la manière dont les utilisateurs interagissent avec votre service. Cela peut vous aider à trier les problèmes en cas d'anomalies de performances. Pour plus de détails sur la vue Services, veuillez consulter Surveillez l'état de fonctionnement de vos applications avec Application Signals.
4. Choisissez l'onglet Opérations de service pour voir les métriques d'application standard pour les opérations de ce service. Les opérations sont, par exemple, les opérations d'API que le service appelle.

   Ensuite, pour afficher les graphiques d'une seule opération de ce service, choisissez le nom de cette opération.
5. Cliquez sur l'onglet Dépendances pour voir les dépendances de votre application, ainsi que les mesures d'application critiques pour chaque dépendance. Les dépendances incluent AWS les services et les services tiers appelés par votre application.
6. Pour afficher les suivis corrélés depuis la page des détails du service, choisissez un point de données dans l'un des trois graphiques situés au-dessus du tableau. Cela remplit un nouveau volet avec les suivis filtrés de la période. Ces suivis sont triés et filtrés en fonction du graphique que vous avez choisi. Par exemple, si vous avez choisi le graphique de latence, les suivis sont triés en fonction du temps de réponse du service.

7. Dans le volet de navigation de la CloudWatch console, choisissez SLO. Vous pouvez voir les SLO créés par le script pour l'exemple d'application. Pour plus d'informations sur les SLO, veuillez consulter Objectifs de niveau de service (SLO).

(Facultatif) Étape 5 : nettoyage

Lorsque vous avez terminé de tester Application Signals, vous pouvez utiliser un script fourni par Amazon pour nettoyer et supprimer les artefacts créés dans votre compte pour l'exemple d'application. Pour effectuer le nettoyage, saisissez la commande suivante. Remplacez *new-cluster-name*par le nom du cluster que vous avez créé pour l'exemple d'application, et remplacez *region* -name par le nom de la AWS région, tel queus-west-1.

```
cd application-signals-demo/scripts/eks/appsignals/one-step && ./cleanup.sh new-
cluster-name region-name
```

Activation d'Application Signals sur d'autres plateformes avec une configuration personnalisée

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Activez les signaux d' CloudWatch application sur des plateformes autres qu'Amazon EKS en suivant les étapes de configuration personnalisées décrites dans ces sections. Sur ces architectures, vous installez et configurez OpenTelemetry vous-même l' CloudWatch agent et AWS Distro.

Sur ces architectures, Application Signals ne découvre pas automatiquement les noms de vos services, de leurs clusters ou de leurs hôtes. Vous devez spécifier ces noms lors de la configuration personnalisée, et les noms que vous spécifiez sont ceux qui sont affichés sur les tableaux de bord d'Application Signals.

Rubriques

- Utilisation d'une configuration personnalisée pour activer Application Signals sur Amazon ECS
- Utilisation d'une configuration personnalisée pour activer Application Signals sur Amazon EC2 et d'autres plateformes

Utilisation d’une configuration personnalisée pour activer Application Signals sur Amazon ECS

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Utilisez ces instructions de configuration personnalisées pour intégrer vos applications sur Amazon ECS à CloudWatch Application Signals. Vous installez et configurez OpenTelemetry vous-même l' CloudWatch agent et AWS Distro.

Sur les clusters Amazon ECS, Application Signals ne découvre pas automatiquement les noms de vos services ni les clusters dans lesquels ils s’exécutent. Vous devez spécifier ces noms lors de la configuration personnalisée, et les noms que vous spécifiez sont ceux qui sont affichés sur les tableaux de bord d’Application Signals.

⚠ Important

Seul le mode réseau `awsvpc` est pris en charge.

Étape 1 : activer Application Signals dans votre compte

Si vous n’avez pas encore activé Application Signals dans ce compte, vous devez accorder à Application Signals les autorisations nécessaires pour découvrir vos services. Pour ce faire, procédez comme suit. Cette opération ne doit être effectuée qu’une seule fois par compte.

Pour activer Application Signals pour vos applications

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Services.
3. Choisissez Commencer à découvrir vos services.
4. Cochez la case et choisissez Commencer à découvrir les services.

   Si vous effectuez cette étape pour la première fois dans votre compte, le rôle AWSServiceRoleForCloudWatchApplicationSignalslié au service est créé. Ce rôle accorde à Application Signals les autorisations suivantes :

- `xray:GetServiceGraph`
- `logs:StartQuery`
- `logs:GetQueryResults`
- `cloudwatch:GetMetricData`
- `cloudwatch:ListMetrics`
- `tag:GetResources`

Pour plus d'informations sur ce rôle, consultez Autorisations de rôle liées au service pour les signaux d'application CloudWatch .

Étape 2 : créer un rôle IAM

Vous devez créer deux rôles IAM. Si vous avez déjà créé ces rôles, vous devrez peut-être leur ajouter des autorisations.

- Rôle de tâche ECS : les conteneurs utilisent ce rôle pour s'exécuter. Les autorisations doivent correspondre aux besoins de vos applications, plus CloudWatchAgentServerPolicyet AWSXRayWriteOnlyAccess.
- Rôle d'exécution de tâche ECS : Amazon ECS utilise ce rôle pour lancer et exécuter vos conteneurs. Si vous avez déjà créé ce rôle, joignez-y le code AmazonSSMTaskExecutionRolePolicy, ReadOnlyAccess AmazonECS et CloudWatchAgentServerPolicyles politiques.

  Si vous devez stocker des données plus sensibles que Amazon ECS peut utiliser, veuillez consulter Spécification de données sensibles.

Pour plus d'informations sur la création de rôles IAM, consultez Création de rôles IAM.

Étape 3 : Préparation de la configuration de CloudWatch l'agent

Préparez d'abord la configuration de l'agent avec Application Signals activé. Pour ce faire, créez un fichier local nommé `/tmp/ecs-cwagent.json`.

```
{
  "traces": {
    "traces_collected": {
```

```
      "app_signals": {}
    }
  },
  "logs": {
    "metrics_collected": {
      "app_signals": {}
    }
  }
}
```

Chargez ensuite la configuration de l'agent dans le magasin de paramètres SSM. Pour ce faire, entrez la commande suivante : Dans le fichier, remplacez *$REGION* par le nom réel de votre région.

```
aws ssm put-parameter \
--name "ecs-cwagent" \
--type "String" \
--value "`cat /tmp/ecs-cwagent.json`" \
--region "$REGION"
```

Étape 4 : Instrumenter votre application avec l' CloudWatch agent

L'étape suivante consiste à instrumenter votre application pour les signaux CloudWatch d'application.

Java

Pour instrumenter votre application sur Amazon ECS à l'aide de l' CloudWatch agent

1. Spécifiez d'abord un montage lié. Le volume sera utilisé pour partager des fichiers entre conteneurs au cours des prochaines étapes. Vous utiliserez ce montage lié plus tard dans cette procédure.

   ```
   "volumes": [
     {
       "name": "opentelemetry-auto-instrumentation"
     }
   ]
   ```

2. Ajoutez une définition de sidecar d' CloudWatch agent. Pour ce faire, ajoutez un nouveau conteneur appelé `ecs-cwagent` à la définition de tâche de votre application. Remplacez *$REGION* par le nom de votre région. Remplacez-le par le chemin d'accès à la dernière image de CloudWatch conteneur sur Amazon Elastic Container Registry. Pour plus d'informations, consultez le référentiel cloudwatch-agent sur Amazon ECR.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Ajoutez un nouveau conteneur `init` à la définition de tâche de votre application. Remplacez *$IMAGE* par la dernière image du référentiel d'images AWS Distro for OpenTelemetry Amazon ECR.

```
{
  "name": "init",
  "image": "$IMAGE",
  "essential": false,
  "command": [
    "cp",
    "/javaagent.jar",
    "/otel-auto-instrumentation/javaagent.jar"
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
```

```
}
```

4. Ajoutez les variables d’environnement suivantes à votre conteneur d’application. Pour plus d'informations, veuillez consulter la rubrique

Variable d'environnement	Configuration pour activer Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Remplacez `$SVC_NAME` par le nom de votre application. Ce nom sera affiché comme celui de l’application dans les tableaux de bord Application Signals. Remplacez `$HOST_ENV` par l’environnement hôte dans lequel votre application est exécutée. Cet environnement sera affiché comme l’environnement hébergé de votre application dans les tableaux de bord Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Réglez sur `true` pour activer les signaux d'application SpanMetricsProcessor.
`OTEL_METRICS_EXPORTER`	Définissez sur `none` pour désactiver les autres exportateurs de mesures.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Réglez sur `http://127.0.0.1:4315` pour envoyer des métriques au CloudWatch sidecar.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Réglez sur `http://127.0.0.1:4315` pour envoyer des traces au CloudWatch sidecar.
`OTEL_TRACES_SAMPLER`	Définissez X-Ray comme échantillonneur de suivis.
`OTEL_PROPAGATORS`	Ajoutez X-Ray comme l’un des propagateurs.

Variable d'environnement	Configuration pour activer Application Signals
`JAVA_TOOL_OPTIONS`	Injectez l'agent AWS Distro for OpenTelemetry Java.

5. Montez le volume `opentelemetry-auto-instrumentation` que vous avez défini à l'étape 1 de cette procédure.

   Pour une application Java, utilisez ce qui suit.

```
{
  "name": "app",
   ...
  "environment": [
    {
      "name": "OTEL_RESOURCE_ATTRIBUTES",
      "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAME"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
      "value": "true"
    },
    {
      "name": "OTEL_METRICS_EXPORTER",
      "value": "none"
    },
    {
      "name": "JAVA_TOOL_OPTIONS",
      "value": " -javaagent:/otel-auto-instrumentation/javaagent.jar"
    },
    {
      "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
    {
      "name": "OTEL_TRACES_SAMPLER",
      "value": "xray"
    },
    {
      "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
      "value": "http://127.0.0.1:4315"
    },
```

```
    {
      "name": "OTEL_PROPAGATORS",
      "value": "tracecontext,baggage,b3,xray"
    }
  ],
  "mountPoints": [
    {
      "sourceVolume": "opentelemetry-auto-instrumentation",
      "containerPath": "/otel-auto-instrumentation",
      "readOnly": false
    }
  ]
}
```

Python

Avant d'activer les signaux d'application pour vos applications Python, tenez compte des points suivants.

- Dans certaines applications conteneurisées, PYTHONPATH l'absence d'une variable d'environnement peut parfois empêcher l'application de démarrer. Pour résoudre ce problème, veillez à définir la variable d'PYTHONPATHenvironnement à l'emplacement du répertoire de travail de votre application. Cela est dû à un problème connu lié à l' OpenTelemetry auto-instrumentation. Pour plus d'informations sur ce problème, consultez la section Le paramètre d'auto-instrumentation Python de PYTHONPATH n'est pas conforme.
- Pour les applications Django, des configurations supplémentaires sont requises, qui sont décrites dans la documentation OpenTelemetry Python.
  - Utilisez le `--noreload` drapeau pour empêcher le rechargement automatique.
  - Définissez la variable d'DJANGO_SETTINGS_MODULEenvironnement sur l'emplacement du `settings.py` fichier de votre application Django. Cela garantit que OpenTelemetry vous pouvez accéder et intégrer correctement vos paramètres Django.

Pour instrumenter votre application Python sur Amazon ECS avec l' CloudWatch agent

1. Spécifiez d’abord un montage lié. Le volume sera utilisé pour partager des fichiers entre conteneurs au cours des prochaines étapes. Vous utiliserez ce montage lié plus tard dans cette procédure.

```
"volumes": [
  {
    "name": "opentelemetry-auto-instrumentation-python"
  }
]
```

2. Ajoutez une définition de sidecar d' CloudWatch agent. Pour ce faire, ajoutez un nouveau conteneur appelé `ecs-cwagent` à la définition de tâche de votre application. Remplacez *$REGION* par le nom de votre région. Remplacez-le par le chemin d'accès à la dernière image de CloudWatch conteneur sur Amazon Elastic Container Registry. Pour plus d'informations, consultez le référentiel cloudwatch-agent sur Amazon ECR.

```
{
  "name": "ecs-cwagent",
  "image": "$IMAGE",
  "essential": true,
  "secrets": [
    {
      "name": "CW_CONFIG_CONTENT",
      "valueFrom": "ecs-cwagent"
    }
  ],
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/ecs-cwagent",
      "awslogs-region": "$REGION",
      "awslogs-stream-prefix": "ecs"
    }
  }
}
```

3. Ajoutez un nouveau conteneur `init` à la définition de tâche de votre application. Remplacez *$IMAGE* par la dernière image du référentiel d'images AWS Distro for OpenTelemetry Amazon ECR.

```
{
    "name": "init",
    "image": "$IMAGE",
    "essential": false,
```

```
    "command": [
        "cp",
        "-a",
        "/autoinstrumentation/.",
        "/otel-auto-instrumentation-python"
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

4. Ajoutez les variables d'environnement suivantes à votre conteneur d'application. Pour plus d'informations, veuillez consulter la rubrique

Variable d'environnement	Configuration pour activer Application Signals
`OTEL_RESOURCE_ATTRIBUTES`	Remplacez `$SVC_NAME` par le nom de votre application. Ce nom sera affiché comme celui de l'application dans les tableaux de bord Application Signals. Remplacez `$HOST_ENV` par l'environnement hôte dans lequel votre application est exécutée. Cet environnement sera affiché comme l'environnement hébergé de votre application dans les tableaux de bord Application Signals.
`OTEL_AWS_APP_SIGNALS_ENABLED`	Réglez sur `true` pour activer les signaux d'application SpanMetricsProcessor.
`OTEL_METRICS_EXPORTER`	Définissez sur `none` pour désactiver les autres exportateurs de mesures.

Variable d'environnement	Configuration pour activer Application Signals
`OTEL_EXPORTER_OTLP_PROTOCOL`	Définissez ce paramètre `http/protobuf` sur pour envoyer des métriques et des traces CloudWatch via HTTP.
`OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`	Réglez sur `http://127.0.0.1:4316/v1/metrics` pour envoyer des métriques au CloudWatch sidecar.
`OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`	Réglez sur `http://127.0.0.1:4316/v1/traces` pour envoyer des traces au CloudWatch sidecar.
`OTEL_TRACES_SAMPLER`	Définissez X-Ray comme échantillonneur de suivis.
`OTEL_PROPAGATORS`	Ajoutez X-Ray comme l'un des propagateurs.
`OTEL_PYTHON_DISTRO`	Définissez sur `aws_distro` pour utiliser l'instrumentation ADOT Python.
`OTEL_PYTHON_CONFIGURATOR`	Définissez sur `aws_configuration` pour utiliser la configuration ADOT Python.
`PYTHONPATH`	Remplacez `$APP_PATH` par l'emplacement du répertoire de travail de l'application dans le conteneur. Cela est nécessaire pour que l'interpréteur Python puisse trouver les modules de votre application.
`DJANGO_SETTINGS_MODULE`	Nécessaire uniquement pour les applications Django. Réglez-le à l'emplacement du `settings.py` fichier de votre application Django. Remplacez`$PATH_TO_SETTINGS` .

5. Montez le volume `opentelemetry-auto-instrumentation-python` que vous avez défini à l'étape 1 de cette procédure.

   Pour une application Python, utilisez ce qui suit.

```
{
    "name": "app",
    ...
    "environment": [
        {
            "name": "PYTHONPATH",
            "value": "/otel-auto-instrumentation-python/opentelemetry/
instrumentation/auto_instrumentation:$APP_PATH:/otel-auto-instrumentation-
python"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_PROTOCOL",
            "value": "http/protobuf"
        },
        {
            "name": "OTEL_TRACES_SAMPLER",
            "value": "xray"
        },
        {
            "name": "OTEL_TRACES_SAMPLER_ARG",
            "value": "endpoint=http://localhost:2000"
        },
        {
            "name": "OTEL_LOGS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_PYTHON_DISTRO",
            "value": "aws_distro"
        },
        {
            "name": "OTEL_PYTHON_CONFIGURATOR",
            "value": "aws_configurator"
        },
        {
            "name": "OTEL_EXPORTER_OTLP_TRACES_ENDPOINT",
            "value": "http://localhost:4316/v1/traces"
        },
```

```
        {
            "name": "OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT",
            "value": "http://localhost:4316/v1/metrics"
        },
        {
            "name": "OTEL_METRICS_EXPORTER",
            "value": "none"
        },
        {
            "name": "OTEL_AWS_APP_SIGNALS_ENABLED",
            "value": "true"
        },
        {
            "name": "OTEL_RESOURCE_ATTRIBUTES",
            "value": "aws.hostedin.environment=$HOST_ENV,service.name=$SVC_NAM"
        },
        {
            "name": "DJANGO_SETTINGS_MODULE",
            "value": "$PATH_TO_SETTINGS.settings"
        }
    ],
    "mountPoints": [
        {
            "sourceVolume": "opentelemetry-auto-instrumentation-python",
            "containerPath": "/otel-auto-instrumentation-python",
            "readOnly": false
        }
    ]
}
```

Étape 5 : déployer votre application

Créez une nouvelle révision de votre définition de tâche et déployez-la dans votre cluster d'applications. Vous devriez voir trois conteneurs dans la tâche nouvellement créée :

- `init`
- `ecs-cwagent`
- `app`

Utilisation d’une configuration personnalisée pour activer Application Signals sur Amazon EC2 et d’autres plateformes

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Pour les applications exécutées sur Amazon EC2 et d'autres architectures autres qu'Amazon EKS, vous devez installer et configurer vous-même l' CloudWatch agent et AWS Distro. OpenTelemetry Sur ces architectures activées avec une configuration personnalisée d’Application Signals, Application Signals ne découvre pas automatiquement les noms de vos services ni les hôtes ou clusters sur lesquels ils s’exécutent. Vous devez spécifier ces noms lors de la configuration personnalisée, et les noms que vous spécifiez sont ceux qui sont affichés sur les tableaux de bord d’Application Signals.

Les étapes suivantes ont été testées sur des instances Amazon EC2, mais elles devraient également fonctionner sur d'autres architectures compatibles avec AWS Distro for. OpenTelemetry

Prérequis

- Pour obtenir de l'aide pour Application Signals, vous devez utiliser la version la plus récente de l' CloudWatchagent et de la AWS distribution pour l' OpenTelemetry agent.
- Vous devez l'avoir AWS CLI installé sur l'instance. Nous recommandons AWS CLI la version 2, mais la version 1 devrait également fonctionner. Pour plus d'informations sur l'installation du AWS CLI, voir Installer ou mettre à jour la dernière version du AWS CLI.

⚠ Important

Si vous utilisez OpenTelemetry déjà une application que vous avez l'intention d'activer pour les signaux d'application, consultez la section OpenTelemetry considérations relatives à la compatibilité avant d'activer les signaux d'application.

Étape 1 : activer Application Signals dans votre compte

Si vous n’avez pas encore activé Application Signals dans ce compte, vous devez accorder à Application Signals les autorisations nécessaires pour découvrir vos services. Pour ce faire, procédez comme suit. Cette opération ne doit être effectuée qu’une seule fois par compte.

Pour activer Application Signals pour vos applications

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Services.
3. Choisissez Commencer à découvrir vos services.
4. Cochez la case et choisissez Commencer à découvrir les services.

   Si vous effectuez cette étape pour la première fois dans votre compte, le rôle AWSServiceRoleForCloudWatchApplicationSignalslié au service est créé. Ce rôle accorde à Application Signals les autorisations suivantes :

   - `xray:GetServiceGraph`
   - `logs:StartQuery`
   - `logs:GetQueryResults`
   - `cloudwatch:GetMetricData`
   - `cloudwatch:ListMetrics`
   - `tag:GetResources`

   Pour plus d’informations sur ce rôle, consultez Autorisations de rôle liées au service pour les signaux d'application CloudWatch .

Étape 2 : télécharger et démarrer l' CloudWatch agent

Pour installer l' CloudWatch agent dans le cadre de l'activation des signaux d'application sur une instance Amazon EC2

1. Téléchargez la dernière version de l' CloudWatch agent sur l'instance. Si l' CloudWatch agent est déjà installé sur l'instance, vous devrez peut-être le mettre à jour. Seules les versions de l'agent publiées le 30 novembre 2023 ou une version ultérieure prennent en charge les signaux CloudWatch d'application.

Pour plus d'informations sur le téléchargement de CloudWatch l'agent, consultezTéléchargez le package de CloudWatch l'agent.

2. Avant de démarrer l' CloudWatch agent, configurez-le pour activer les signaux d'application. L'exemple suivant est une configuration d' CloudWatch agent qui active les signaux d'application pour les métriques et les traces sur un hôte EC2.

   Vous pouvez créer ce fichier en saisissant la commande suivante :

   ```
   vim amazon-cloudwatch-agent.json
   ```

   Ajoutez ce qui suit au contenu de ce fichier.

   ```
   {
     "traces": {
       "traces_collected": {
         "app_signals": {}
       }
     },
     "logs": {
       "metrics_collected": {
         "app_signals": {}
       }
     }
   }
   ```

3. Associez les politiques CloudWatchAgentServerPolicyet AWSXrayWriteOnlyAccessIAM au rôle IAM de votre instance Amazon EC2.

   a. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.

   b. Choisissez Rôles et recherchez le rôle utilisé par votre instance Amazon EC2. Choisissez ensuite le nom de ce rôle.

   c. Sous l’onglet Autorisations, choisissez Ajouter des autorisations, Attacher des politiques.

   d. Trouvez CloudWatchAgentServerPolicy. Utilisez le champ de recherche si nécessaire. Cochez la case correspondant à la politique, puis choisissez Ajouter des autorisations.

   e. Trouvez AWSXrayWriteOnlyAccess. Utilisez le champ de recherche si nécessaire. Cochez la case correspondant à la politique, puis choisissez Ajouter des autorisations.

4. Démarrez l' CloudWatch agent en saisissant les commandes suivantes. Remplacez *agent-config-file-path*par le chemin d'accès au fichier de configuration de l' CloudWatch agent, tel que `./amazon-cloudwatch-agent.json`. Vous devez inclure le préfixe `file:` comme indiqué.

```
export CONFIG_FILE_PATH=./amazon-cloudwatch-agent.json
```

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \
-a fetch-config \
-m ec2 -s -c file:$CONFIG_FILE_PATH
```

Étape 3 : instrumenter votre application et la démarrer

L'étape suivante consiste à instrumenter votre application pour les signaux CloudWatch d'application.

Java

Pour instrumenter vos applications Java dans le cadre de l'activation des signaux d'application sur une instance Amazon EC2

1. Téléchargez la dernière version de l'agent d'auto-instrumentation de AWS Distro pour OpenTelemetry Java. Vous pouvez télécharger la version la plus récente en utilisant ce lien. Vous pouvez consulter les informations relatives à toutes les versions publiées dans la section aws-otel-java-instrumentation Releases.
2. Pour optimiser vos avantages d'Application Signals, utilisez des variables d'environnement pour fournir des informations supplémentaires avant de démarrer votre application. Ces informations seront affichées dans les tableaux de bord d'Application Signals.
   a. Pour la variable `OTEL_RESOURCE_ATTRIBUTES`, spécifiez les informations suivantes sous forme de paires clé-valeur :
      - `aws.hostedin.environment` définit l'environnement dans lequel l'application s'exécute. Cet environnement sera affiché comme l'environnement hébergé de votre application dans les tableaux de bord Application Signals. Cette clé d'attribut est utilisée uniquement par Application Signals et est convertie en annotations de trace X-Ray et en dimensions CloudWatch métriques. Si vous ne fournissez pas de valeur pour cette clé, la valeur par défaut `Generic` est utilisée.

- `service.name` défini le nom du service. Il sera affiché comme nom de service pour votre application dans les tableaux de bord d'Application Signals. Si vous ne fournissez pas de valeur pour cette clé, la valeur par défaut `unknown_service` est utilisée.

b. Pour la variable `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, spécifiez l'URL du point de terminaison de base vers lequel les suivis doivent être exportés. L' CloudWatch agent expose 4315 comme port OLTP. Sur Amazon EC2, étant donné que les applications communiquent avec l' CloudWatch agent local, vous devez définir cette valeur sur `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315`

c. Pour la variable `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, spécifiez l'URL du point de terminaison de base vers lequel les métriques doivent être exportées. L' CloudWatch agent expose 4315 comme port OLTP. Sur Amazon EC2, étant donné que les applications communiquent avec l' CloudWatch agent local, vous devez définir cette valeur sur `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315`

d. Pour la `JAVA_TOOL_OPTIONS` variable, spécifiez le chemin où l'agent d'auto-instrumentation de AWS Distro for OpenTelemetry Java est stocké.

```
export JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH'
```

Par exemple :

```
export ADOT_AGENT_PATH=./aws-opentelemetry-agent.jar
```

e. Pour la variable `OTEL_METRICS_EXPORTER`, nous vous recommandons de définir la valeur sur `none`. Cela désactive les autres exportateurs de métriques afin que seul l'exportateur d'Application Signals soit utilisé.

f. Pour la `OTEL_AWS_APP_SIGNALS_ENABLED` variable, activez le SpanMetricProcessor (SMP) en réglant `OTEL_AWS_APP_SIGNALS_ENABLED` sur`true`. Cela génère des métriques d'Application Signals à partir des suivis.

3. Démarrez votre application avec les variables d'environnement décrites à l'étape précédente. Voici un exemple de script de démarrage.

```
JAVA_TOOL_OPTIONS=' -javaagent:$ADOT_AGENT_PATH' \
OTEL_METRICS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
```

```
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4315 \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4315 \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$YOUR_HOST_ENV,service.name=
$YOUR_SVC_NAME \
java -jar $MY_JAVA_APP.jar
```

Python

Pour instrumenter vos applications Python dans le cadre de l'activation des signaux d'application sur une instance Amazon EC2

1. Téléchargez la dernière version de l'agent d'auto-instrumentation AWS Distro for OpenTelemetry Python. Pour l'installer, exécutez la commande d' ci-dessous.

   ```
   pip install aws-opentelemetry-distro
   ```

   Vous pouvez consulter des informations sur toutes les versions publiées sur AWS Distro pour l'instrumentation OpenTelemetry Python.
2. Pour optimiser vos avantages d’Application Signals, utilisez des variables d’environnement pour fournir des informations supplémentaires avant de démarrer votre application. Ces informations seront affichées dans les tableaux de bord d’Application Signals.

   a. Pour la variable `OTEL_RESOURCE_ATTRIBUTES`, spécifiez les informations suivantes sous forme de paires clé-valeur :

      - `aws.hostedin.environment` définit l’environnement dans lequel l’application s’exécute. Cet environnement sera affiché comme l’environnement hébergé de votre application dans les tableaux de bord Application Signals. Cette clé d'attribut est utilisée uniquement par Application Signals et est convertie en annotations de trace X-Ray et en dimensions CloudWatch métriques. Si vous ne fournissez pas de valeur pour cette clé, la valeur par défaut `Generic` est utilisée.
      - `service.name` défini le nom du service. Il sera affiché comme nom de service pour votre application dans les tableaux de bord d’Application Signals. Si vous ne fournissez pas de valeur pour cette clé, la valeur par défaut `unknown_service` est utilisée.

b. Pour la `OTEL_EXPORTER_OTLP_PROTOCOL` variable, spécifiez d'`http/protobuf`exporter les données de télémétrie via HTTP vers les points de terminaison de l' CloudWatch agent répertoriés dans les étapes suivantes.

c. Pour la variable `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT`, spécifiez l'URL du point de terminaison de base vers lequel les suivis doivent être exportés. L' CloudWatch agent expose 4316 comme port OLTP sur HTTP. Sur Amazon EC2, étant donné que les applications communiquent avec l' CloudWatch agent local, vous devez définir cette valeur sur `OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces`

d. Pour la variable `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT`, spécifiez l'URL du point de terminaison de base vers lequel les métriques doivent être exportées. L' CloudWatch agent expose 4316 comme port OLTP sur HTTP. Sur Amazon EC2, étant donné que les applications communiquent avec l' CloudWatch agent local, vous devez définir cette valeur sur `OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics`

e. Pour la variable `OTEL_METRICS_EXPORTER`, nous vous recommandons de définir la valeur sur `none`. Cela désactive les autres exportateurs de métriques afin que seul l'exportateur d'Application Signals soit utilisé.

f. Pour la `OTEL_AWS_APP_SIGNALS_ENABLED` variable, activez le paramètre SpanMetricProcessor by `OTEL_AWS_APP_SIGNALS_ENABLED` sur`true`. Cela génère des métriques d'Application Signals à partir des suivis.

3. Démarrez votre application avec les variables d'environnement décrites à l'étape précédente. Voici un exemple de script de démarrage.

   - Remplacez `$HOST_ENV` par l'environnement hôte dans lequel votre application est exécutée. Il sera affiché sous forme d'environnement hébergé pour votre application, dans les tableaux de bord des signaux d'application.
   - Remplacez `$SVC_NAME` par le nom de votre application. Il sera affiché sous forme de nom de l'application, dans les tableaux de bord des signaux d'application.
   - Remplacez `$PYTHON_APP` par l'emplacement et le nom de votre application.

```
OTEL_METRICS_EXPORTER=none \
OTEL_LOGS_EXPORTER=none \
OTEL_AWS_APP_SIGNALS_ENABLED=true \
OTEL_PYTHON_DISTRO=aws_distro \
```

```
OTEL_PYTHON_CONFIGURATOR=aws_configurator \
OTEL_EXPORTER_OTLP_PROTOCOL=http/protobuf \
OTEL_TRACES_SAMPLER=xray \
OTEL_TRACES_SAMPLER_ARG="endpoint=http://localhost:2000" \
OTEL_AWS_APP_SIGNALS_EXPORTER_ENDPOINT=http://localhost:4316/v1/metrics \
OTEL_EXPORTER_OTLP_TRACES_ENDPOINT=http://localhost:4316/v1/traces \
OTEL_RESOURCE_ATTRIBUTES=aws.hostedin.environment=$HOST_ENV,service.name=
$SVC_NAME \
opentelemetry-instrument python $PYTHON_APP.py
```

Avant d'activer les signaux d'application pour vos applications Python, tenez compte des points suivants.

- Dans certaines applications conteneurisées, PYTHONPATH l'absence d'une variable d'environnement peut parfois empêcher l'application de démarrer. Pour résoudre ce problème, veillez à définir la variable d'PYTHONPATHenvironnement à l'emplacement du répertoire de travail de votre application. Cela est dû à un problème connu lié à l' OpenTelemetry auto-instrumentation. Pour plus d'informations sur ce problème, consultez la section Le paramètre d'auto-instrumentation Python de PYTHONPATH n'est pas conforme.
- Pour les applications Django, des configurations supplémentaires sont requises, qui sont décrites dans la documentation OpenTelemetry Python.
  - Utilisez le `--noreload` drapeau pour empêcher le rechargement automatique.
  - Définissez la variable d'DJANGO_SETTINGS_MODULEenvironnement sur l'emplacement du `settings.py` fichier de votre application Django. Cela garantit que OpenTelemetry vous pouvez accéder et intégrer correctement vos paramètres Django.

Résolution des problèmes liés à l’installation d’Application Signals

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Cette section contient des conseils de résolution des problèmes relatifs aux signaux CloudWatch d'application.

Rubriques

L’application ne démarre pas après l’activation d’Application Signals

Si votre application sur un cluster Amazon EKS ne démarre pas une fois que vous avez activé Application Signals sur le cluster, vérifiez les points suivants :

- Vérifiez si l’application a été instrumentée par une autre solution de surveillance. Application Signals ne prend pas en charge la coexistence avec d’autres solutions d’instrumentation.
- Vérifiez que votre application répond aux exigences de compatibilité pour utiliser Application Signals. Pour plus d’informations, veuillez consulter la rubrique Systèmes compatibles avec Application Signals .
- Si votre application ne parvient pas à extraire les artefacts Application Signals tels que l'agent et CloudWatch les images de l'agent AWS Distro for OpenTelemetery Java ou Python, cela peut être dû à un problème de réseau.

Pour atténuer le problème, supprimez l'annotation `instrumentation.opentelemetry.io/inject-java: "true"` ou `instrumentation.opentelemetry.io/inject-python: "true"` du manifeste de déploiement de votre application, puis redéployez votre application. Vérifiez ensuite si l’application fonctionne.

L'application Python ne démarre pas une fois les signaux d'application activés

Le fait qu'une variable d'`PYTHONPATH`environnement manquante puisse parfois empêcher le démarrage de l'application est un problème connu en matière d' OpenTelemetry instrumentation automatique. Pour résoudre ce problème, assurez-vous de définir la variable d'`PYTHONPATH`environnement à l'emplacement du répertoire de travail de votre application. Pour plus d'informations sur ce problème, consultez la section Le paramètre d'auto-instrumentation Python de PYTHONPATH n'est pas conforme au comportement de résolution des modules de Python, ce qui perturbe les applications Django.

Pour les applications Django, des configurations supplémentaires sont requises, qui sont décrites dans la documentation OpenTelemetry Python.

- Utilisez le `--noreload` drapeau pour empêcher le rechargement automatique.
- Définissez la variable d'`DJANGO_SETTINGS_MODULE`environnement sur l'emplacement du `settings.py` fichier de votre application Django. Cela garantit que OpenTelemetry vous pouvez accéder et intégrer correctement vos paramètres Django.

Les données de télémétrie sont manquantes dans CloudWatch et X-Ray

Si des métriques ou des suivis sont absents des tableaux de bord d'Application Signals, cela peut être dû aux causes suivantes. N'étudiez ces causes que si vous avez attendu 15 minutes pour qu'Application Signals collecte et affiche les données depuis votre dernière mise à jour.

- Assurez-vous que la bibliothèque et le framework que vous utilisez sont pris en charge par l'agent Java ADOT. Pour plus d'informations, veuillez consulter la rubrique Bibliothèques/Frameworks.
- Assurez-vous que l' CloudWatch agent est en cours d'exécution. Vérifiez d'abord l'état des modules d' CloudWatch agents et assurez-vous qu'ils le `Running` sont tous.

  ```
  kubectl -n amazon-cloudwatch get pods.
  ```

  Ajoutez ce qui suit au fichier de configuration de l' CloudWatch agent pour activer les journaux de débogage, puis redémarrez l'agent.

  ```
  "agent": {
  >>>>>>> streams
    "region": "${REGION}",
    "debug": true
  },
  ```

  Vérifiez ensuite l'absence d'erreurs dans les modules CloudWatch d'agent.
- Vérifiez l'absence de problèmes de configuration avec l' CloudWatch agent. Vérifiez que les informations suivantes se trouvent toujours dans le CloudWatch fichier de configuration de l'agent et que l'agent a été redémarré depuis son ajout.

  ```
  "agent": {
    "region": "${REGION}",
    "debug": true
  ```

```
},
```

Vérifiez ensuite les journaux de OpenTelemetry débogage pour détecter les messages d'erreur tels que`ERROR io.opentelemetry.exporter.internal.grpc.OkHttpGrpcExporter - Failed to export ...`. Ces messages peuvent indiquer le problème.

Si cela ne résout pas le problème, videz et vérifiez les variables d'environnement dont le nom commence par `OTEL_` en décrivant le pod à l'aide de la commande `kubectl describe pod`.

- Pour activer la journalisation du débogage en OpenTelemetry Python, définissez la variable d'environnement `OTEL_PYTHON_LOG_LEVEL` sur `debug` et redéployez l'application.
- Vérifiez que les autorisations d'exportation des données depuis l' CloudWatchagent ne sont pas correctes ou insuffisantes. Si vous voyez `Access Denied` des messages dans les journaux des CloudWatch agents, cela peut être à l'origine du problème. Il est possible que les autorisations appliquées lors de l'installation de l' CloudWatch agent aient été modifiées ou révoquées ultérieurement.
- Vérifiez l'absence d'un problème de AWS distribution pour OpenTelemetry (ADOT) lors de la génération de données de télémétrie.

  Assurez-vous que les annotations d'instrumentation `instrumentation.opentelemetry.io/inject-java` et `sidecar.opentelemetry.io/inject-java` sont appliquées au déploiement de l'application et que la valeur est `true`. Sans celles-ci, les pods d'application ne seront pas instrumentés même si le module complémentaire ADOT est correctement installé.

  Ensuite, vérifiez si le conteneur `Init` est appliqué sur l'application et si son état `Ready` est `True`. Si le conteneur `init` n'est pas prêt, veuillez consulter l'état pour en connaître la raison.

  Si le problème persiste, procédez comme suit pour activer la journalisation du débogage sur le SDK OpenTelemetry Java. Recherchez ensuite les messages commençant par `ERROR io.telemetry`.

  Pour activer la journalisation du débogage, définissez la variable d'environnement `OTEL_JAVAAGENT_DEBUG` sur true et redéployez l'application.
- L'exportateur de métrique/d'intervalle est peut-être en train de supprimer des données. Pour le savoir, consultez le journal des applications pour voir s'il contient des messages incluant `Failed to export...`.

- L' CloudWatch agent est peut-être limité lorsqu'il envoie des métriques ou des spans à Application Signals. Vérifiez la présence de messages indiquant une limitation dans les journaux de l' CloudWatch agent.

Les métriques de dépendance ont des valeurs inconnues

Si vous voyez UnknownOperationUnknownRemoteService, ou UnknownRemoteOperationpour un nom de dépendance ou une opération dans les tableaux de bord des signaux d'application, vérifiez si l'occurrence de points de données pour le service distant inconnu et le fonctionnement à distance inconnu coïncident avec leurs déploiements. Il s'agit d'un problème connu sur Application Signals et il est prévu de le corriger dans une future version.

Gestion d'un ConfigurationConflict lors de la gestion du module complémentaire Amazon CloudWatch Observability EKS

Lorsque vous installez ou mettez à jour le module complémentaire Amazon CloudWatch Observability EKS, si vous remarquez une défaillance causée par un type `Health Issue ConfigurationConflict` de fichier dont la description commence par`Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, c'est probablement parce que vous avez déjà ClusterRoleBinding installé l' CloudWatch agent et ses composants associés tels que le ServiceAccount, le ClusterRole et le sur le cluster. Lorsque le module complémentaire tente d'installer l' CloudWatch agent et ses composants associés, s'il détecte une modification du contenu, il échoue par défaut à l'installation ou à la mise à jour pour éviter de modifier l'état des ressources du cluster.

Si vous essayez d'intégrer le module complémentaire Amazon CloudWatch Observability EKS et que vous constatez cet échec, nous vous recommandons de supprimer une configuration d' CloudWatch agent existante que vous aviez précédemment installée sur le cluster, puis d'installer le module complémentaire EKS. Veillez à sauvegarder toutes les personnalisations que vous avez éventuellement apportées à la configuration d'origine de l' CloudWatch agent, telle qu'une configuration d'agent personnalisée, et à les fournir au module complémentaire Amazon CloudWatch Observability EKS lors de sa prochaine installation ou mise à jour. Si vous avez déjà installé l' CloudWatch agent pour l'intégration à Container Insights, consultez Suppression de l' CloudWatch agent et de Fluent Bit for Container Insights pour plus d'informations.

Le module complémentaire prend également en charge une option de configuration de résolution des conflits capable de spécifier `OVERWRITE`. Vous pouvez utiliser cette option pour procéder à l'installation ou à la mise à jour du module complémentaire en remplaçant les conflits sur le cluster.

Si vous utilisez la console Amazon EKS, vous trouverez la Méthode de résolution des conflits lorsque vous choisissez les Paramètres de configuration facultatifs lorsque vous créez ou mettez à jour le module complémentaire. Si vous utilisez le AWS CLI, vous pouvez fournir le `--resolve-conflicts OVERWRITE` à votre commande pour créer ou mettre à jour le module complémentaire.

Configuration d'Application Signals

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Cette section contient des informations sur la configuration des signaux CloudWatch d'application.

Taux d'échantillonnage du suivi

Par défaut, lorsque vous activez Application Signals, l'échantillonnage centralisé X-Ray est activé en utilisant les paramètres de fréquence d'échantillonnage par défaut de `reservoir=1/s` et `fixed_rate=5%`. Les variables d'environnement de l'agent SDK AWS Distro for OpenTelemetry (ADOT) sont définies comme suit.

Variable d'environnement	Valeur	Remarque
`OTEL_TRACES_SAMPLER`	`xray`	
`OTEL_TRACES_SAMPLER_ARG`	`endpoint=http://cloudwatch-agent.amazon-cloudwatch:2000`	Point de terminaison de l' CloudWatch agent

Pour plus d'informations sur la configuration d'échantillonnage, veuillez consulter les ressources suivantes :

- Pour modifier l'échantillonnage par X-Ray, veuillez consulter la rubrique Customizing sampling rules.
- Pour modifier l'échantillonnage ADOT, voir Configuration du OpenTelemetry collecteur pour le prélèvement à distance par rayons X

Si vous souhaitez désactiver l'échantillonnage centralisé X-Ray et utiliser l'échantillonnage local à la place, définissez les valeurs suivantes pour l'agent Java du SDK ADOT comme indiqué ci-dessous. L'exemple suivant définit le taux d'échantillonnage à 5 %.

Variable d'environnement	Valeur
`OTEL_TRACES_SAMPLER`	`parentbased_traceidratio`
`OTEL_TRACES_SAMPLER_ARG`	`0.05`

Pour plus d'informations sur les paramètres d'échantillonnage plus avancés, veuillez consulter OTEL_TRACES_SAMPLER.

Gérez les opérations à haute cardinalité

Application Signals inclut des paramètres dans l' CloudWatch agent que vous pouvez utiliser pour gérer la cardinalité de vos opérations et gérer l'exportation des métriques afin d'optimiser les coûts. Par défaut, la fonction de limitation des métriques devient active lorsque le nombre d'opérations distinctes pour un service au fil du temps dépasse le seuil par défaut de 500. Vous pouvez ajuster le comportement en ajustant les paramètres de configuration.

Déterminez si la limitation métrique est activée

Vous pouvez utiliser les méthodes suivantes pour déterminer si la limite métrique par défaut est respectée. Si tel est le cas, vous devez envisager d'optimiser le contrôle de cardinalité en suivant les étapes de la section suivante.

- Dans la CloudWatch console, choisissez Application Signals, Services. Si vous voyez une opération nommée AllOtherOperationsou RemoteOperationnommée AllOtherRemoteOperations, cela signifie que la limitation des métriques est en cours.
- Si des métriques collectées par Application Signals ont la valeur correspondant `AllOtherOperations` à leur `Operation` dimension, cela signifie que la limitation des métriques se produit.
- Si des métriques collectées par Application Signals ont la valeur correspondant `AllOtherRemoteOperations` à leur `RemoteOperation` dimension, cela signifie que la limitation des métriques se produit.

Optimisez le contrôle de la cardinalité

Pour optimiser votre contrôle de cardinalité, vous pouvez effectuer les opérations suivantes :

- Créez des règles personnalisées pour agréger les opérations.
- Configurez votre politique de limitation des métriques.

Créez des règles personnalisées pour agréger les opérations

Les opérations à cardinalité élevée peuvent parfois être causées par des valeurs uniques inappropriées extraites du contexte. Par exemple, l'envoi de requêtes HTTP/S qui incluent des identifiants utilisateur ou des identifiants de session dans le chemin peut entraîner des centaines d'opérations disparates. Pour résoudre ces problèmes, nous vous recommandons de configurer l' CloudWatch agent avec des règles de personnalisation afin de réécrire ces opérations.

Dans les cas où il y a une augmentation de la génération de nombreux indicateurs différents par le biais d'`RemoteOperation`appels individuels `PUT /api/customer/owners/123``PUT /api/customer/owners/456`, tels que, et de demandes similaires, nous vous recommandons de regrouper ces opérations en une seule`RemoteOperation`. L'une des approches consiste à normaliser tous les `RemoteOperation` appels commençant `PUT /api/customer/owners/` par un format uniforme, en particulier`PUT /api/customer/owners/{ownerId}`. L'exemple suivant illustre ce scénario. Pour plus d'informations sur les autres règles de personnalisation, consultezActiver les signaux CloudWatch d'application.

```
{
   "logs":{
      "metrics_collected":{
         "app_signals":{
            "rules":[
               {
                  "selectors":[
                     {
                        "dimension":"RemoteOperation",
                        "match":"PUT /api/customer/owners/*"
                     }
                  ],
                  "replacements":[
                     {
                        "target_dimension":"RemoteOperation",
                        "value":"PUT /api/customer/owners/{ownerId}"
```

```
                    }
                ],
                "action":"replace"
            }
        ]
      }
    }
  }
}
```

Dans d'autres cas, les métriques à haute cardinalité peuvent avoir été agrégées`AllOtherRemoteOperations`, et il est possible que les métriques spécifiques incluses ne soient pas claires. L' CloudWatch agent est en mesure de consigner les opérations abandonnées. Pour identifier les opérations abandonnées, utilisez la configuration de l'exemple suivant pour activer la journalisation jusqu'à ce que le problème réapparaisse. Inspectez ensuite les journaux de l' CloudWatch agent (accessibles par conteneur `stdout` ou par fichier journal EC2) et recherchez le mot clé`drop metric data`.

```
{
  "agent": {
    "config": {
      "agent": {
        "debug": true
      },
      "traces": {
        "traces_collected": {
          "app_signals": {
          }
        }
      },
      "logs": {
        "metrics_collected": {
          "app_signals": {
            "limiter": {
              "log_dropped_metrics": true
            }
          }
        }
      }
    }
  }
}
```

Créez votre politique de limitation des métriques

Si la configuration de limitation métrique par défaut ne tient pas compte de la cardinalité de votre service, vous pouvez personnaliser la configuration du limiteur métrique. Pour ce faire, ajoutez une `limiter` section sous la `logs/metrics_collected/app_signals` section du fichier de configuration de l' CloudWatch agent.

L'exemple suivant abaisse le seuil de limitation des métriques de 500 métriques distinctes à 100.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "limiter": {
          "drop_threshold": 100
        }
      }
    }
  }
}
```

Objectifs de niveau de service (SLO)

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Vous pouvez utiliser Application Signals pour créer des objectifs de niveau de service pour les services destinés à vos opérations métier critiques. En créant des SLO sur ces services, vous pourrez les suivre sur le tableau de bord SLO, ce qui vous donnera une at-a-glance vue d'ensemble de vos opérations les plus importantes.

En plus de créer un aperçu rapide que vos opérateurs peuvent utiliser pour connaître l'état actuel des opérations critiques, vous pouvez utiliser les SLO pour suivre les performances à long terme de vos services, afin de vous assurer qu'ils répondent à vos attentes. Si vous avez conclu des accords de niveau de service avec des clients, les SLO sont un excellent outil pour garantir leur respect.

L’évaluation de l’état de santé de vos services à l’aide des SLO commence par la définition d’objectifs clairs et mesurables basés sur des indicateurs de performance clés, à savoir des indicateurs de niveau de service (SLI). Un SLO suit les performances du SLI par rapport au seuil et à l’objectif que vous avez définis, et indique dans quelle mesure les performances de votre application se situent par rapport au seuil.

Application Signals vous aide à définir des SLO sur vos indicateurs de performance clés. Application Signals collecte les métriques `Latency` et `Availability` automatiquement pour chaque service et chaque opération qu’elle découvre, et ces métriques sont souvent idéales pour être utilisées en tant que SLI. Avec l’assistant de création de SLO, vous pouvez utiliser ces métriques pour vos SLO. Vous pouvez ensuite suivre l’état de tous vos SLO à l’aide des tableaux de bord d’Application Signals.

Vous pouvez définir des SLO pour des opérations spécifiques que votre service appelle ou utilise. Vous pouvez utiliser n'importe quelle CloudWatch métrique ou expression métrique comme SLI, en plus d'utiliser les `Availability` métriques `Latency` et.

La création de SLO est très importante pour tirer le meilleur parti des signaux d' CloudWatchapplication. Une fois que vous avez créé des SLO, vous pouvez consulter leur état dans la console Application Signals pour voir rapidement lesquels de vos services et opérations critiques fonctionnent bien et lesquels ne le sont pas. Le fait d’avoir des SLO à suivre offre les principaux avantages suivants :

- Il est plus facile pour vos opérateurs de services de voir l’état de fonctionnement actuel des services critiques par rapport au SLI. Ils peuvent ensuite rapidement trier et identifier les services et les opérations non saines.
- Vous pouvez suivre les performances de vos services par rapport à des objectifs métier mesurables sur de longues périodes.

En choisissant les paramètres sur lesquels définir les SLO, vous priorisez ce qui est important pour vous. Les tableaux de bord d’Application Signals présentent automatiquement des informations sur ce que vous avez priorisé.

Lorsque vous créez un SLO, vous pouvez également choisir de créer des CloudWatch alarmes en même temps pour surveiller les SLO. Vous pouvez définir des alarmes qui surveillent les dépassements du seuil, ainsi que les niveaux d’alerte. Ces alarmes peuvent vous avertir automatiquement si les métriques SLO dépassent le seuil que vous avez défini ou s’approchent d’un seuil d’avertissement. Par exemple, un SLO proche de son seuil d’alerte peut vous indiquer que votre

équipe devra peut-être ralentir le taux de désabonnement de l'application pour s'assurer que les objectifs de performance à long terme sont atteints.

Rubriques

Concepts SLO

Un SLO comprend les composants suivants :

- Un indicateur de niveau de service (SLI), qui est une métrique de performance clé que vous spécifiez. Il représente le niveau de performance souhaité pour votre application. Application Signals collecte les métriques clés `Latency` et `Availability` automatiquement pour les services et opérations qu'elle découvre, et ces métriques sont souvent idéales pour être utilisées en tant que SLO.

  Vous choisissez le seuil à utiliser pour votre SLI. Par exemple, 200 ms pour la latence.
- Un objectif ou un objectif de réalisation, qui est le pourcentage de temps pendant lequel le SLI devrait atteindre le seuil sur chaque intervalle de temps. Les intervalles de temps peuvent être de quelques heures ou d'une année.

  Les intervalles peuvent être des intervalles calendaires ou des intervalles glissants.
  - Les intervalles du calendrier sont alignés sur le calendrier, par exemple pour un SLO suivi par mois. CloudWatch ajuste automatiquement les chiffres de santé, de budget et de réussite en fonction du nombre de jours par mois. Les intervalles calendaires sont mieux adaptés aux objectifs métier mesurés sur une base alignée sur le calendrier.
  - Les intervalles glissants sont calculés sur une base continue. Les intervalles glissants sont mieux adaptés au suivi de l'expérience utilisateur récente de votre application.
- La période est une unité de temps plus courte, et plusieurs périodes constituent un intervalle. Les performances de l'application sont comparées au SLI pendant chaque période comprise dans l'intervalle. Pour chaque période, il est déterminé que l'application a atteint ou non les performances nécessaires.

Par exemple, un objectif de 99 % avec un intervalle calendaire d'un jour et une période d'une minute signifie que l'application doit atteindre ou atteindre le seuil de réussite pendant 99 % des périodes d'une minute de la journée. Si c'est le cas, le SLO est atteint pour ce jour-là. Le jour suivant correspond à un nouvel intervalle d'évaluation, et l'application doit atteindre ou atteindre le seuil de réussite pendant 99 % des périodes d'une minute du deuxième jour pour atteindre le SLO du deuxième jour.

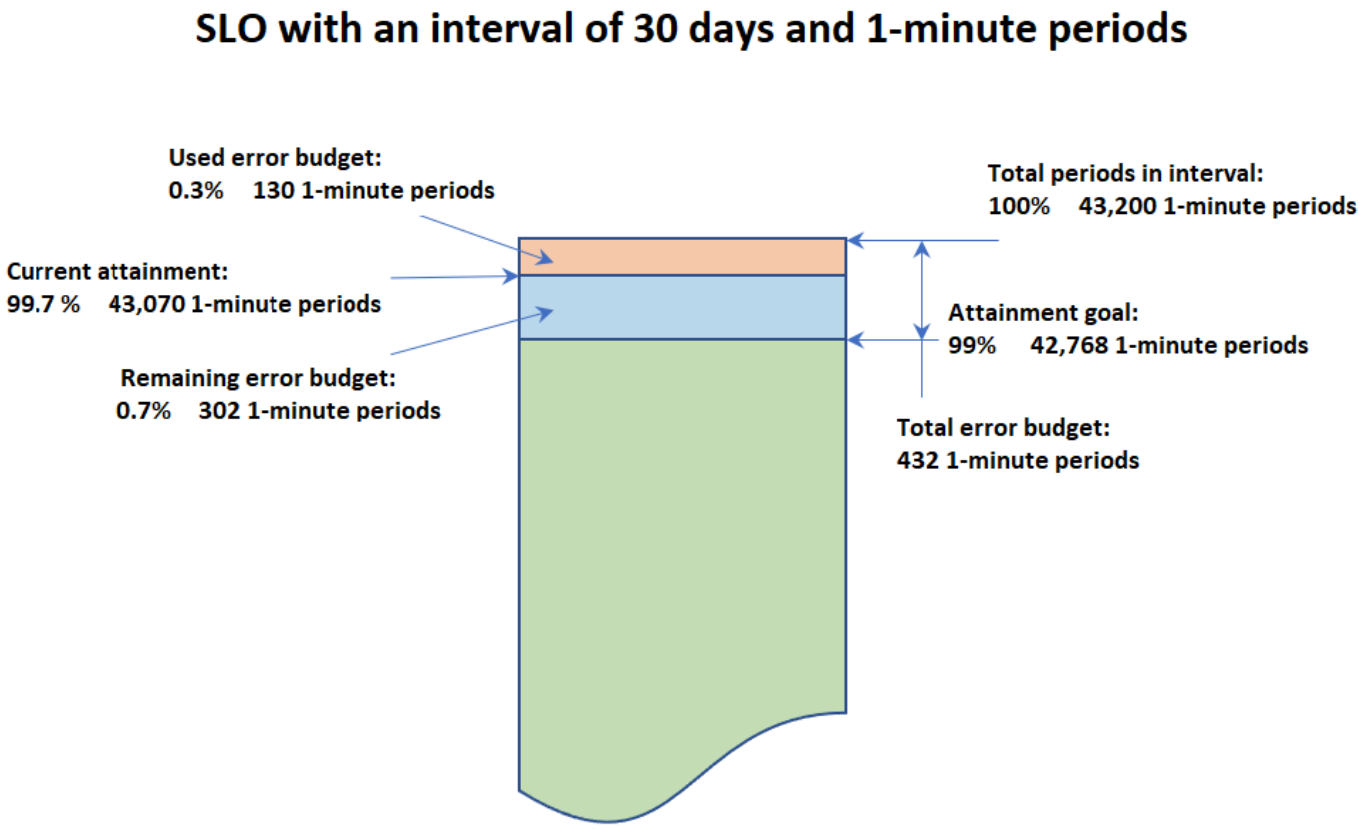
Un SLI peut être basé sur l'une des nouvelles métriques d'application standard collectées par Application Signals. Il peut également s'agir de n'importe quelle CloudWatch métrique ou expression métrique. Les métriques d'application standard que vous pouvez utiliser pour un SLI sont `Latency` et `Availability`. `Availability` représente le nombre de réponses réussies divisé par le nombre total de demandes. Il est calculé sous la forme (1 - taux de défaillance) * 100, les réponses aux défaillances étant des erreurs `5xx`. Les réponses positives sont des réponses sans erreur `5XX`. Les réponses `4XX` sont considérées comme réussies.

> **Note**
>
> Actuellement, seuls les calculs basés sur les périodes sont pris en charge. La prise en charge des calculs basés sur le volume ou les requêtes est prévue pour les prochaines versions.

Calcul du budget d'erreur et du résultat atteint

Lorsque vous consultez les informations relatives à un SLO, vous pouvez voir son état de santé actuel et son budget d'erreurs. Le budget d'erreur est le laps de temps compris dans l'intervalle pendant lequel il est possible de dépasser le seuil tout en permettant d'atteindre le SLO. Le budget d'erreurs total est la quantité totale de temps de dépassement qui peut être tolérée sur l'ensemble de l'intervalle. Le budget d'erreurs restant est le temps de dépassement restant qui peut être toléré pendant l'intervalle en cours. Ceci après avoir soustrait du budget d'erreur total le temps de dépassement qui s'est déjà produit.

La figure suivante illustre les concepts de budget de réalisation et d'erreur pour un objectif avec un intervalle de 30 jours, des périodes d'une minute et un objectif de réalisation de 99 %. 30 jours comprennent 43 200 périodes d'une minute. 99 % de 43 200, c'est 42 768, donc 42 768 minutes par mois doivent être saines pour que le SLO soit atteint. Jusqu'à présent, dans l'intervalle actuel, 130 des périodes d'une minute n'étaient pas saines.



Détermination du succès au cours de chaque période

Au cours de chaque période, les données du SLI sont agrégées en un seul point de données sur la base des statistiques utilisées pour le SLI. Ce point de données représente la durée totale de la période. Ce point de données unique est comparé au seuil SLI pour déterminer si la période est saine. L'affichage sur le tableau de bord des périodes non saines pendant l'intervalle de temps en cours peut avertir vos opérateurs de services que le service doit être trié.

S'il est déterminé que la période n'est pas saine, la durée totale de la période est prise en compte comme un échec dans le calcul du budget d'erreur. Le suivi du budget d'erreurs vous permet de savoir si le service atteint les performances souhaitées sur une longue période.

Création d'un SLO

Nous vous recommandons de définir des SLO de latence et de disponibilité pour vos applications critiques. Ces indicateurs collectés par Application Signals correspondent aux objectifs métier communs.

Vous pouvez également définir des SLO pour n'importe quelle CloudWatch métrique ou expression mathématique de métrique aboutissant à une seule série chronologique.

La première fois que vous créez un SLO dans votre compte, le rôle AWSServiceRoleForCloudWatchApplicationSignalslié au service est CloudWatch automatiquement créé dans votre compte, s'il n'existe pas déjà. Ce rôle lié au service permet de CloudWatch collecter des données de CloudWatch journal, des données de suivi X-Ray, des données CloudWatch métriques et des données de balisage à partir des applications de votre compte. Pour plus d'informations sur les rôles CloudWatch liés à un service, consultez. Utilisation des rôles liés aux services pour CloudWatch

Pour créer un SLO

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Objectifs de niveau de service (SLO).
3. Choisissez Créer un SLO.
4. Saisissez un nom pour le SLO. L'inclusion du nom d'un service ou d'une opération, ainsi que des mots clés appropriés tels que la latence ou la disponibilité, vous aidera à identifier rapidement ce que l'état du SLO indique lors du triage.
5. Pour Définir un indicateur de niveau de service (SLI), procédez de l'une des manières suivantes :
   - Pour définir le SLO sur l'une des métriques d'application standard `Latency` ou `Availability` :
     a. Choisissez Opération de service.
     b. Sélectionnez le service que ce SLO surveillera.
     c. Sélectionnez l'opération que ce SLO surveillera.

        Les listes déroulantes Sélectionner un service et Sélectionner une opération contiennent les services et les opérations qui ont été actifs au cours des dernières 24 heures.
     d. Choisissez Disponibilité ou Latence, puis définissez le seuil.
   - Pour définir le SLO sur une CloudWatch métrique ou une expression mathématique de CloudWatch métrique, procédez comme suit :
     a. Choisissez CloudWatch Metric.
     b. Choisissez Sélectionner une CloudWatch métrique.

L’écran Sélectionner une métrique apparaît. Utilisez les onglets Parcourir ou Requête pour trouver la métrique souhaitée, ou créez une expression mathématique de métrique.

Après avoir sélectionné la métrique souhaitée, choisissez l’onglet Graphiques des métriques et sélectionnez la Statistique et la Période à utiliser pour le SLO. Ensuite, choisissez Select metric (Sélectionner une métrique).

Pour plus d’informations sur ces écrans, veuillez consulter Représenter graphiquement une métrique et Ajouter une expression mathématique à un CloudWatch graphique.

c. Pour Définir la condition, sélectionnez un opérateur de comparaison et un seuil que le SLO utilisera comme indicateur de réussite.

6. Si vous avez sélectionné Opération de service à l’étape 5, vous pouvez éventuellement sélectionner Paramètres supplémentaires, puis ajuster la durée de la période pour ce SLO.
7. Définissez l’intervalle et l’objectif de réalisation pour le SLO. Pour plus d’informations sur les intervalles et les objectifs de réalisation et la manière dont ils fonctionnent ensemble, veuillez consulter.la rubrique Concepts SLO.
8. (Facultatif) Définissez une ou plusieurs CloudWatch alarmes ou un seuil d'avertissement pour le SLO.

   a. CloudWatch les alarmes peuvent utiliser Amazon SNS pour vous avertir de manière proactive si une application est défectueuse en fonction de ses performances SLI.

      Pour créer une alarme, cochez l’une des cases d’alarme et saisissez ou créez la rubrique Amazon SNS à utiliser pour les notifications lorsque l’alarme passe à l’état `ALARM`. Pour plus d'informations sur les CloudWatch alarmes, consultez Utilisation des CloudWatch alarmes Amazon. La création d’alarmes entraîne des frais. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

   b. Si vous définissez un seuil d’avertissement, celui-ci apparaît dans les écrans d’Application Signals pour vous aider à identifier les SLO qui risquent de ne pas être atteints, même s’ils sont actuellement sains.

      Pour définir un seuil d’avertissement, saisissez la valeur du seuil dans Seuil d’avertissement. Lorsque le budget d’erreur du SLO est inférieur au seuil d’avertissement, le SLO est marqué d’un Avertissement sur plusieurs écrans d’Application Signals. Les seuils d’avertissement

apparaissent également sur les graphiques du budget d'erreur. Vous pouvez également créer une Alarme d'avertissement SLO basée sur le seuil d'avertissement.

9. Pour ajouter des tags à ce SLO, choisissez l'onglet Balises, puis choisissez Ajouter une nouvelle balise. Les balises peuvent vous aider à gérer, identifier, organiser, rechercher et filtrer des ressources. Pour plus d'informations sur le balisage, veuillez consulter la rubrique Tagging your AWS resources.

   > **Note**
   >
   > Si l'application à laquelle cette SLO est associée est enregistrée AWS Service Catalog AppRegistry, vous pouvez utiliser la `awsApplication` balise pour associer cette SLO à cette application AppRegistry. Pour plus d'informations, voir Qu'est-ce que c'est AppRegistry ?

10. Choisissez Créer un SLO. Si vous avez également choisi de créer une ou plusieurs alarmes, le nom du bouton change en conséquence.

Afficher et trier le statut du SLO

Vous pouvez rapidement vérifier l'état de vos SLO à l'aide des objectifs de niveau de service ou des options de services de la CloudWatch console. La vue Services fournit une at-a-glance vue du ratio de services défectueux, calculé en fonction des SLO que vous avez définis. Pour plus d'informations sur l'utilisation de l'option Services, veuillez consulter la rubrique Surveillez l'état de fonctionnement de vos applications avec Application Signals.

La vue des Objectifs de niveau de service fournit une vue macro de votre organisation. Vous pouvez voir les SLO atteints et non atteints dans leur ensemble. Cela vous donne une idée du nombre de vos services et opérations qui répondent à vos attentes sur de longues périodes, en fonction des SLI que vous avez choisis.

Pour afficher tous vos SLO à l'aide de la vue Objectifs de niveau de service

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Objectifs de niveau de service (SLO).

   La liste des Objectifs de niveau de service (SLO) apparaît.

Vous pouvez rapidement voir l'état actuel de vos SLO dans la colonne État des SLI. Pour trier les SLO de manière à ce que tous ceux qui ne sont pas sains figurent en haut de la liste, choisissez la colonne État des SLI jusqu'à ce que les SLO non sains apparaissent tous en haut de la liste.

La table SLO comporte les colonnes par défaut suivantes. Vous pouvez ajuster les colonnes affichées en choisissant l'icône représentant un engrenage au-dessus de la liste. Pour plus d'informations sur les objectifs, les SLI, les résultats atteints et les intervalles, veuillez consulter la rubrique Concepts SLO.

- Le nom du SLO.
- La colonne Objectif affiche le pourcentage de périodes pendant chaque intervalle qui doivent atteindre le seuil SLI pour que l'objectif SLO soit atteint. Elle affiche également la durée de l'intervalle pour le SLO.
- État du SLI indique si l'état de fonctionnement actuel de l'application est sain ou non. Si une période quelconque de l'intervalle de temps sélectionné n'était pas saine pour le SLO, État du SLI indique Non sain.
- Le Niveau final est le niveau de réalisation atteint à la fin de l'intervalle de temps sélectionné. Triez selon cette colonne pour voir les SLO les plus susceptibles de ne pas être atteints.
- Le Delta d'atteinte est la différence de niveau de réalisation entre le début et la fin de l'intervalle de temps sélectionné. Un delta négatif signifie que la métrique suit une tendance à la baisse. Triez selon cette colonne pour voir les dernières tendances des SLO.
- Le budget d'erreur de fin (%) est le pourcentage du temps total de la période pendant laquelle il peut y avoir des périodes non saines tout en atteignant le SLO avec succès. Si vous définissez ce paramètre sur 5 % et que le SLI est défectueux pendant 5 % ou moins des périodes restantes de l'intervalle, le SLO est toujours atteint avec succès.
- Le Delta du budget d'erreur est la différence du budget d'erreur entre le début et la fin de l'intervalle de temps sélectionné. Un delta négatif signifie que la métrique suit une tendance défavorable.
- Le Budget d'erreur de fin (temps) est la durée réelle au sein de l'intervalle qui peut être non saine tout en permettant d'atteindre le SLO avec succès. Par exemple, si ce délai est de 14 minutes, si le SLI est non sain pendant moins de 14 minutes pendant l'intervalle restant, le SLO sera toujours atteint avec succès.
- Les colonnes Service, Opération et Type affichent des informations sur le service et l'opération pour lesquels ce SLO est configuré.

3. Pour afficher les graphiques du budget d'atteinte et d'erreur pour un SLO, choisissez la case d'option en regard du nom du SLO.

   Les graphiques en haut de la page indiquent le degré de réalisation du SLO et l'état du budget d'erreur. Un graphique concernant la métrique SLI associée à ce SLO est également affiché.
4. Pour mieux trier un SLO qui n'atteint pas son objectif, choisissez le nom du service ou le nom de l'opération associé à ce SLO. Vous êtes redirigé vers la page de détails où vous pouvez effectuer un tri plus approfondi. Pour plus d'informations, consultez Consultez le détail de l'activité des services et de l'état de fonctionnement sur la page détaillée des services.
5. Pour modifier la plage temporelle des graphiques et des tableaux de la page, choisissez un nouvel intervalle de temps en haut de l'écran.

Modification d'un SLO existant

Suivez ces étapes pour modifier un SLO existant. Lorsque vous modifiez un SLO, vous ne pouvez modifier que le seuil, l'intervalle, l'objectif de réalisation et les balises. Pour modifier d'autres aspects tels que le service, le fonctionnement ou les métriques, créez un SLO au lieu d'en modifier un existant.

La modification d'une partie de la configuration de base d'un SLO, telle que la période ou le seuil, invalide tous les points de données et évaluations précédents concernant les résultats et l'état de santé. En réalité, cela supprime et recrée le SLO.

> **Note**
>
> Si vous modifiez un SLO, les alarmes associées à ce SLO ne sont pas automatiquement mises à jour. Vous devrez peut-être mettre à jour les alarmes pour qu'elles restent synchronisées avec le SLO.

Pour modifier un SLO existant

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Objectifs de niveau de service (SLO).
3. Choisissez la case d'option en regard du SLO que vous souhaitez modifier, puis choisissez Actions, Modifier le SLO.
4. Effectuez les modifications, puis choisissez Enregistrer les modifications.

Suppression d’un SLO

Suivez ces étapes pour supprimer un SLO existant.

> **Note**
>
> Si vous supprimez un SLO, les alarmes associées à ce SLO ne sont pas automatiquement supprimées. Vous devrez les supprimer vous-même. Pour plus d’informations, consultez Gérer les alarmes.

Pour supprimer un SLO

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Objectifs de niveau de service (SLO).
3. Choisissez la case d’option en regard du SLO que vous souhaitez modifier, puis choisissez Actions, Supprimer le SLO.
4. Choisissez Confirmer.

Surveillez l’état de fonctionnement de vos applications avec Application Signals

> **Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.**

Utilisez les signaux d'application dans la CloudWatch console pour surveiller et résoudre les problèmes liés à l'état de fonctionnement de vos applications :

- Surveillez les services de votre application : dans le cadre de la surveillance opérationnelle quotidienne, utilisez la page Services pour consulter un résumé de tous vos services. Identifiez les services présentant le taux de défaillance ou le temps de latence le plus élevé, et identifiez les services présentant des indicateurs de niveau de service (SLI) non sains. Sélectionnez un service pour ouvrir la page Détails du service et consulter les indicateurs détaillés, les opérations

de service, les scripts canary Synthetics et les demandes des clients. Cela vous aide à résoudre les problèmes opérationnels et à identifier la cause première des problèmes opérationnels.

- Inspectez la topologie de votre application : utilisez la Carte des services pour comprendre et surveiller la topologie de votre application au fil du temps, y compris les relations entre les clients, les scripts canary de Synthetics, les services et les dépendances. Consultez instantanément l'état de l'indicateur de niveau de service (SLI) et consultez les indicateurs clés tels que le volume d'appels, le taux d'erreur et la latence. Accédez à des informations plus détaillées sur la page Détails du service.

Découvrez un exemple de scénario qui montre comment ces pages peuvent être utilisées pour résoudre rapidement un problème de santé d'un service opérationnel, de la détection initiale à l'identification de la cause première.

Comment Application Signals permettent de surveiller l'état de fonctionnement des opérations

Une fois que vous avez activé votre application pour Application Signals, vos services applicatifs, vos API et leurs dépendances sont automatiquement découverts et affichés dans les pages Services, Détails des Carte des services. Application Signals collecte des informations provenant de sources multiples pour permettre la découverte de services et la surveillance de l'état de fonctionnement :

- AWS Distro for OpenTelemetry (ADOT) — Dans le cadre de l'activation des signaux d'application, une bibliothèque d'auto-instrumentation OpenTelemetry Java est configurée pour émettre des métriques et des traces collectées par l'agent. CloudWatch Les métriques et les suivis sont utilisés pour permettre la découverte des services, des opérations, des dépendances et d'autres informations sur les services.
- Objectifs de niveau de service (SLO) : une fois que vous avez créé des objectifs de niveau de service pour vos services, les pages Services, Détails du service et Carte des services affichent l'état de l'indicateur de niveau de service (SLI). Les SLI peuvent surveiller la latence, la disponibilité et d'autres indicateurs opérationnels.
- CloudWatch Canaris synthétiques — Lorsque vous configurez le suivi par rayons X sur vos canaris, les appels adressés à vos services depuis vos scripts Canary sont associés à votre service et affichés sur la page détaillée du service.
- CloudWatch Surveillance des utilisateurs réels (RUM) — Lorsque le suivi X-Ray est activé sur votre client Web CloudWatch RUM, les demandes adressées à vos services sont automatiquement associées et affichées sur la page détaillée du service.

- AWS Service Catalog AppRegistry— Application Signals découvre automatiquement AWS les ressources de votre compte et vous permet de les regrouper dans des applications logiques créées dans AppRegistry. Le nom de l'application affiché sur la page Services est basé sur la ressource de calcul sous-jacente sur laquelle vos services s'exécutent.

> **Note**
>
> Application Signals affiche vos services et opérations en fonction des métriques et des suivis émis dans le filtre temporel actuel que vous avez choisi. (Par défaut, il s'agit des trois dernières heures.) S'il n'y a aucune activité dans le filtre temporel actuel pour un service, une opération, une dépendance, un script Canary Synthetics ou une page client, elle ne sera pas affichée.
>
> À l'heure actuelle, jusqu'à 1 000 services peuvent être affichés. La découverte de vos services et de leur topologie peut être retardée de 10 minutes maximum. L'évaluation de l'état de votre indicateur de niveau de service (SLI) peut être retardée jusqu'à 15 minutes.

Affichage de l'activité globale des services et de l'état de fonctionnement sur la page Services

> **Application Signals est en version préliminaire pour Amazon CloudWatch et est susceptible d'être modifiée.**

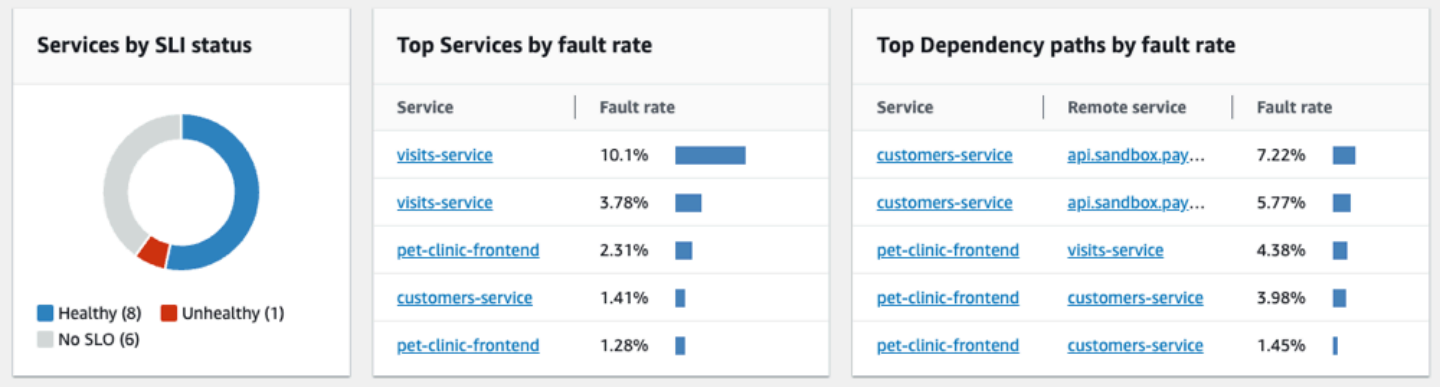
Utilisez la page Services pour voir la liste de vos services qui sont activés pour Application Signals. Vous pouvez également consulter les métriques opérationnelles et identifier rapidement les services dont les indicateurs de niveau de service (SLI) ne sont pas sains. Analysez les anomalies de performance à mesure que vous identifiez la cause première des problèmes opérationnels. Pour afficher cette page, ouvrez la CloudWatch console et choisissez Services dans la section Signaux d'application dans le volet de navigation de gauche.

Explorez les indicateurs d'état de fonctionnement de vos services

Le haut de la page Services comprend un graphique de l'état de fonctionnement global des services et plusieurs tableaux présentant les principaux services et leurs dépendances par taux de défaillance. Le graphique des services sur la gauche affiche une ventilation du nombre de services présentant

des indicateurs de niveau de service (SLI) sains ou non sains pendant le filtre temporel actuel au niveau de la page. Les SLI peuvent surveiller la latence, la disponibilité et d'autres indicateurs opérationnels.

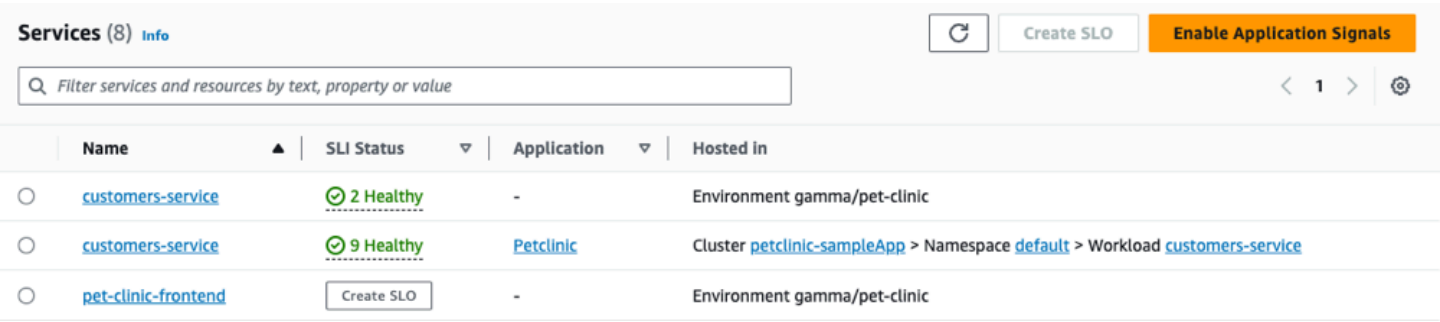
Les deux tableaux situés à côté du graphique présentent une liste des principaux services par taux de défaillance. Choisissez un nom de service dans l'une ou l'autre des tables pour ouvrir une page de détails du service et voir les détails de l'opération de service. Choisissez un chemin de dépendance pour ouvrir la page détaillée et voir les détails des dépendances de service. Les deux tableaux affichent des informations relatives aux trois dernières heures, même si un filtre de période plus longue est sélectionné en haut à droite de la page.



Surveillez l'état de fonctionnement à l'aide du tableau Services

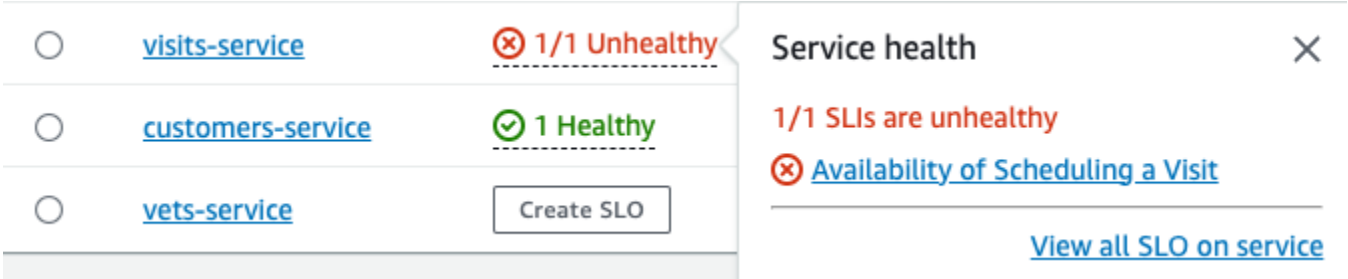
Le tableau Services affiche la liste de vos services qui ont été activés pour Application Signals. Choisissez Activer Application Signals pour ouvrir une page de configuration et commencer à configurer vos services. Pour plus d'informations, veuillez consulter la rubrique Activer Application Signals.

Filtrez le tableau Services pour faciliter la recherche en sélectionnant une ou plusieurs propriétés dans la zone de texte du filtre. Au fur et à mesure que vous choisissez chaque propriété, vous êtes guidé à travers les critères de filtrage. Vous verrez le filtre complet sous la zone de texte du filtre. Choisissez Effacer les filtres à tout moment pour supprimer le filtre du tableau.



Choisissez le nom de n'importe quel service dans le tableau pour afficher une page de détails du service contenant des métriques de niveau de service, des opérations et des détails supplémentaires. Si vous avez associé la ressource de calcul sous-jacente du service à une application dans AppRegistry ou dans la carte Applications de la page d' AWS Management Console accueil, choisissez le nom de l'application pour afficher les détails de l'application sur la page de console MyApplications. Pour les services hébergés dans Amazon EKS, choisissez n'importe quel lien dans la colonne Hébergé dans pour afficher le cluster, l'espace de noms ou la charge de travail dans CloudWatch Container Insights. Pour les services exécutés sur Amazon ECS ou Amazon EC2, la valeur de l'environnement est affichée.

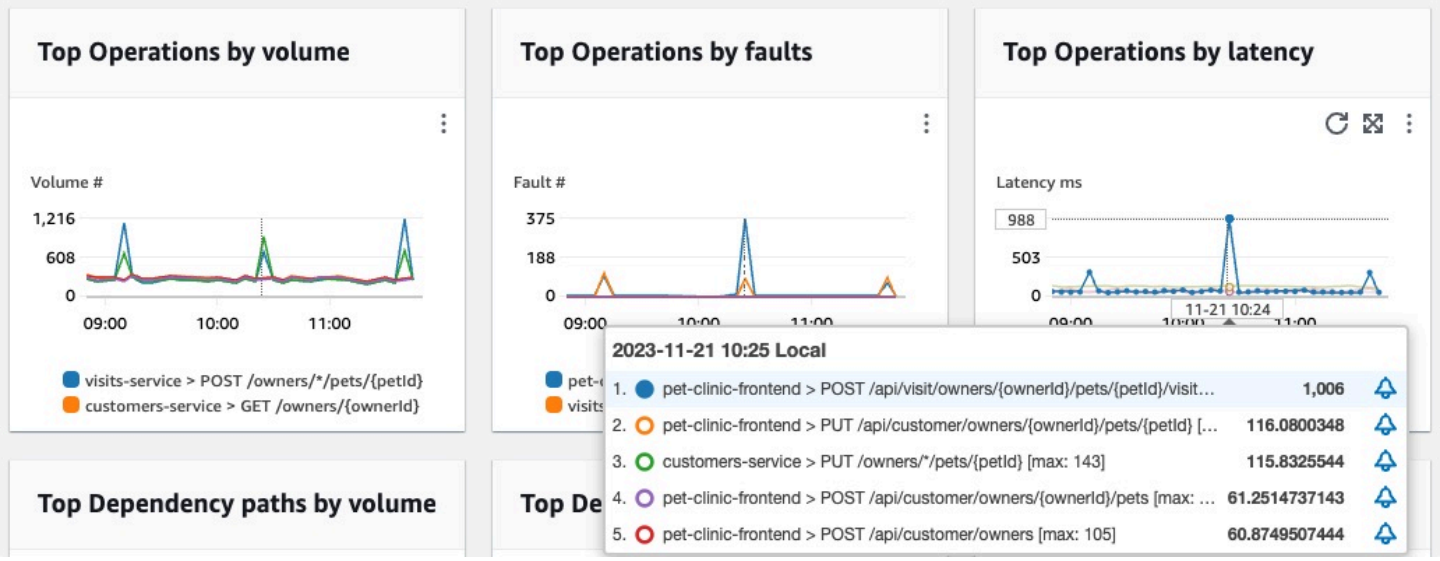
L'état de l'Indicateur de niveau de service (SLI) est affiché pour chaque service dans le tableau. Choisissez l'état du SLI d'un service pour afficher une fenêtre contextuelle contenant un lien vers les SLI non sain et un lien pour afficher tous les SLO associés au service.



Si aucun SLO n'a été créé pour un service, cliquez sur le bouton Créer un SLO dans la colonne État du SLI. Pour créer des SLO supplémentaires pour un service, sélectionnez le bouton radio à côté du nom du service, puis choisissez Créer un SLO dans le menu déroulant Actions en haut à droite du tableau. Lorsque vous créez des SLO, vous pouvez voir en un coup d'œil quels services et opérations fonctionnent bien et ceux qui sont non sains. Consultez les objectifs de niveau de service (SLO) pour plus d'informations.

Afficher les principales métriques de fonctionnement et de dépendance

Sous le tableau des services, vous pouvez consulter les principales opérations et dépendances de tous les services par volume d'appels, défaillances et latence. Cet ensemble de graphiques fournit des informations essentielles sur les opérations ou les dépendances susceptibles de ne pas fonctionner correctement dans tous les services. Choisissez n'importe quel point d'un graphique pour afficher une fenêtre contextuelle contenant des informations plus détaillées sur la série. Survolez les descriptions des séries au bas d'un graphique pour afficher une fenêtre contextuelle contenant des métriques détaillées pour une opération ou un chemin de dépendance spécifique. Sélectionnez le bouton du menu contextuel dans le coin supérieur droit d'un graphique pour afficher des options supplémentaires, notamment l'affichage des CloudWatch statistiques ou des pages de journaux.



Consultez le détail de l'activité des services et de l'état de fonctionnement sur la page détaillée des services

> ⚠ Application Signals est en version préliminaire pour Amazon CloudWatch et est susceptible d'être modifiée.

Lorsque vous instrumentez votre application, Amazon CloudWatch Application Signals cartographie tous les services découverts par votre application. Utilisez la page détaillée du service pour avoir un aperçu de vos services, opérations, dépendances, canaries et demandes des clients pour un service unique. Pour afficher la page détaillée du service, procédez comme suit :

- Ouvrez la CloudWatch console.
- Choisissez Services dans la section Signaux d'application dans le volet de navigation de gauche.
- Choisissez le nom de n'importe quel service dans les tables Services, Top services ou dépendances.

La page détaillée du service est organisée dans les onglets suivants :

- Vue d'ensemble : utilisez cet onglet pour obtenir une vue d'ensemble d'un service, notamment le nombre d'opérations, de dépendances, de pages synthétiques et de pages client. L'onglet affiche les indicateurs clés pour l'ensemble de votre service, les principales opérations et les

dépendances. Ces mesures incluent des séries chronologiques sur la latence, les défaillances et les erreurs liées à toutes les opérations de service associées à ce service.

- Opérations de service : utilisez cet onglet pour consulter la liste des opérations effectuées par votre service, ainsi que des graphiques interactifs présentant des indicateurs clés qui mesurent l'état de santé de chaque opération. Vous pouvez sélectionner un point de données dans un graphique pour obtenir des informations sur les traces, les journaux ou les métriques associés à ce point de données.
- Dépendances : utilisez cet onglet pour consulter la liste des dépendances que votre service appelle, ainsi que la liste des mesures relatives à ces dépendances.
- Canaris synthétiques : utilisez cet onglet pour consulter la liste des canaris synthétiques simulant les appels des utilisateurs à votre service, ainsi que les principaux indicateurs de performance indiquant comment ces canaris sont utilisés.
- Pages clients : utilisez cet onglet pour consulter la liste des pages clients qui font appel à votre service, ainsi que les indicateurs qui mesurent la qualité des interactions des clients avec votre application.

Afficher l'aperçu de votre service

Utilisez la page de présentation des services pour afficher un résumé détaillé des mesures relatives à toutes les opérations de service au même endroit. Vérifiez les performances de toutes les opérations, dépendances, pages clients et canaris synthétiques qui interagissent avec votre application. Utilisez ces informations pour vous aider à déterminer où concentrer vos efforts afin d'identifier les problèmes, de résoudre les erreurs et de trouver des opportunités d'optimisation.

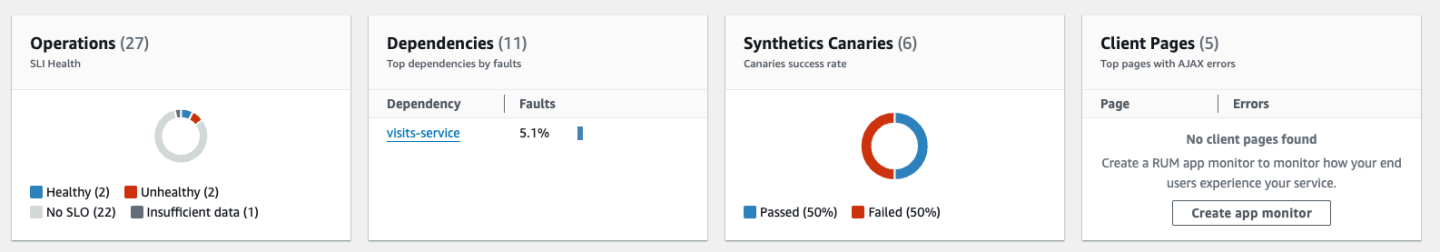
Cliquez sur n'importe quel lien dans Détails du service pour afficher les informations relatives à un service spécifique. Par exemple, pour les services hébergés dans Amazon EKS, la page de détails du service affiche les informations relatives au cluster, à l'espace de nommage et à la charge de travail. Pour les services hébergés sur Amazon ECS ou Amazon EC2, la page des détails du service indique la valeur de l'environnement.

Sous Services, l'onglet Vue d'ensemble affiche un résumé des éléments suivants :

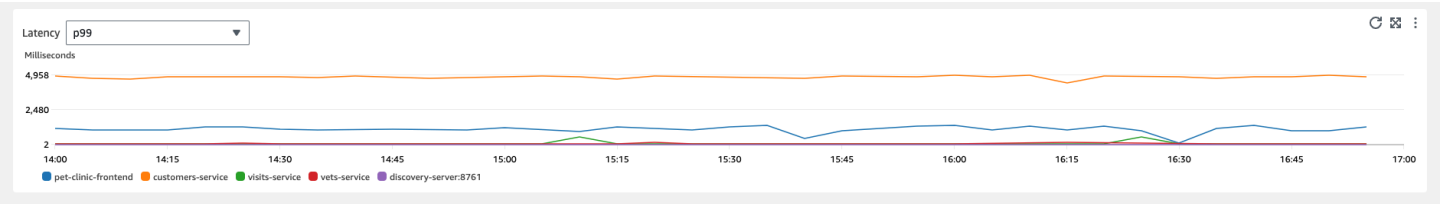
- Opérations — Utilisez cet onglet pour voir l'état de vos opérations de service. L'état de santé est déterminé par des indicateurs de niveau de service (SLI) définis dans le cadre d'un objectif de niveau de service (SLO).
- Dépendances : utilisez ce tableau pour voir les principales dépendances des services appelés par votre application, répertoriées par taux d'erreur.

- Synthetics canaries : utilisez cet onglet pour voir le résultat des appels simulés aux endpoints ou aux API associés à votre service, ainsi que le nombre de canaris ayant échoué.
- Pages client : utilisez cet onglet pour voir les principales pages appelées par des clients présentant des erreurs asynchrones JavaScript et XML (AJAX).

L'illustration suivante donne un aperçu de vos services :



L'onglet Vue d'ensemble affiche également un graphique des dépendances présentant la latence la plus élevée pour tous les services. Utilisez les métriques de latence p99, p90 et p50 pour évaluer rapidement les dépendances qui contribuent à la latence totale de votre service, comme suit :

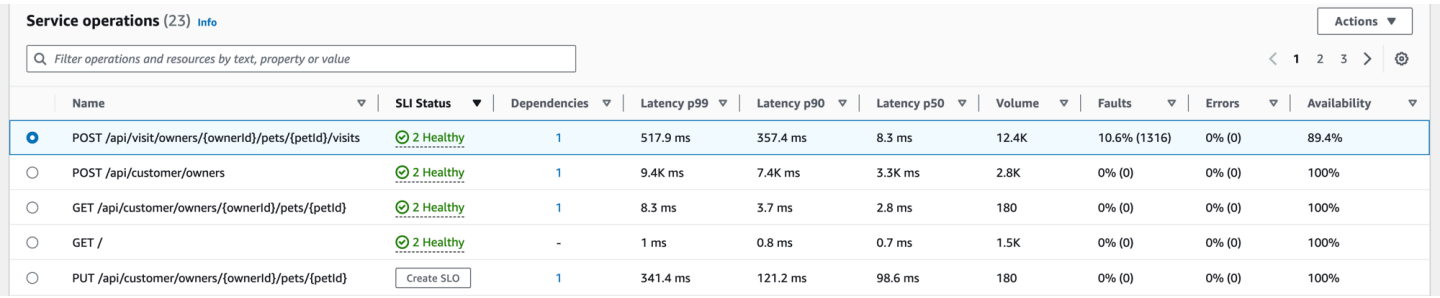


Par exemple, le graphique précédent montre que 99 % des demandes adressées à la dépendance au service client ont été traitées en environ 4 950 millisecondes. Les autres dépendances ont pris moins de temps.

Les graphiques présentant les quatre principales opérations de service en termes de latence indiquent le volume de demandes, la disponibilité, le taux d'erreur et le taux d'erreur pour ces services, comme indiqué dans l'image suivante :

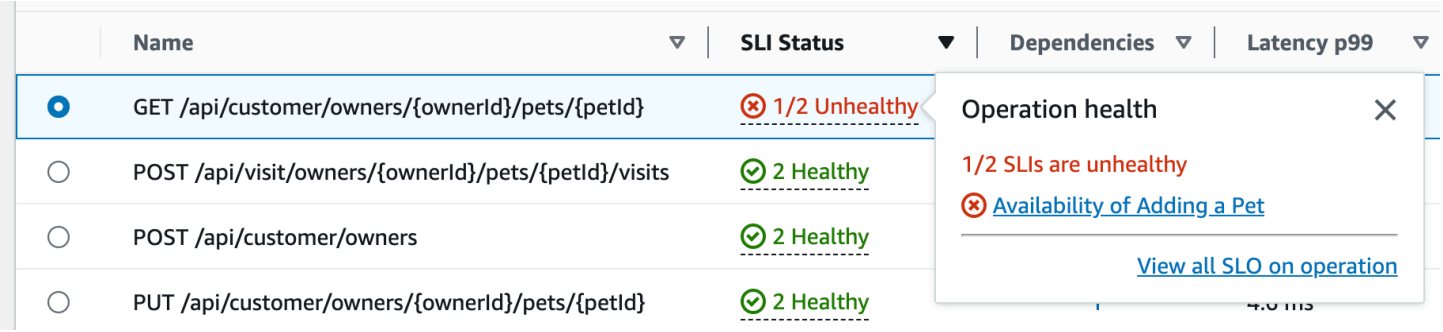
Affichage de vos opérations de service

Lorsque vous instrumentez votre application, Application Signals découvre toutes les opérations de service que votre application appelle. Utilisez l'onglet Opérations de service pour consulter un tableau contenant les opérations de service et un ensemble de mesures qui mesurent les performances d'une opération sélectionnée. Ces indicateurs incluent l'état du SLI, le nombre de dépendances, la latence, le volume, les défaillances, les erreurs et la disponibilité, comme le montre l'image suivante :



Filtrez le tableau pour trouver plus facilement une opération de service en choisissant une ou plusieurs propriétés dans la zone de texte du filtre. Au fur et à mesure que vous choisissez chaque propriété, vous êtes guidé à travers les critères de filtrage et vous verrez le filtre complet sous la zone de texte du filtre. Choisissez Effacer les filtres à tout moment pour supprimer le filtre du tableau.

Choisissez l'état SLI d'une opération pour afficher une fenêtre contextuelle contenant un lien vers tout SLI défectueux, ainsi qu'un lien permettant de voir tous les SLO associés à l'opération, comme indiqué dans le tableau suivant :



Le tableau des opérations de service répertorie l'état des SLI, le nombre de SLI sains ou défectueux et le nombre total de SLO pour chaque opération.

Utilisez les SLI pour surveiller la latence, la disponibilité et d'autres indicateurs opérationnels qui mesurent la santé opérationnelle d'un service. Utilisez un SLO pour vérifier les performances et l'état de santé de vos services et opérations.

Pour créer un SLO, procédez comme suit :

- Si une opération n'a pas de SLO, cliquez sur le bouton Create SLO dans la colonne SLI Status.
- Si une opération possède déjà un SLO, procédez comme suit :
  - Sélectionnez le bouton radio à côté du nom de l'opération.
  - Choisissez Create SLO dans la flèche Actions vers le bas en haut à droite du tableau.

Pour plus d’informations, veuillez consulter la rubrique Objectifs de niveau de service (SLO).

La colonne Dépendances indique le nombre de dépendances appelées par cette opération. Choisissez ce numéro pour ouvrir l’onglet Dépendances filtré en fonction de l’opération sélectionnée.

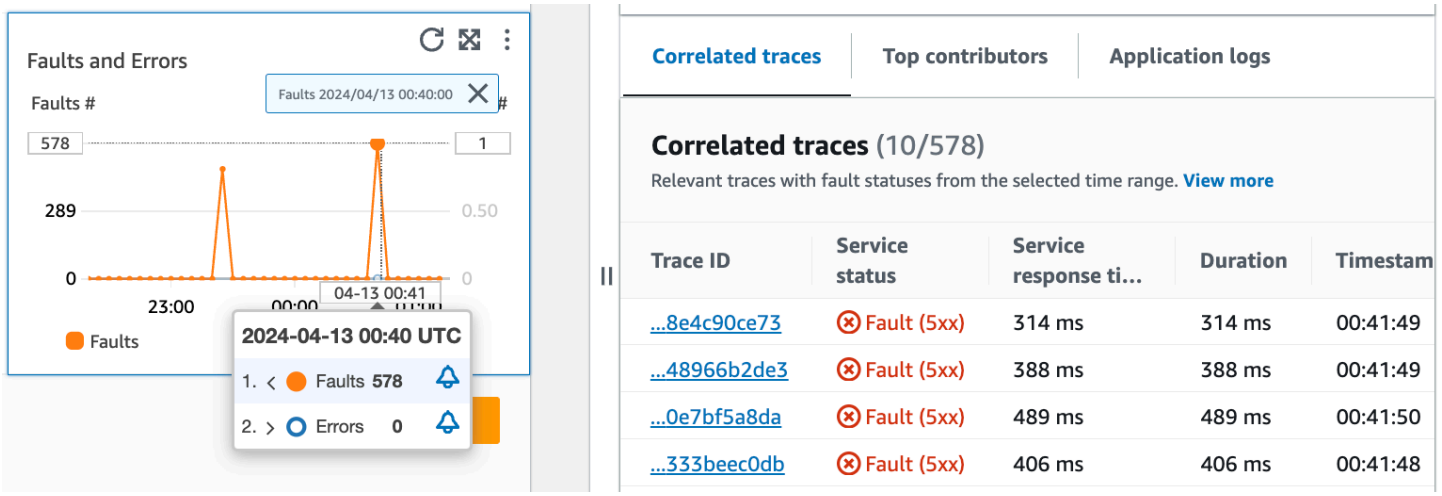
Afficher les métriques des opérations de service, les traces corrélées et les journaux des applications

Application Signals met en corrélation les indicateurs de fonctionnement des services avec les AWS X-Ray traces, CloudWatch Container Insights et les journaux des applications. Utilisez ces mesures pour résoudre les problèmes de santé opérationnelle. Pour afficher les métriques sous forme d'informations graphiques, procédez comme suit :

1. Sélectionnez une opération de service dans le tableau des opérations de service pour voir un ensemble de graphiques correspondant à l'opération sélectionnée au-dessus du tableau avec des mesures relatives au volume et à la disponibilité, à la latence, aux défaillances et aux erreurs.

2. Passez le pointeur de la souris sur un point d'un graphique pour afficher plus d'informations.
3. Sélectionnez un point pour ouvrir un volet de diagnostic qui affiche les traces, les mesures et les journaux d'application corrélés pour le point sélectionné dans le graphique.

L'image suivante montre l'infobulle qui apparaît lorsque vous survolez un point du graphique et le volet de diagnostic qui apparaît après avoir cliqué sur un point. L'infobulle contient des informations sur le point de données associé dans le graphique Faults and Errors. Le volet contient les traces corrélées, les principaux contributeurs et les journaux d'application associés au point sélectionné.



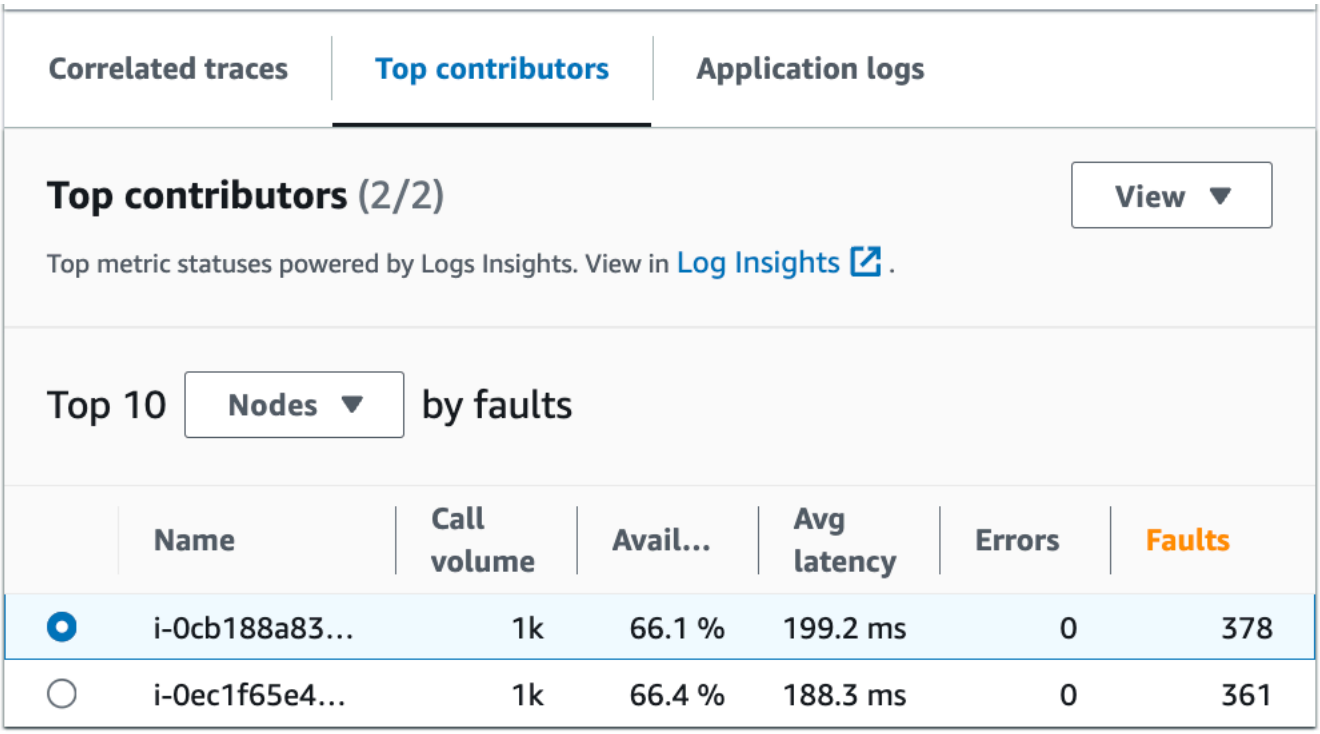
Traces corrélées

Examinez les traces associées pour comprendre un problème sous-jacent lié à une trace. Vous pouvez vérifier si les traces corrélées ou les nœuds de service qui leur sont associés se comportent de la même manière. Pour examiner les traces corrélées, choisissez un ID de trace dans le tableau des traces corrélées pour ouvrir la page des détails des traces X-Ray pour la trace choisie. La page des détails du suivi contient une carte des nœuds de service associés au suivi sélectionné et une chronologie des segments du suivi.

Principaux contributeurs

Consultez les principaux contributeurs pour trouver les principales sources d'entrée d'une métrique. Regroupez les contributeurs par différents composants afin de rechercher des similitudes au sein du groupe et de comprendre en quoi le comportement de traçage diffère entre eux.

L'onglet Principaux contributeurs fournit des mesures relatives au volume des appels, à la disponibilité, à la latence moyenne, aux erreurs et aux défauts pour chaque groupe. L'exemple d'image suivant montre les principaux contributeurs à une suite de métriques pour une application déployée sur une plateforme Amazon EKS :



Les principaux contributeurs contiennent les indicateurs suivants :

- Volume d'appels : utilisez le volume d'appels pour comprendre le nombre de demandes par intervalle de temps pour un groupe.
- Disponibilité : utilisez la disponibilité pour connaître le pourcentage de temps pendant lequel aucun défaut n'a été détecté pour un groupe.
- Latence moyenne : utilisez le temps de latence pour vérifier la durée moyenne d'exécution des demandes pour un groupe sur un intervalle de temps qui dépend de la date à laquelle les demandes que vous étudiez ont été effectuées. Les demandes effectuées moins de 15 jours auparavant sont évaluées à intervalles d'une minute. Les demandes faites entre 15 et 30 jours auparavant inclus sont évaluées à intervalles de 5 minutes. Par exemple, si vous étudiez des demandes qui ont provoqué une erreur il y a 15 jours, la métrique du volume d'appels est égale au nombre de demandes par intervalle de 5 minutes.
- Erreurs : nombre d'erreurs par groupe mesurées sur un intervalle de temps.
- Défauts : nombre de défauts par groupe sur un intervalle de temps.

Principaux contributeurs utilisant Amazon EKS ou Kubernetes

Utilisez les informations sur les principaux contributeurs aux applications déployées sur Amazon EKS ou Kubernetes pour consulter les indicateurs de santé opérationnelle regroupés par nœud, pod et PodTemplateHash. Les définitions suivantes s'appliquent :

- Un pod est un groupe d'un ou de plusieurs Docker conteneurs qui partagent le stockage et les ressources. Un pod est la plus petite unité qui peut être déployée sur une Kubernetes plate-forme. Regroupez-les par modules pour vérifier si les erreurs sont liées à des limitations spécifiques aux pods.
- Un nœud est un serveur qui exécute des pods. Regroupez-les par nœuds pour vérifier si les erreurs sont liées à des limitations spécifiques aux nœuds.
- Un hachage de modèle de pod est utilisé pour trouver une version particulière d'un déploiement. Regroupez par hachage du modèle de module pour vérifier si des erreurs sont liées à un déploiement particulier.

Principaux contributeurs utilisant Amazon EC2

Utilisez les informations sur les principaux contributeurs aux applications déployées sur Amazon EKS pour consulter les indicateurs de santé opérationnelle regroupés par ID d'instance et par groupe de dimensionnement automatique. Les définitions suivantes s'appliquent :

- Un ID d'instance est un identifiant unique pour l'instance Amazon EC2 exécutée par votre service. Regroupez par ID d'instance pour vérifier si les erreurs sont liées à une instance Amazon EC2 spécifique.
- Un groupe de dimensionnement automatique est un ensemble d'instances Amazon EC2 qui vous permettent d'augmenter ou de réduire les ressources dont vous avez besoin pour répondre aux demandes de vos applications. Regroupez par groupe de mise à l'échelle automatique si vous souhaitez vérifier si les erreurs sont limitées aux instances du groupe.

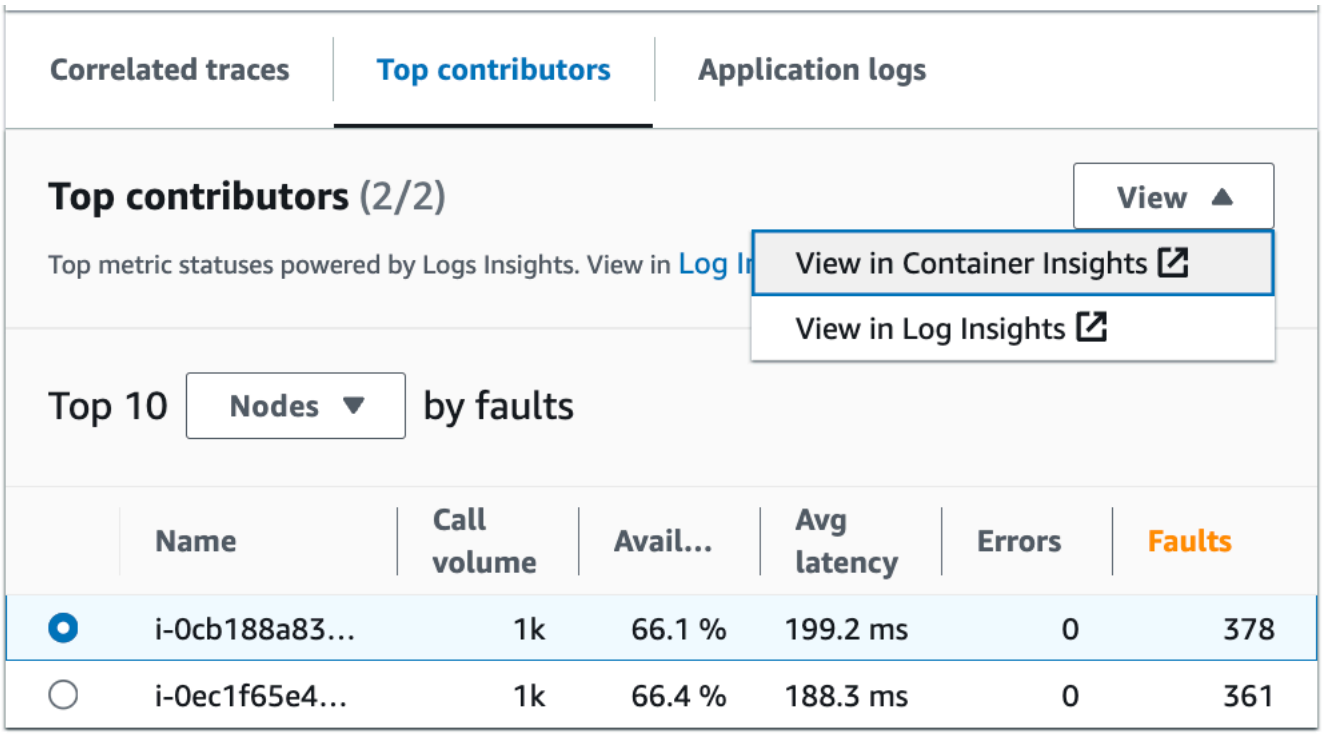
Principaux contributeurs utilisant une plateforme personnalisée

Utilisez les informations relatives aux principaux contributeurs aux applications déployées à l'aide d'instruments personnalisés pour consulter les indicateurs de santé opérationnelle regroupés par nom d'hôte. Les définitions suivantes s'appliquent :

- Un nom d'hôte identifie un appareil tel qu'un point de terminaison ou une instance Amazon EC2 connecté à un réseau. Regroupez par nom d'hôte pour vérifier si vos erreurs sont liées à un périphérique physique ou virtuel spécifique.

Afficher les meilleurs contributeurs dans Log Insights et Container Insights

Consultez et modifiez la requête automatique qui a généré des statistiques pour vos principaux contributeurs dans Log Insights. Consultez les indicateurs de performance de l'infrastructure par groupes spécifiques tels que les pods ou les nœuds dans Container Insights. Vous pouvez trier les clusters, les nœuds ou les charges de travail en fonction de la consommation de ressources, identifier rapidement les anomalies ou atténuer les risques de manière proactive avant que l'expérience de l'utilisateur final ne soit affectée. L'image ci-dessous montre comment sélectionner ces options :



Dans Container Insights, vous pouvez consulter les métriques de votre conteneur Amazon EKS ou Amazon ECS spécifiques au groupe de vos principaux contributeurs. Par exemple, si vous avez regroupé par module un conteneur EKS afin de générer les meilleurs contributeurs, Container Insights affichera les métriques et les statistiques filtrées pour votre module.

Dans Log Insights, vous pouvez modifier la requête qui a généré les statistiques sous Principaux contributeurs en procédant comme suit :

1. Sélectionnez Afficher dans Log Insights. La page Logs Insights qui s'ouvre contient une requête générée automatiquement et contient les informations suivantes :

   - Le nom du groupe de clusters de journaux.
   - L'opération sur laquelle vous enquêtiez CloudWatch.

- L'agrégat de la métrique de santé opérationnelle interagit sur le graphique.

Les résultats du journal sont automatiquement filtrés pour afficher les données des cinq dernières minutes avant que vous ne sélectionniez le point de données sur le graphique de service.

2. Pour modifier la requête, remplacez le texte généré par vos modifications. Vous pouvez également utiliser le générateur de requêtes pour générer une nouvelle requête ou mettre à jour la requête existante.

Journaux d'application

Utilisez la requête dans l'onglet Journaux de l'application pour générer des informations enregistrées pour votre groupe de journaux et votre service actuels et insérez un horodatage. Un groupe de journaux est un groupe de flux de journaux que vous pouvez définir lorsque vous configurez votre application.

Utilisez un groupe de journaux pour organiser les journaux présentant des caractéristiques similaires, notamment les suivantes :

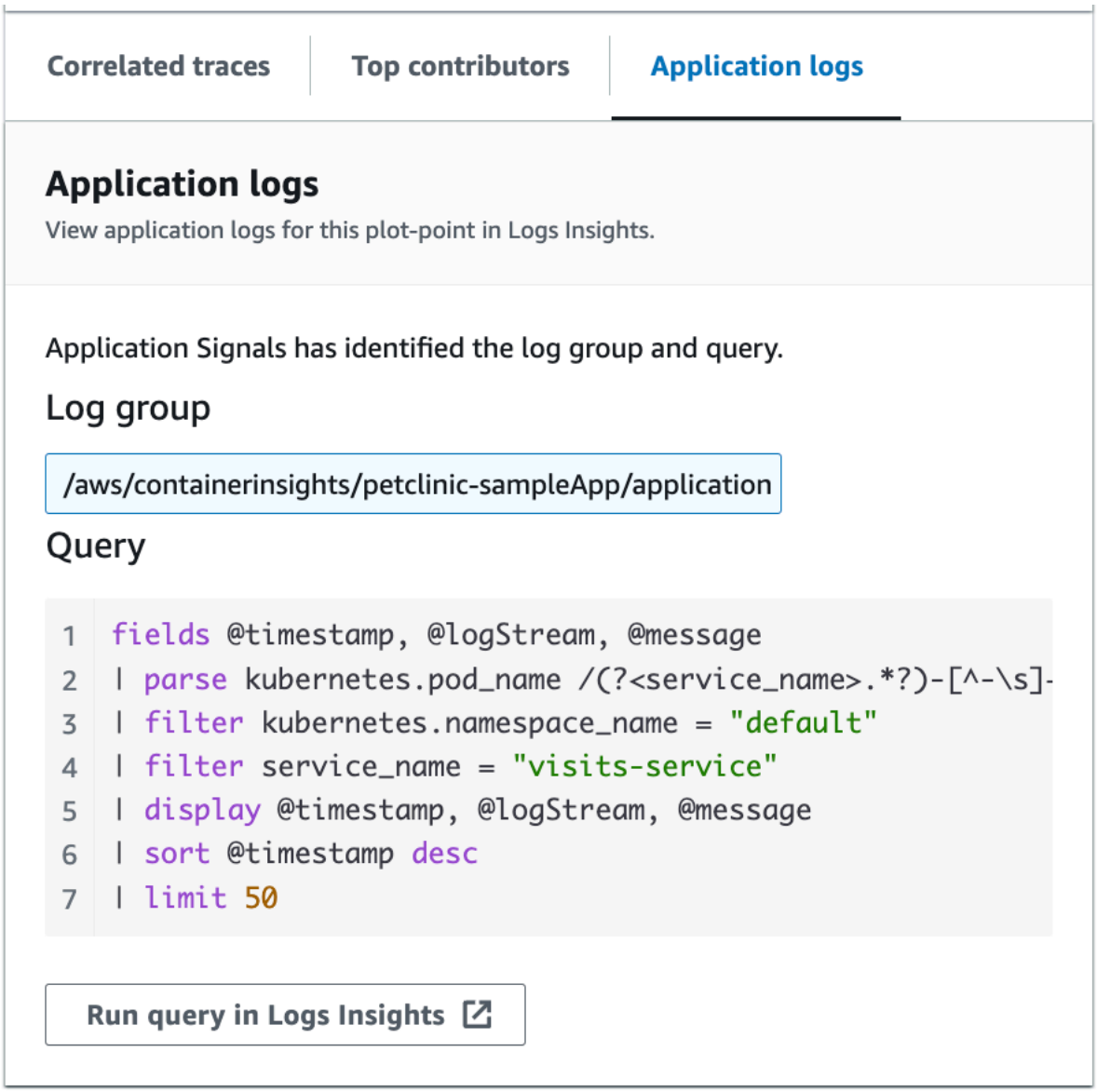
- Capturez les journaux d'une organisation, d'une source ou d'une fonction spécifique.
- Capturez les journaux auxquels un utilisateur en particulier accède.
- Capturez les journaux pour une période donnée.

Utilisez ces flux de journaux pour suivre des groupes ou des périodes spécifiques. Vous pouvez également configurer des règles de surveillance, des alarmes et des notifications pour ces groupes de journaux. Pour plus d'informations sur les groupes de journaux, consultez la section Utilisation des groupes de journaux et des flux de journaux.

La requête des journaux de l'application renvoie les journaux, les modèles de texte récurrents et les visualisations graphiques de vos groupes de journaux.

Pour exécuter la requête, sélectionnez Exécuter la requête dans Logs Insights pour exécuter la requête générée automatiquement ou modifier la requête. Pour modifier la requête, remplacez le texte généré automatiquement par vos modifications. Vous pouvez également utiliser le générateur de requêtes pour générer une nouvelle requête ou mettre à jour la requête existante.

L'image suivante montre l'exemple de requête généré automatiquement en fonction du point sélectionné dans le graphique des opérations de service :



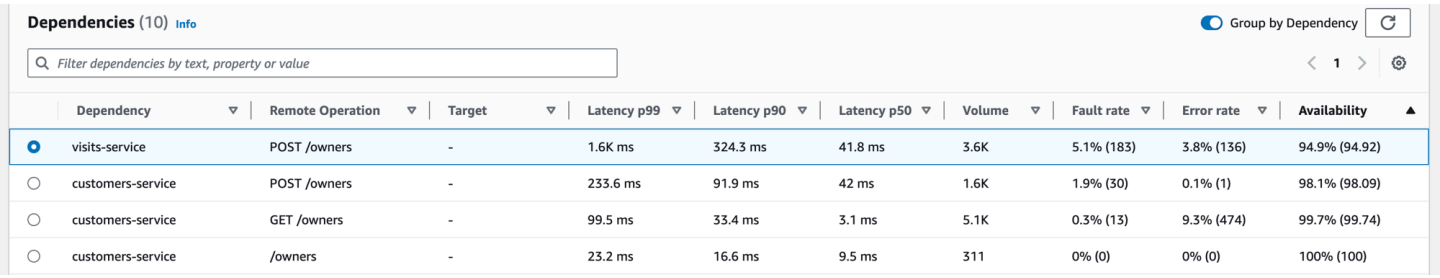
Dans l'image précédente, CloudWatch a automatiquement détecté le groupe de log associé au point sélectionné et l'a inclus dans une requête générée.

Affichage de vos dépendances de service

Choisissez l'onglet Dépendances pour afficher le tableau Dépendances et un ensemble de métriques relatives aux dépendances de toutes les opérations de service ou d'une seule opération. Le tableau contient une liste des dépendances découvertes par Application Signals, notamment les métriques relatives à la latence, au volume d'appels, au taux d'erreur, au taux d'erreur et à la disponibilité.

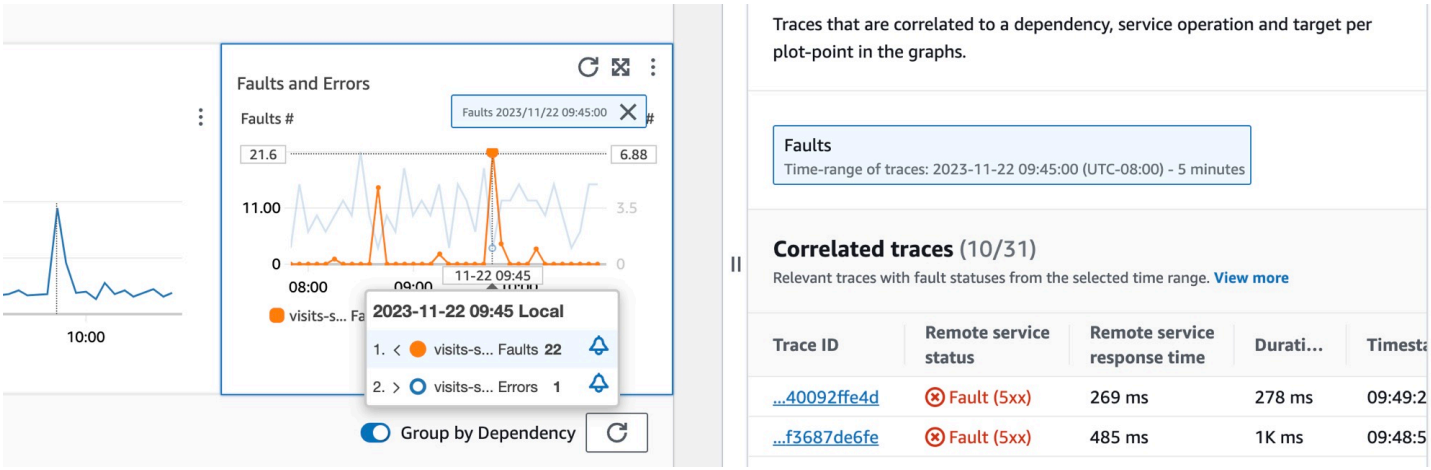
En haut de la page, choisissez une opération dans la liste des flèches vers le bas pour afficher ses dépendances, ou choisissez Tout pour voir les dépendances de toutes les opérations.

Filtrez le tableau pour faciliter la recherche de ce que vous recherchez en choisissant une ou plusieurs propriétés dans la zone de texte du filtre. Au fur et à mesure que vous choisissez chaque propriété, vous êtes guidé à travers les critères de filtrage et vous verrez le filtre complet sous la zone de texte du filtre. Choisissez Effacer les filtres à tout moment pour supprimer le filtre du tableau. Sélectionnez Grouper par dépendance en haut à droite du tableau pour regrouper les dépendances par nom de service et d'opération. Lorsque le regroupement est activé, développez ou réduisez un groupe de dépendances à l'aide de l'icône + à côté du nom de la dépendance.



La colonne Dépendance affiche le nom du service de dépendance, tandis que la colonne Opération à distance affiche le nom de l'opération de service. Lorsque vous appelez AWS des services, la colonne Target affiche la AWS ressource, telle que la table DynamoDB ou la file d'attente Amazon SNS.

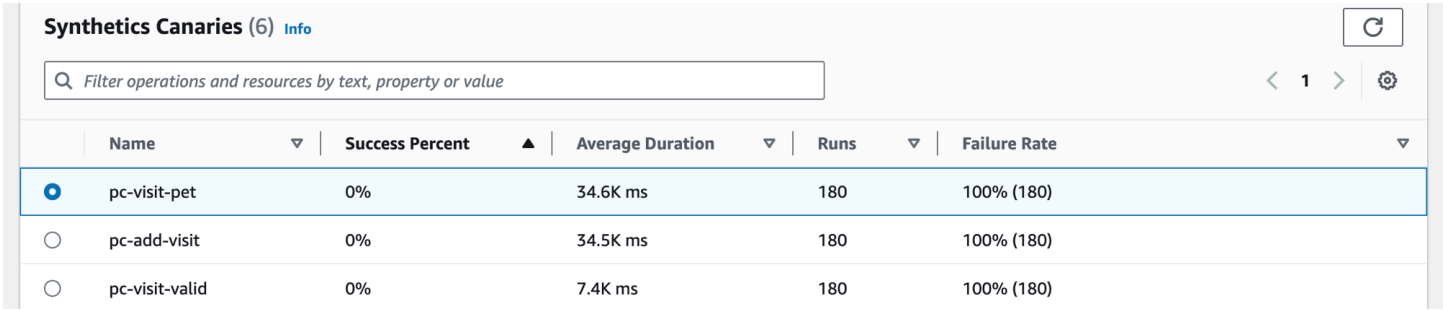
Pour sélectionner une dépendance, sélectionnez l'option située à côté d'une dépendance dans le tableau Dépendances. Cela montre un ensemble de graphiques qui affichent des mesures détaillées concernant le volume d'appels, la disponibilité, les défaillances et les erreurs. Passez le pointeur de la souris sur un point d'un graphique pour afficher une fenêtre contextuelle contenant plus d'informations. Sélectionnez un point dans un graphique pour ouvrir un volet de diagnostic qui affiche les traces corrélées pour le point sélectionné dans le graphique. Choisissez un ID de trace dans le tableau des traces corrélées pour ouvrir la page de détails de X-Ray Trace pour la trace sélectionnée.



Affichage de vos scripts canary Synthetics

Choisissez l'onglet Scripts canary Synthetics pour afficher le tableau Scripts canary Synthetics et un ensemble de métriques pour chaque script canary du tableau. Le tableau inclut des mesures relatives au pourcentage de réussite, à la durée moyenne, aux cycles et au taux d'échec. Seuls les canaris dont le AWS X-Ray traçage est activé sont affichés.

Utilisez la zone de texte du filtre dans le tableau des canaris synthétiques pour trouver le canari qui vous intéresse. Chaque filtre que vous créez apparaît sous la zone de texte du filtre. Choisissez Effacer les filtres à tout moment pour supprimer le filtre du tableau.



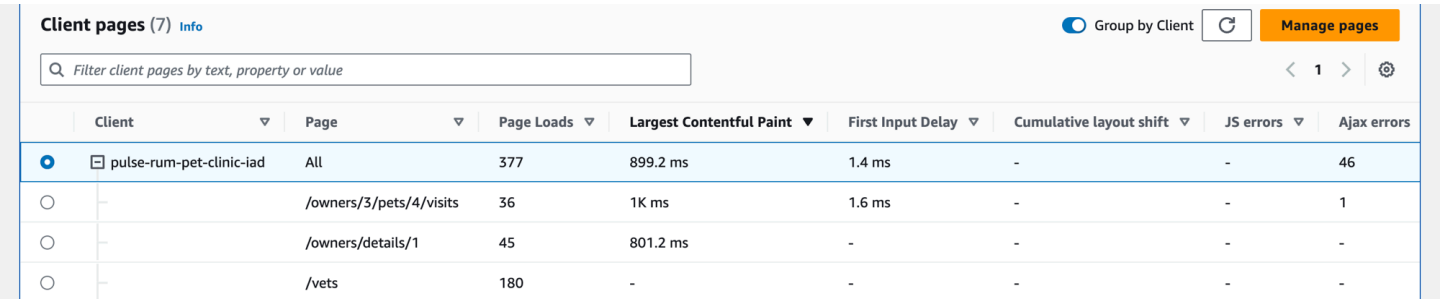
Cliquez sur le bouton radio à côté du nom du canari pour afficher un ensemble d'onglets contenant des graphiques, des mesures détaillées, notamment le pourcentage de réussite, les erreurs et la durée. Passez le pointeur de la souris sur un point d'un graphique pour afficher une fenêtre contextuelle contenant plus d'informations. Sélectionnez un point dans un graphique pour ouvrir un volet de diagnostic qui affiche les courses Canary corrélées au point sélectionné. Sélectionnez un Canary Run et choisissez la durée d'exécution pour voir les artefacts associés à votre Canary Run sélectionné, notamment les journaux, les fichiers d'HTTPArchive (HAR), les captures d'écran et les étapes suggérées pour vous aider à résoudre les problèmes. Choisissez En savoir plus pour ouvrir la page CloudWatch Synthetics Canaries à côté de Canary runs.

Consultation de vos pages clients

Choisissez l'onglet Pages clients pour afficher la liste des pages Web des clients qui font appel à votre service. Utilisez l'ensemble de mesures de la page client sélectionnée pour mesurer la qualité de l'expérience de votre client lorsqu'il interagit avec un service ou une application. Ces statistiques incluent les chargements de pages, les données vitales sur le Web et les erreurs.

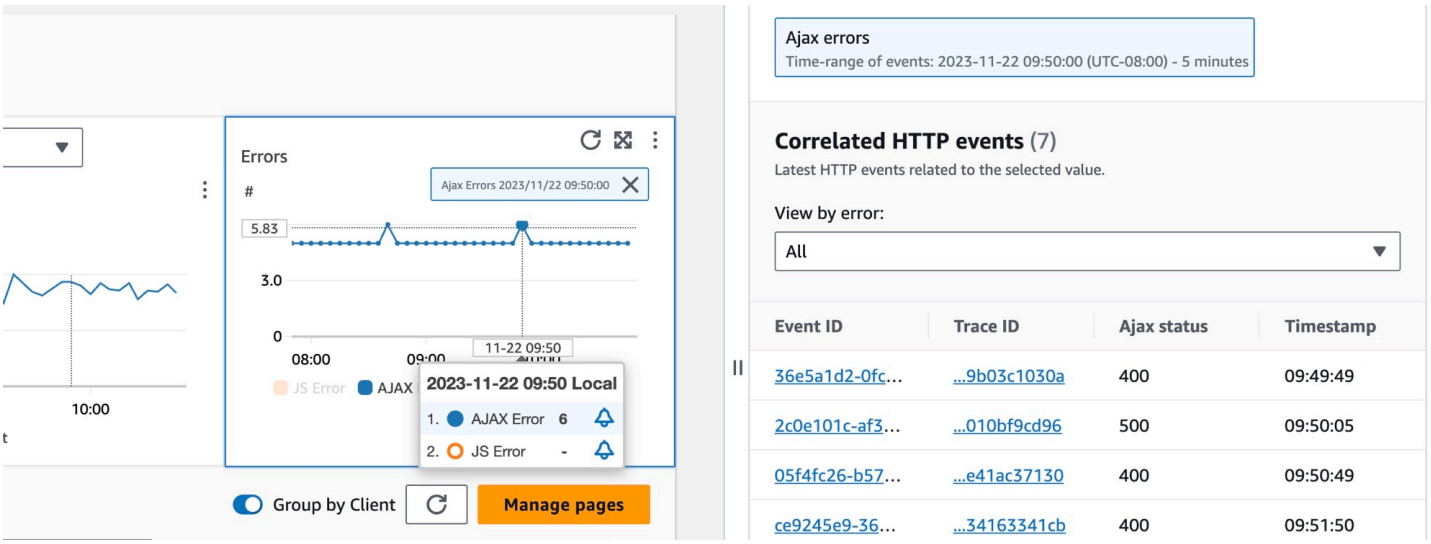
Pour afficher les pages de vos clients dans le tableau, vous devez configurer votre client Web CloudWatch RUM pour le suivi X-Ray et activer les métriques Application Signals pour vos pages clientes. Choisissez Gérer les pages pour sélectionner les pages activées pour les métriques des signaux d'application.

Utilisez la zone de texte du filtre pour trouver la page client ou le moniteur d'application qui vous intéresse sous la zone de texte du filtre. Choisissez Effacer les filtres pour supprimer le filtre du tableau. Sélectionnez Grouper par client pour regrouper les pages client par client. Lorsque vous êtes groupés, cliquez sur l'icône + à côté du nom d'un client pour développer la ligne et afficher toutes les pages relatives à ce client.



Pour sélectionner une page client, sélectionnez l'option située à côté d'une page client dans le tableau Pages client. Vous verrez un ensemble de graphiques affichant des métriques détaillées. Passez le pointeur de la souris sur un point d'un graphique pour afficher une fenêtre contextuelle contenant plus d'informations. Sélectionnez un point dans un graphique pour ouvrir un volet de diagnostic qui affiche les événements de navigation liés aux performances corrélés pour le point

sélectionné dans le graphique. Choisissez un identifiant d'événement dans la liste des événements de navigation pour ouvrir la vue de la page CloudWatch RUM pour l'événement choisi.



> **Note**
>
> Pour voir les erreurs AJAX dans vos pages client, utilisez le client Web CloudWatch RUM version 1.15 ou ultérieure.
> Actuellement, jusqu'à 100 opérations, scripts canary et pages clients, et jusqu'à 250 dépendances, peuvent être affichés par service.

Consultez la topologie de votre application et surveillez l'état de fonctionnement grâce à la carte des CloudWatch services

> **Application Signals est en version préliminaire pour Amazon CloudWatch et est susceptible d'être modifiée.**

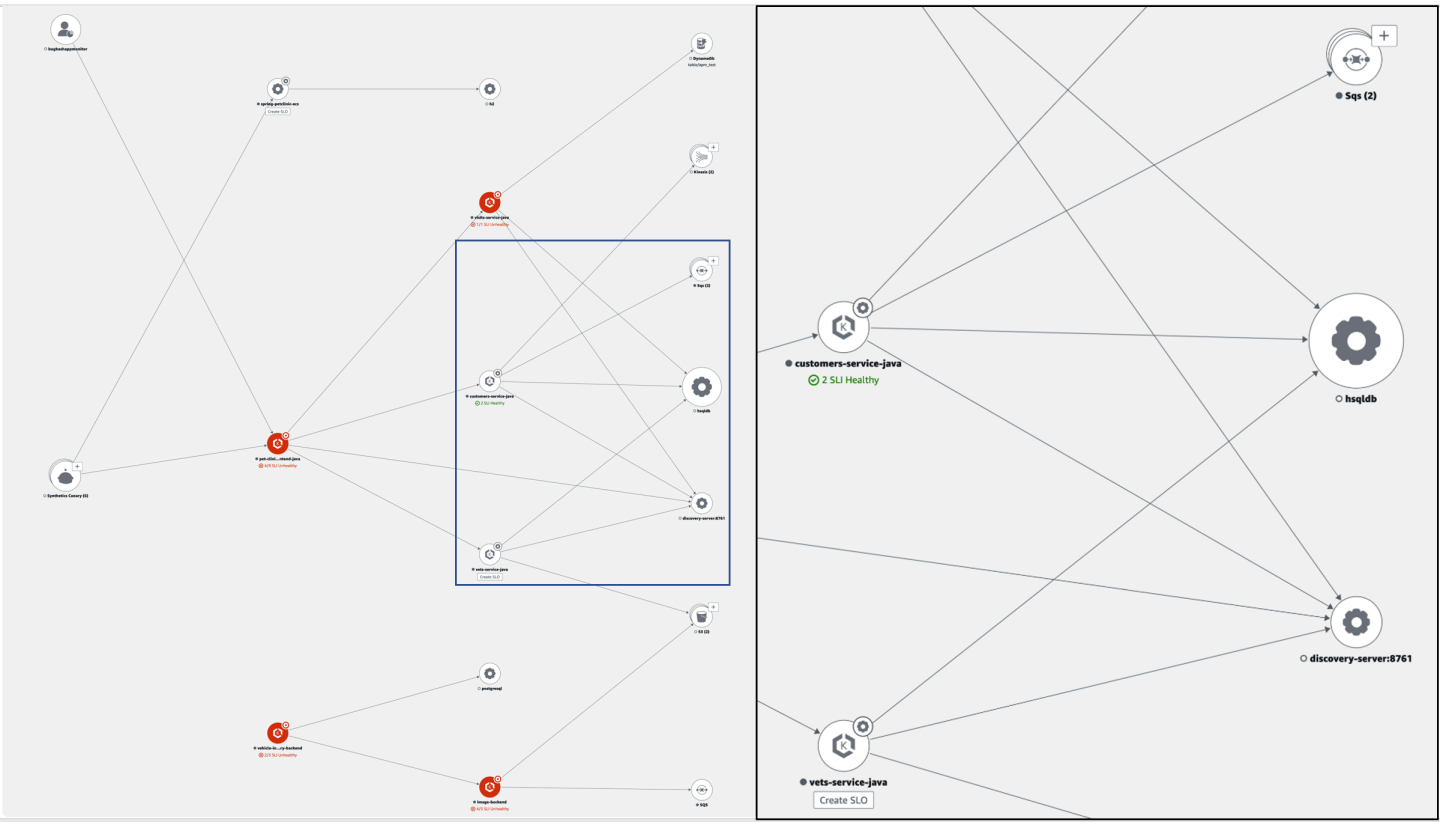
> **Note**
>
> La carte des CloudWatch services remplace la ServiceLens carte. Pour afficher une carte de votre application basée sur AWS X-Ray des traces, ouvrez la X-Ray Trace Map. Choisissez Trace Map dans la section X-Ray du volet de navigation de gauche de la CloudWatch console.

Utilisez la carte des services pour visualiser la topologie de vos clients d'applications, de vos canaris synthétiques, de vos services et de vos dépendances, et pour surveiller l'état de fonctionnement. Pour afficher la carte des services, ouvrez la CloudWatch console et choisissez Service Map dans la section Application Signals dans le volet de navigation de gauche.

Après avoir activé votre application pour Application Signals, utilisez la carte des services pour faciliter le suivi de l'état de fonctionnement de votre application :

- Visualisez les connexions entre les nœuds de client, de script canary, de service et de dépendance pour vous aider à comprendre la topologie et le flux d'exécution de votre application. Cela est particulièrement utile si vos opérateurs de services ne font pas partie de votre équipe de développement.
- Découvrez quels services répondent ou non à vos objectifs de niveau de service (SLO). Lorsqu'un service ne respecte pas vos SLO, vous pouvez rapidement identifier si un service en aval ou une dépendance peut contribuer au problème ou avoir un impact sur plusieurs services en amont.
- Sélectionnez un client individuel, Synthetics Canary, un service ou un nœud de dépendance pour voir les métriques associées. La page des détails du service contient des informations plus détaillées sur les opérations, les dépendances, les canaris synthétiques et les pages clients.
- Filtrez et zoomez sur la carte des services afin de vous concentrer plus facilement sur une partie de la topologie de votre application ou de visualiser la carte dans son intégralité. Créez un filtre en choisissant une ou plusieurs propriétés dans la zone de texte du filtre. Au fur et à mesure que vous choisissez chaque propriété, vous êtes guidé à travers les critères de filtrage. Vous verrez le filtre complet sous la zone de texte du filtre. Choisissez Effacer les filtres à tout moment pour supprimer le filtre.

L'exemple de carte des services suivant montre des services dont les arêtes les relient aux composants avec lesquels ils interagissent. Si un SLO est défini, la carte des services indique également l'état de santé.

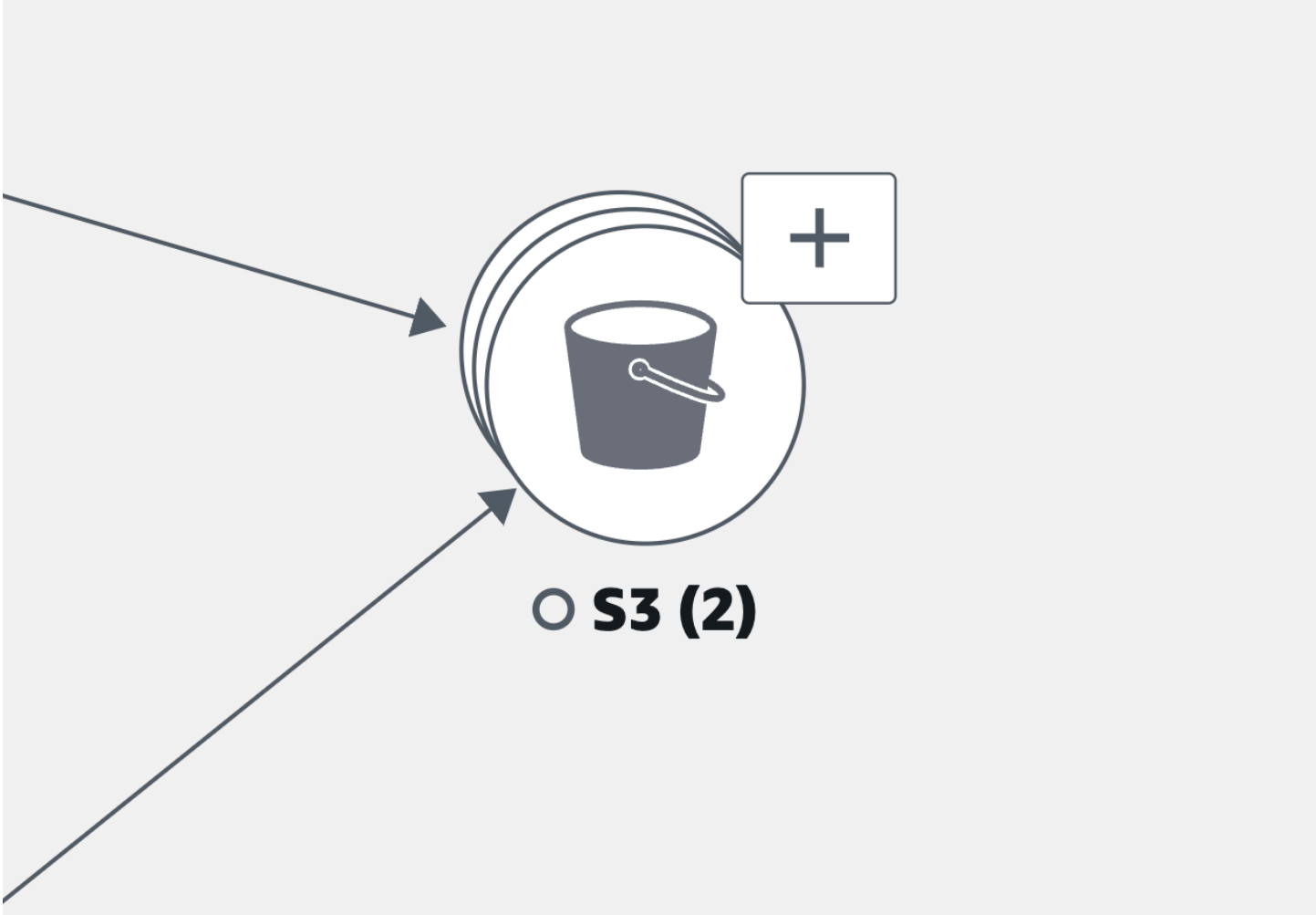


Explorez la carte des services

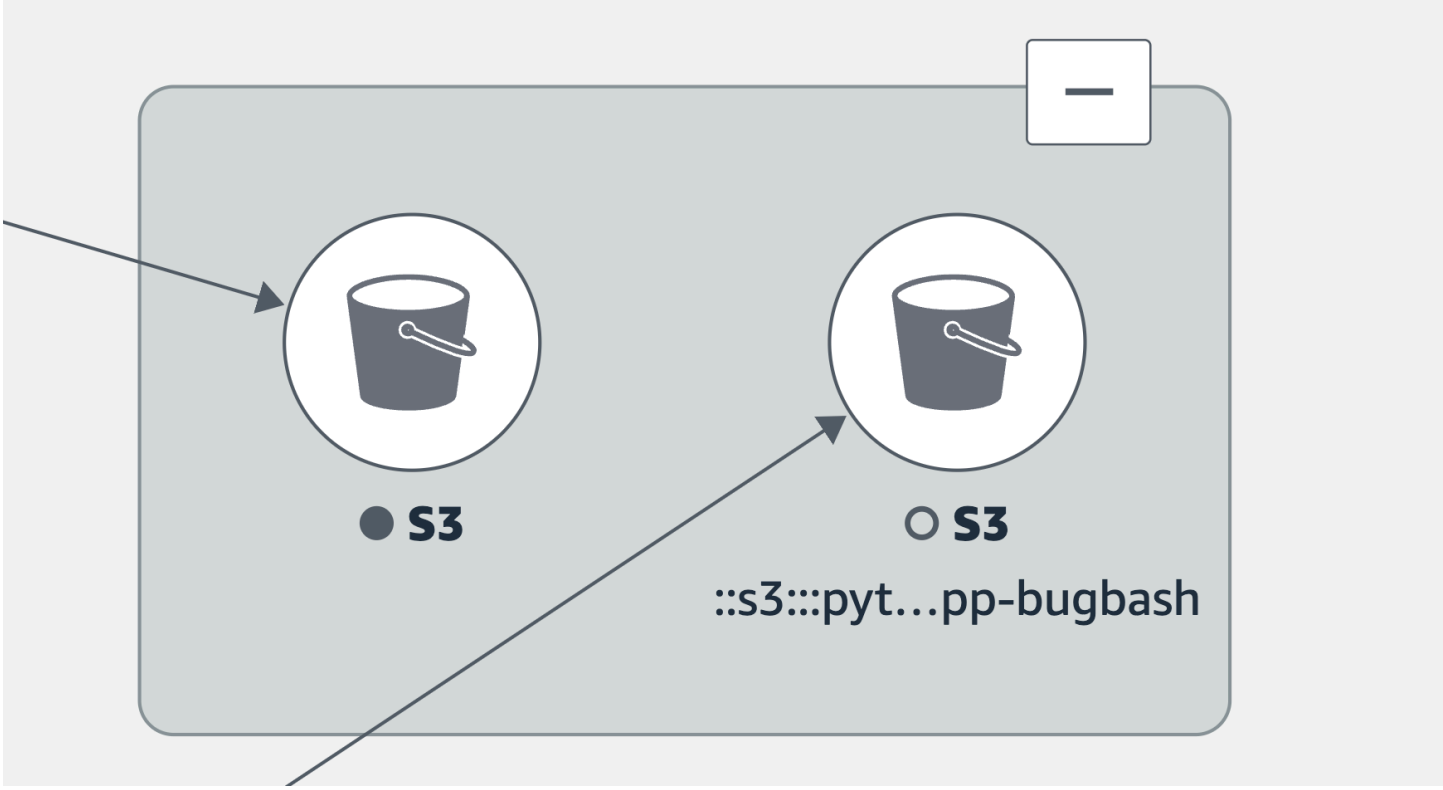
Une fois que vous avez activé les signaux d'application dans votre application, la carte des services affiche les nœuds représentant vos services et leurs dépendances.

Activez le suivi actif pour vos clients CloudWatch RUM et Synthetics Canaries afin de voir les nœuds clients et Canary sur la carte.

Par défaut, les canaris, les clients RUM et les dépendances de AWS service du même type sont regroupés dans une seule icône extensible dans la carte des services. Les dépendances de service extérieures à ne AWS sont pas regroupées par défaut. Par exemple, dans l'image suivante, tous les compartiments Amazon S3 sont regroupés sous une seule icône extensible :



Dans l'image précédente, l'étiquette située entre le regroupement Amazon S3 et le service d'origine indique le nombre d'arêtes du groupe entre parenthèses sous l'icône de la dépendance. Sélectionnez l'icône (+) pour développer le groupe et voir ses différents éléments, comme illustré dans l'image suivante :

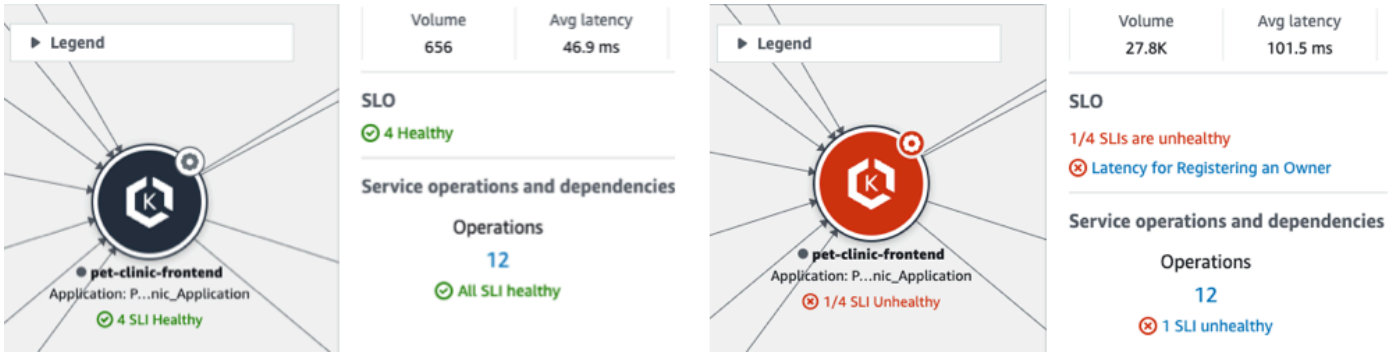


Choisissez un onglet pour obtenir des informations sur l'exploration de chaque type de nœud et des arêtes (connexions) entre eux.

View your application services

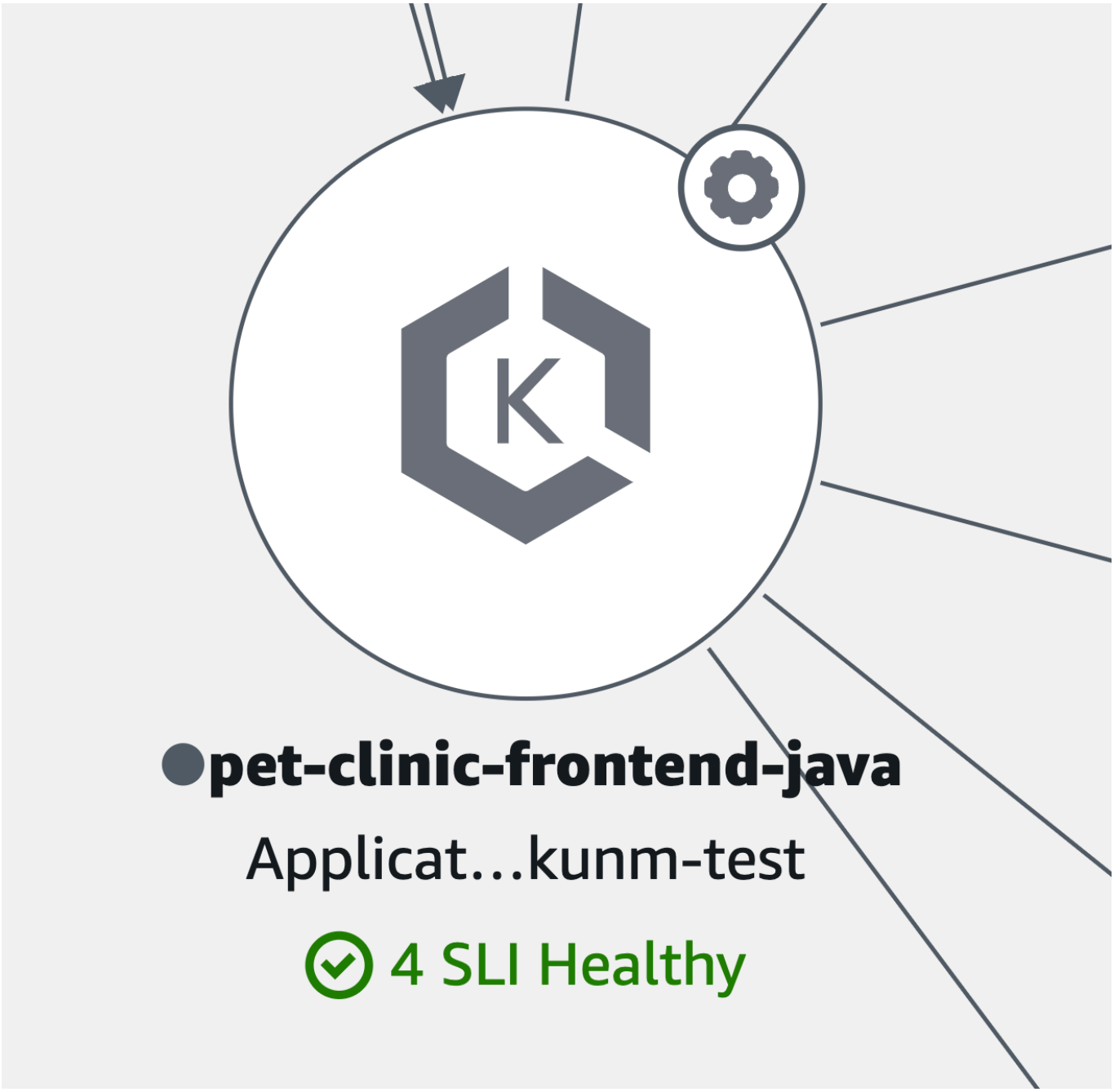
Vous pouvez consulter les services de votre application ainsi que l'état de leurs SLO et indicateurs de niveau de service (SLI) dans la carte des services. Si vous n'avez pas créé de SLO pour un service, cliquez sur le bouton Créer un SLO situé sous le nœud de service.

La carte des services affiche tous vos services. Il indique également les clients et les canaris qui consomment le service et les dépendances que vos services appellent, comme le montre l'image suivante :



Les icônes suivantes représentent des exemples de services applicatifs dans la carte des services :

- Amazon Elastic Kubernetes Service :



- Un conteneur Kubernetes :

- Amazon Elastic Compute Cloud (Amazon EC2) :

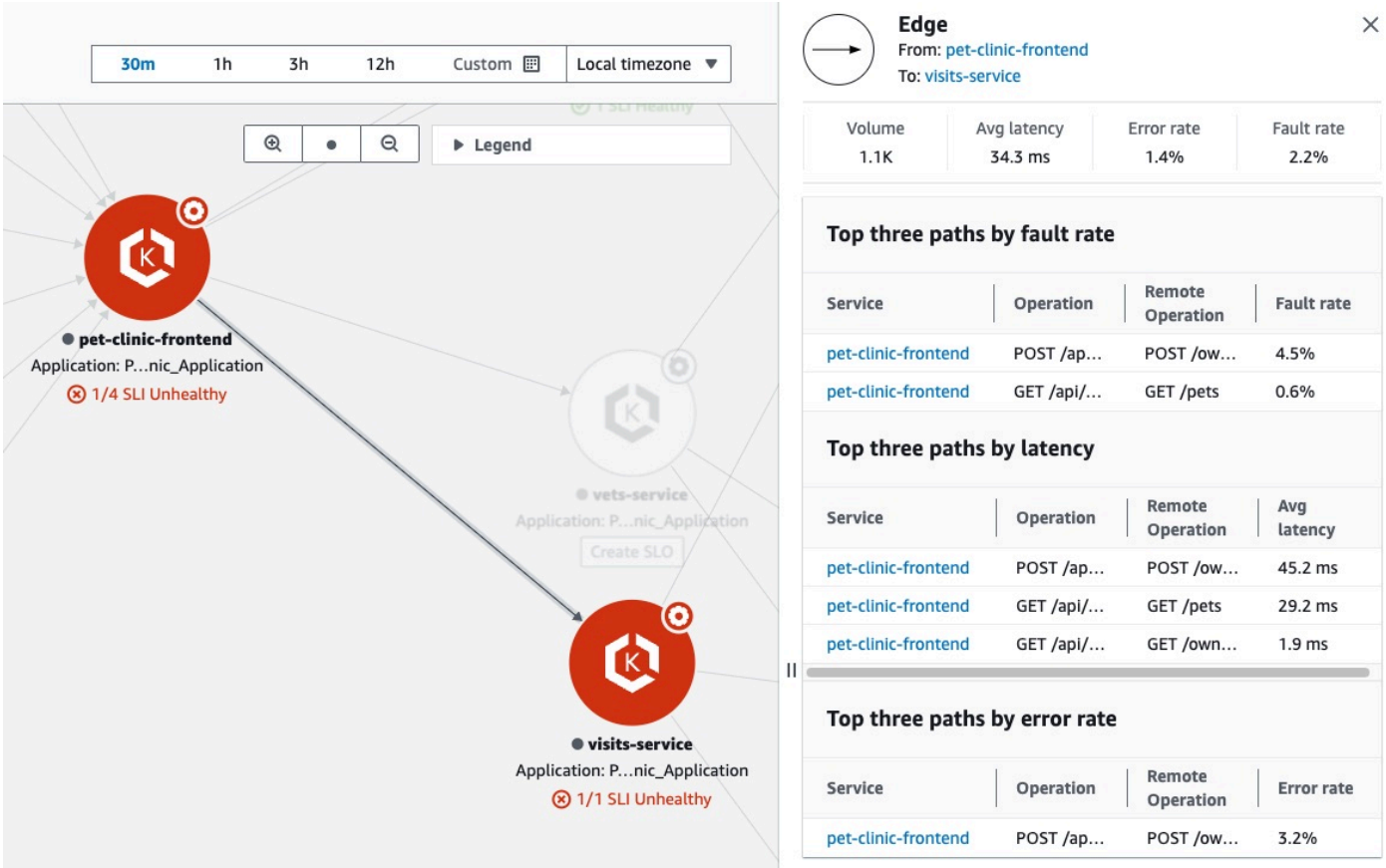
- Autres types de services d'application non répertoriés précédemment :

Lorsque vous sélectionnez un nœud de service, un volet s'ouvre et affiche des informations détaillées sur le service :

- Métriques relatives au volume d'appels, à la latence, aux erreurs et au taux d'erreur.
- Le nombre de SLI et de SLO qui sont `healthy` ou. `unhealthy`
- L'option permettant d'afficher plus d'informations sur un SLO.
- Le nombre d'opérations de service, de dépendances, de canaris synthétiques et de pages clients.
- L'option permettant de sélectionner chaque numéro pour ouvrir la page des détails du service correspondant.

- Le nom de l'application, si vous avez associé la ressource de calcul sous-jacente à une application à l'aide AppRegistry de la carte Applications sur la page d' AWS Management Console accueil.
  - Choisissez le nom de l’application pour afficher les détails de l’application sur la page de console myApplications.
- Le `ClusterNamespace`, et `Workload` pour les services hébergés sur Amazon EKS, ou `Environment` pour les services hébergés sur Amazon ECS ou Amazon EC2. Pour les services hébergés par Amazon EKS, choisissez n'importe quel lien pour ouvrir CloudWatch Container Insights.

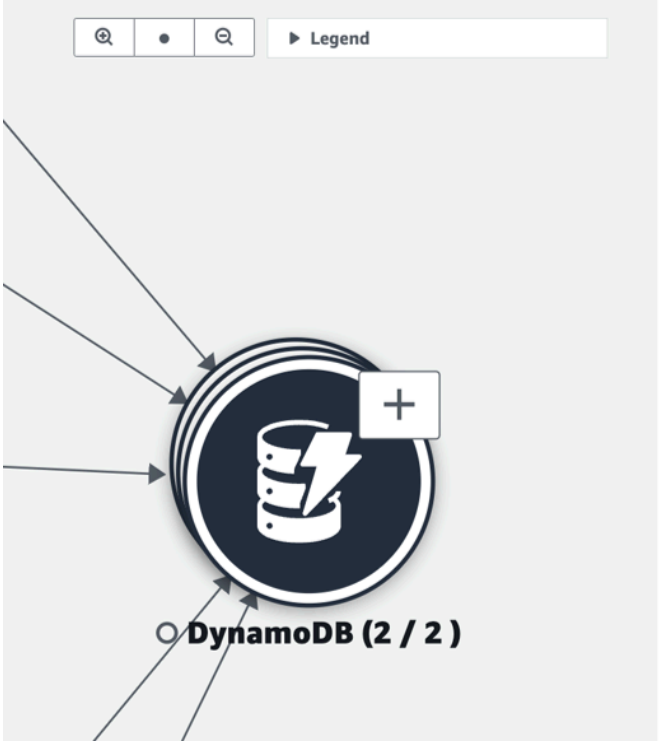
Sélectionnez un périphérique ou une connexion entre un nœud de service et un nœud de service ou de dépendance en aval. Cela ouvre un volet contenant les meilleurs chemins par taux d'erreur, latence et taux d'erreur, comme indiqué dans l'exemple d'image suivant. Choisissez n'importe quel lien dans le volet pour ouvrir la page des détails du service et consulter les informations détaillées du service ou de la dépendance choisi.



View dependencies

Les dépendances de vos applications sont affichées sur la carte des services, connectées aux services qui les appellent.

Choisissez un nœud de dépendance pour ouvrir un volet contenant les meilleurs chemins par taux de défaillance, latence et taux d'erreur. Choisissez un lien de service ou de cible pour ouvrir la page Détails du service et consulter des informations détaillées sur le service ou la cible de dépendance choisi, comme indiqué dans l'exemple d'image ci-dessous :



Volume	Avg latency	Error rate	Fault rate
-	-	-	-

Top three paths by fault rate

Service	Remote operation	Fault rate
No paths with faults		

Top three paths by latency

Service	Remote operation	Avg latency
billing-service-ec2-python	PutItem	282.8 ms
billing-service-python	PutItem	75.6 ms
visits-service-java	PutItem	64.9 ms

Top three paths by error rate

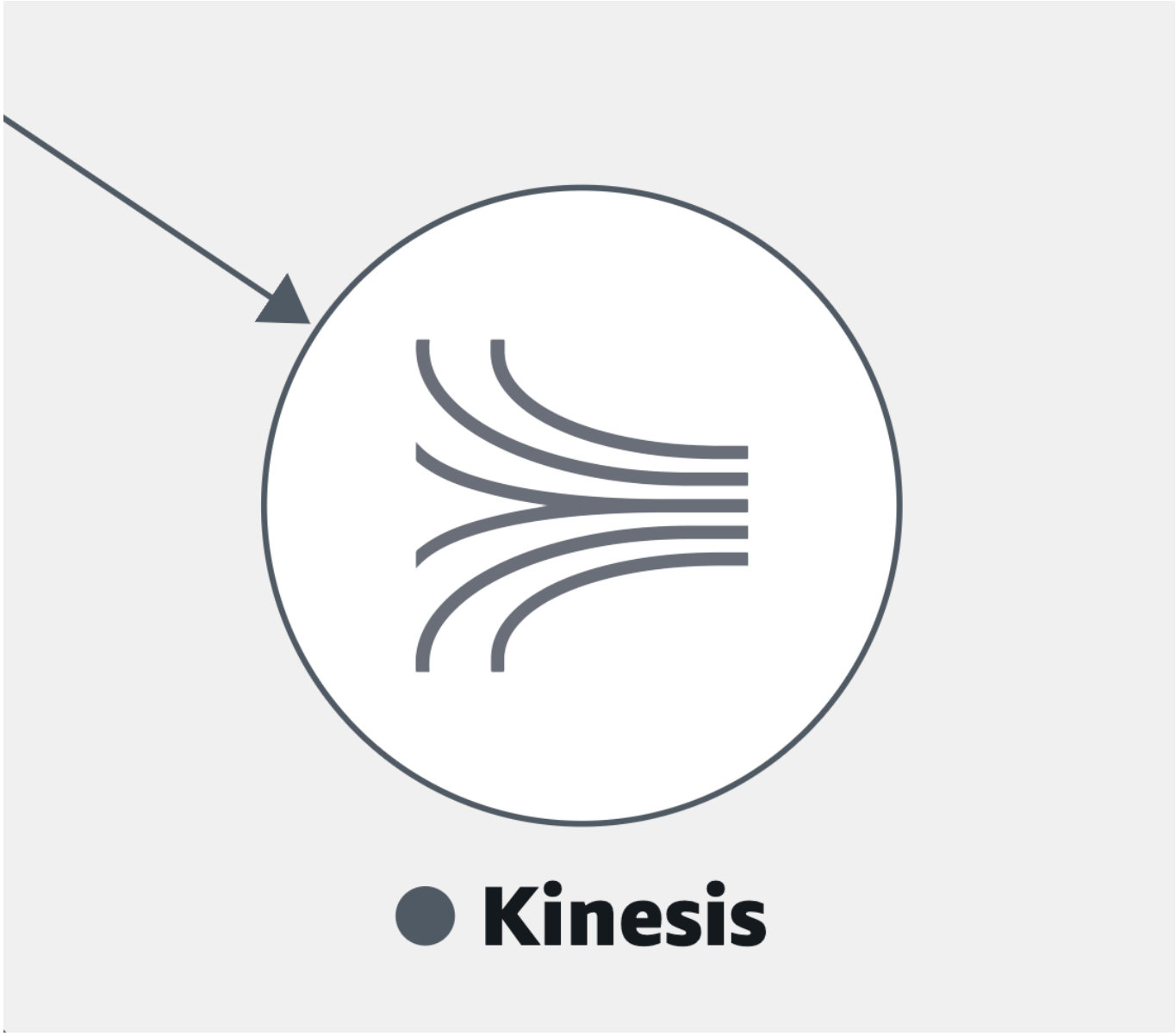
Service	Remote operation	Error rate
visits-service-java	PutItem	9.6%

Les dépendances des services sont regroupées par défaut dans une seule icône extensible. Sélectionnez l'icône (+), comme indiqué dans l'image précédente, pour développer le groupe et voir ses différents éléments.

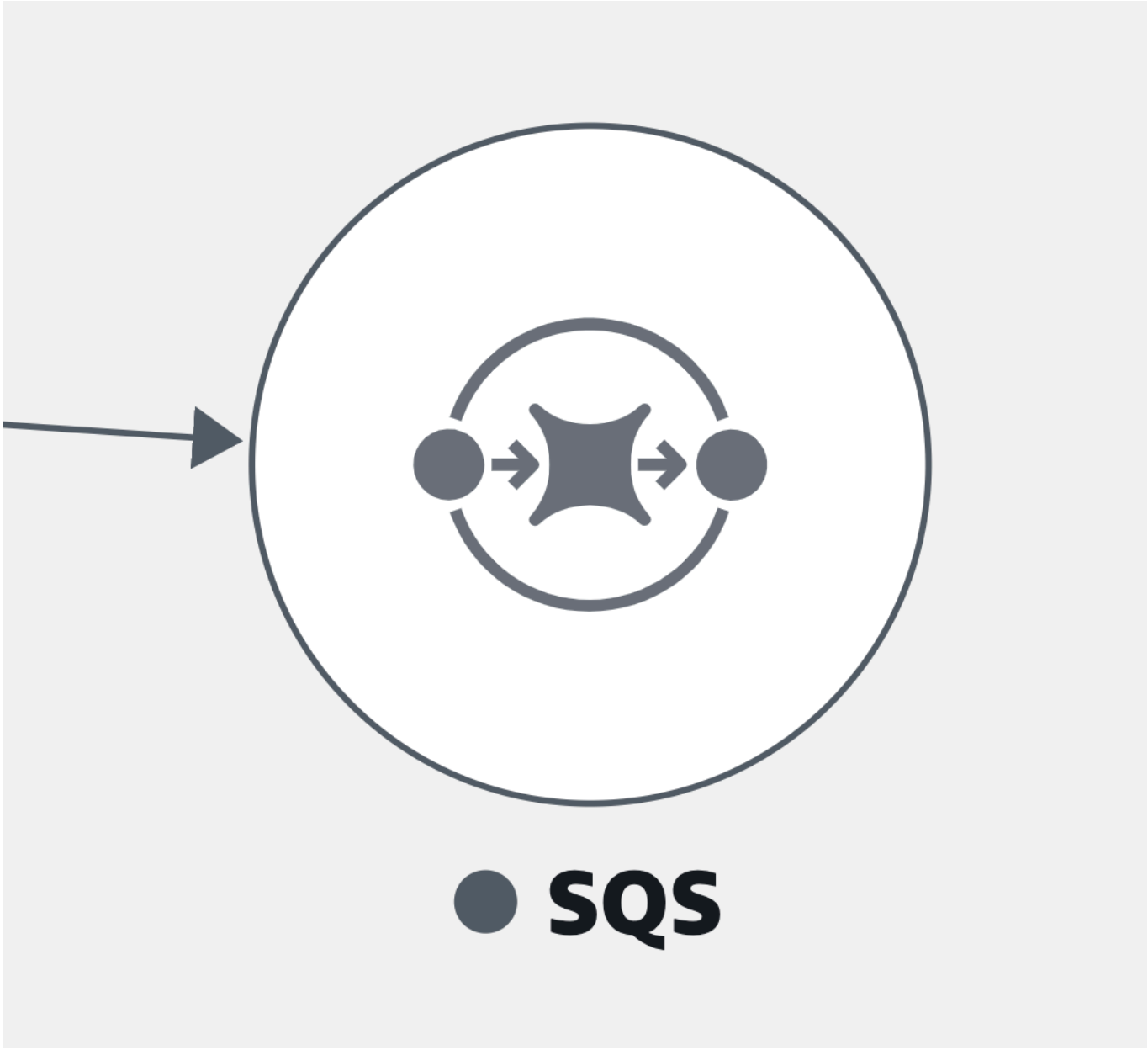
Les icônes suivantes représentent des exemples de nœuds de dépendance dans la carte des services :

- Un compartiment Amazon S3 :

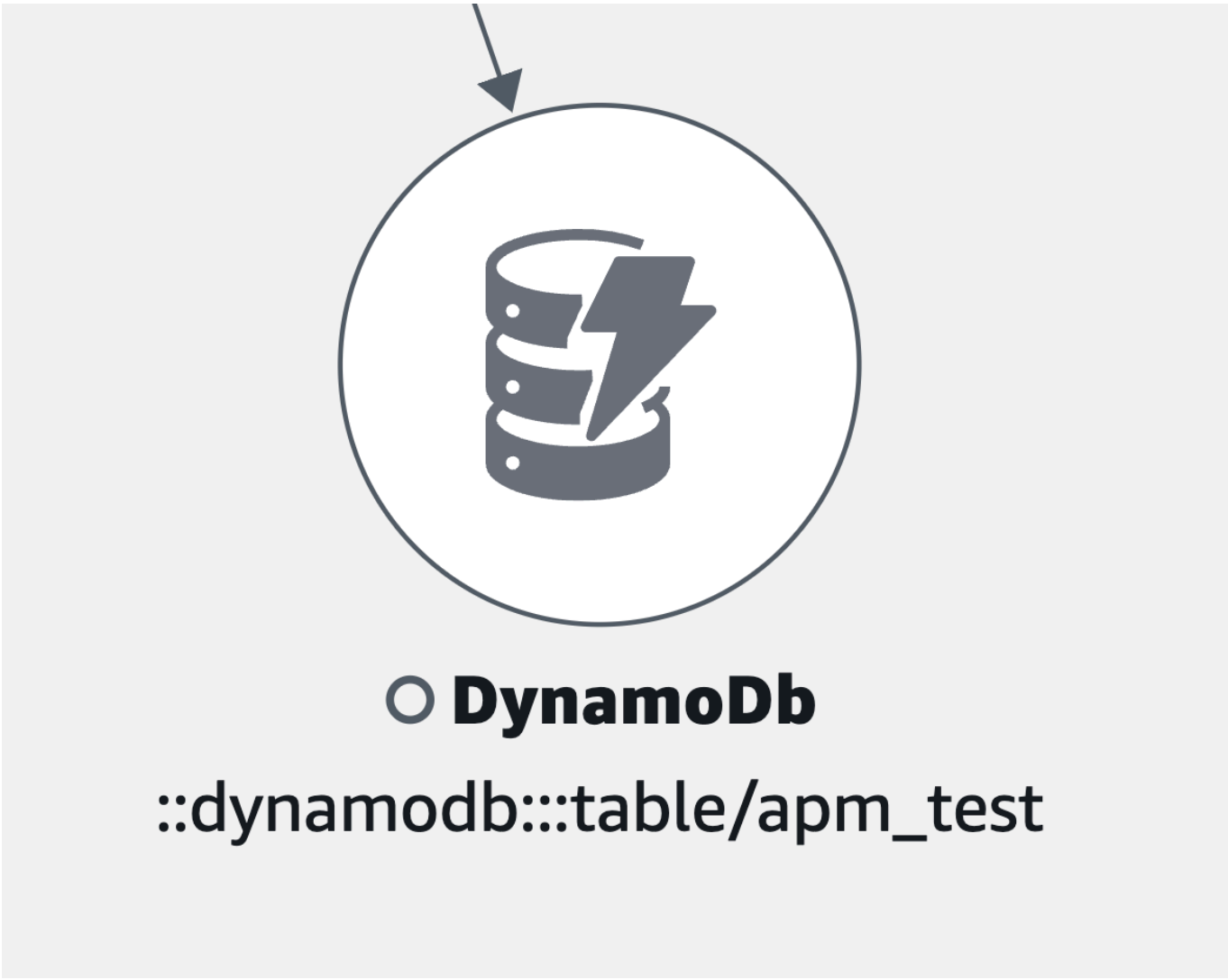
- Un stream Amazon Kinesis :



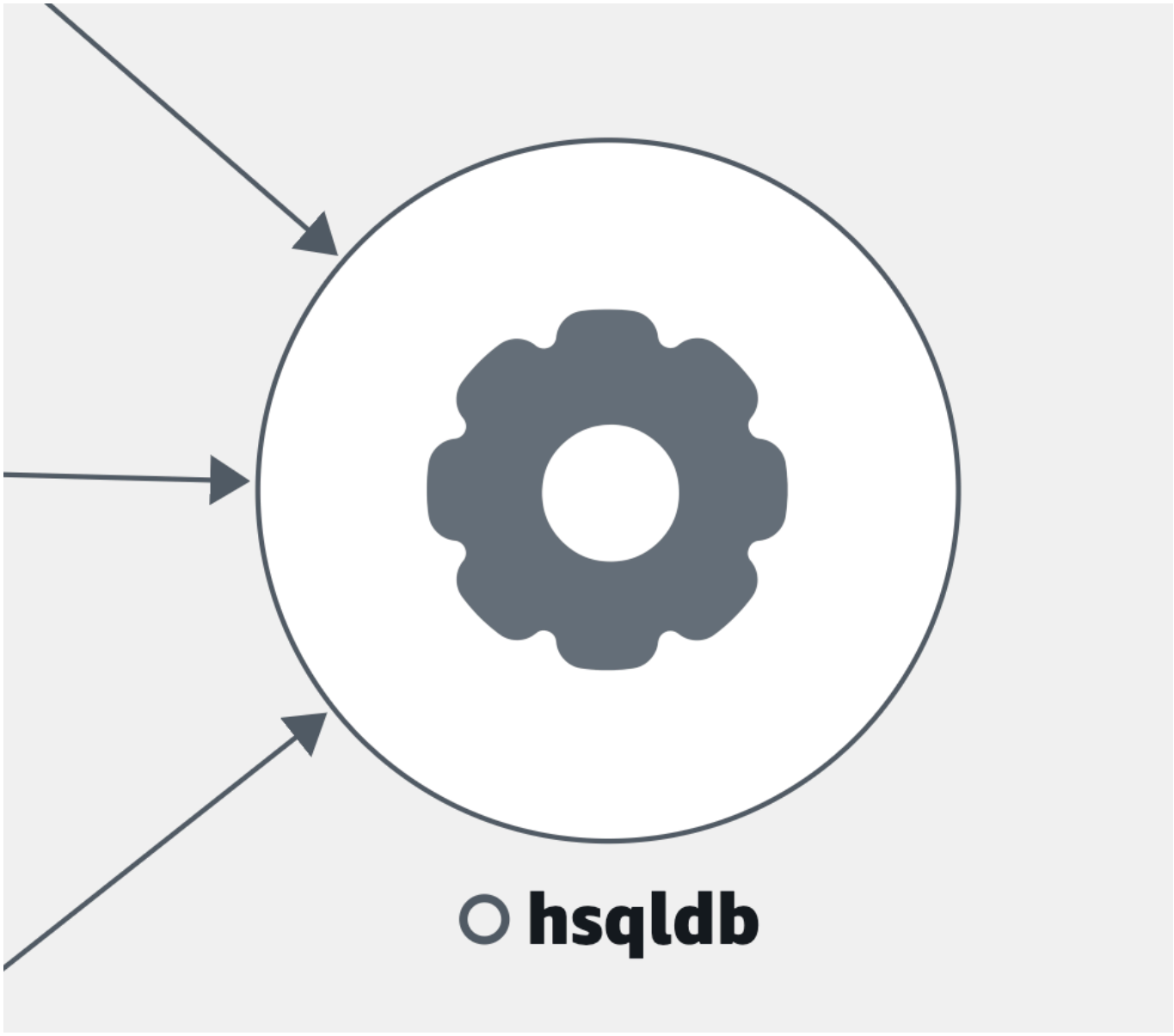
- Amazon Simple Queue Service (Amazon SQS) :



- Une table Amazon DynamoDB :



- Autres types de dépendances non répertoriés précédemment :



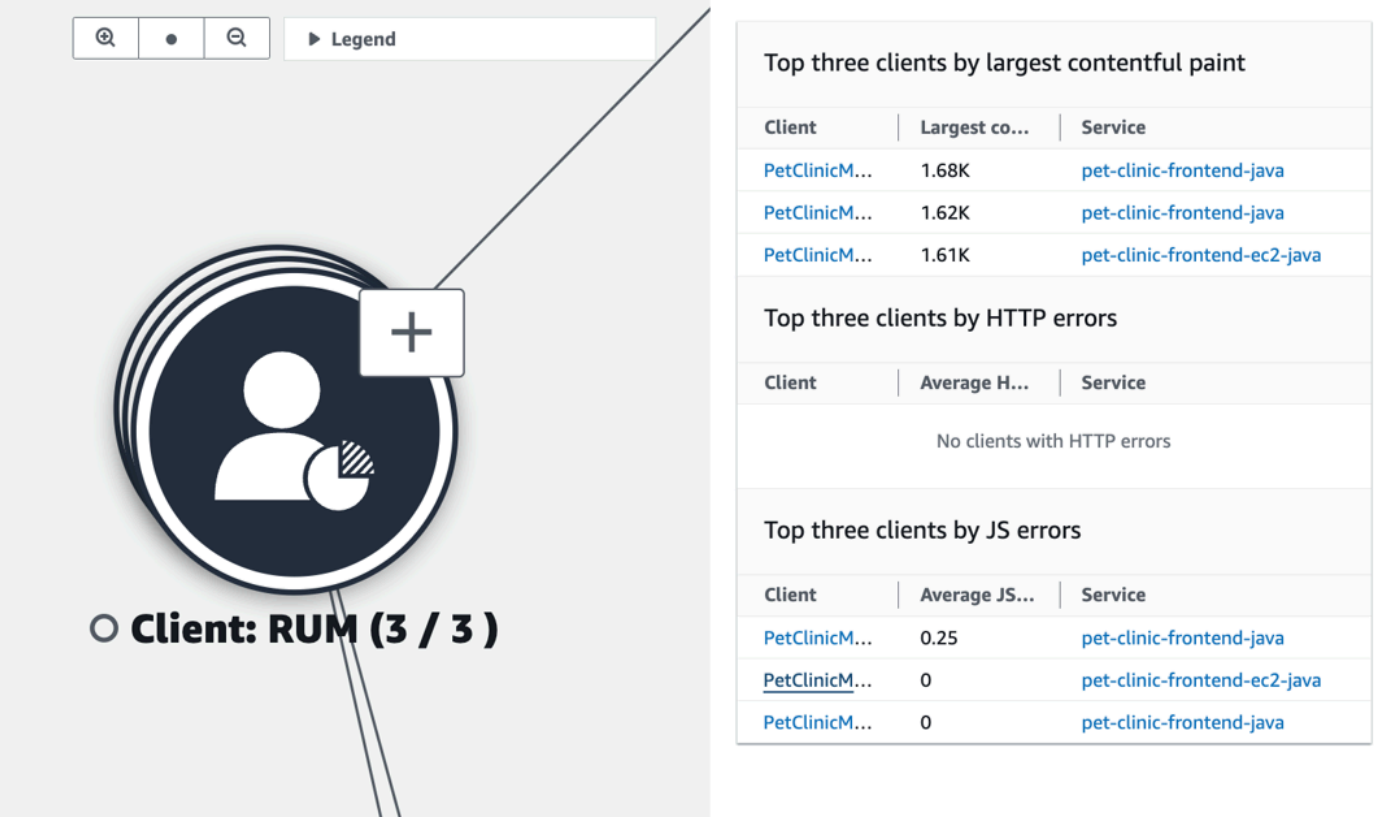
View clients

Une fois que vous avez activé le suivi X-Ray pour vos clients Web CloudWatch RUM, ils s'affichent sur la carte des services connectés aux services qu'ils appellent.

Choisissez un nœud client pour ouvrir un volet affichant des informations détaillées sur le client :

- Métriques relatives au chargement des pages, au temps de chargement moyen, aux erreurs et aux données vitales moyennes sur le Web.
- Un graphique présentant la répartition des erreurs.
- Un lien pour afficher les détails du client dans CloudWatch RUM.

Les clients RUM sont regroupés par défaut dans une seule icône extensible. Sélectionnez l'icône (+), comme indiqué dans l'image suivante, pour développer le groupe et voir ses différents éléments.



L'icône suivante représente un exemple de client RUM dans la carte des services :

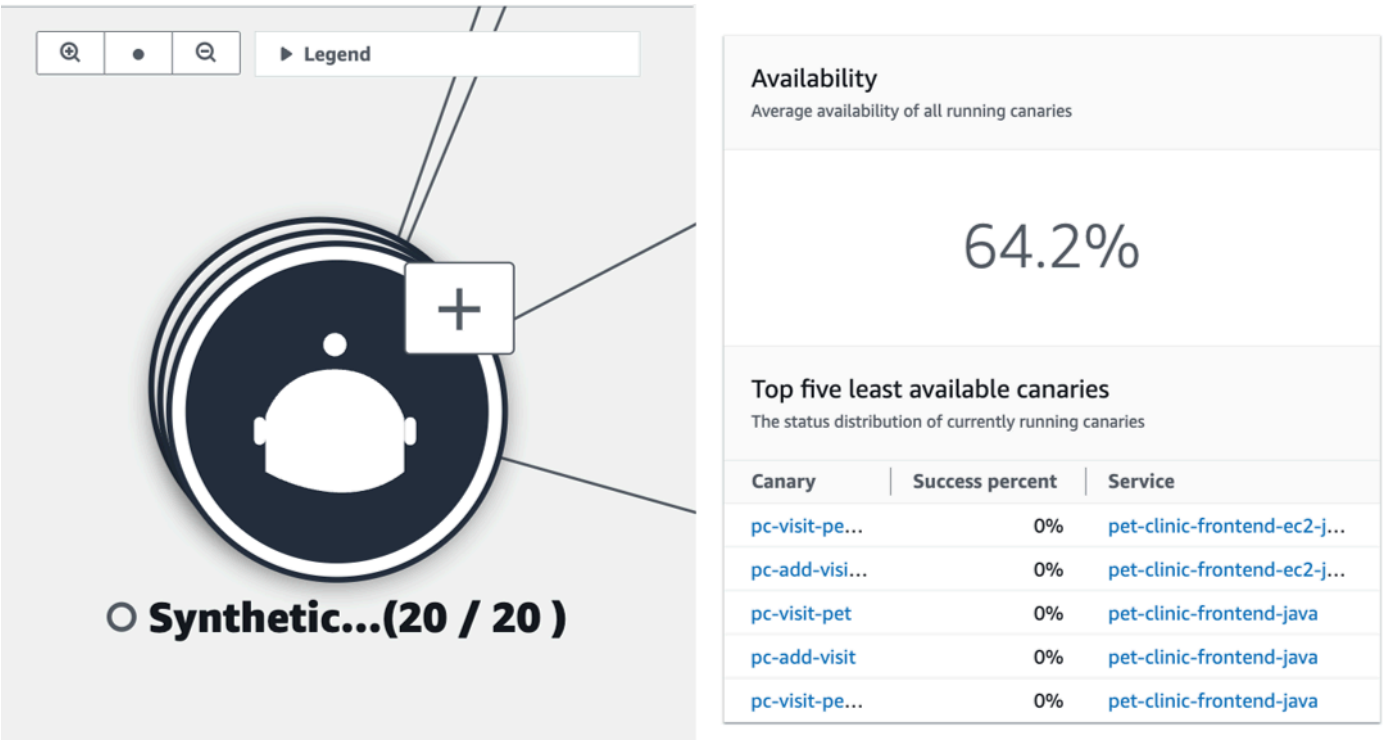
- Un client de RUM —

> **Note**
>
> Pour voir les erreurs AJAX dans vos pages client, utilisez le client Web CloudWatch RUM version 1.15 ou ultérieure.

View synthetics canaries

Une fois que vous avez activé le AWS X-Ray traçage pour vos CloudWatch canaris Synthetics, ils apparaissent sur la carte des services associés aux services qu'ils appellent, comme le montre l'exemple d'image suivant :

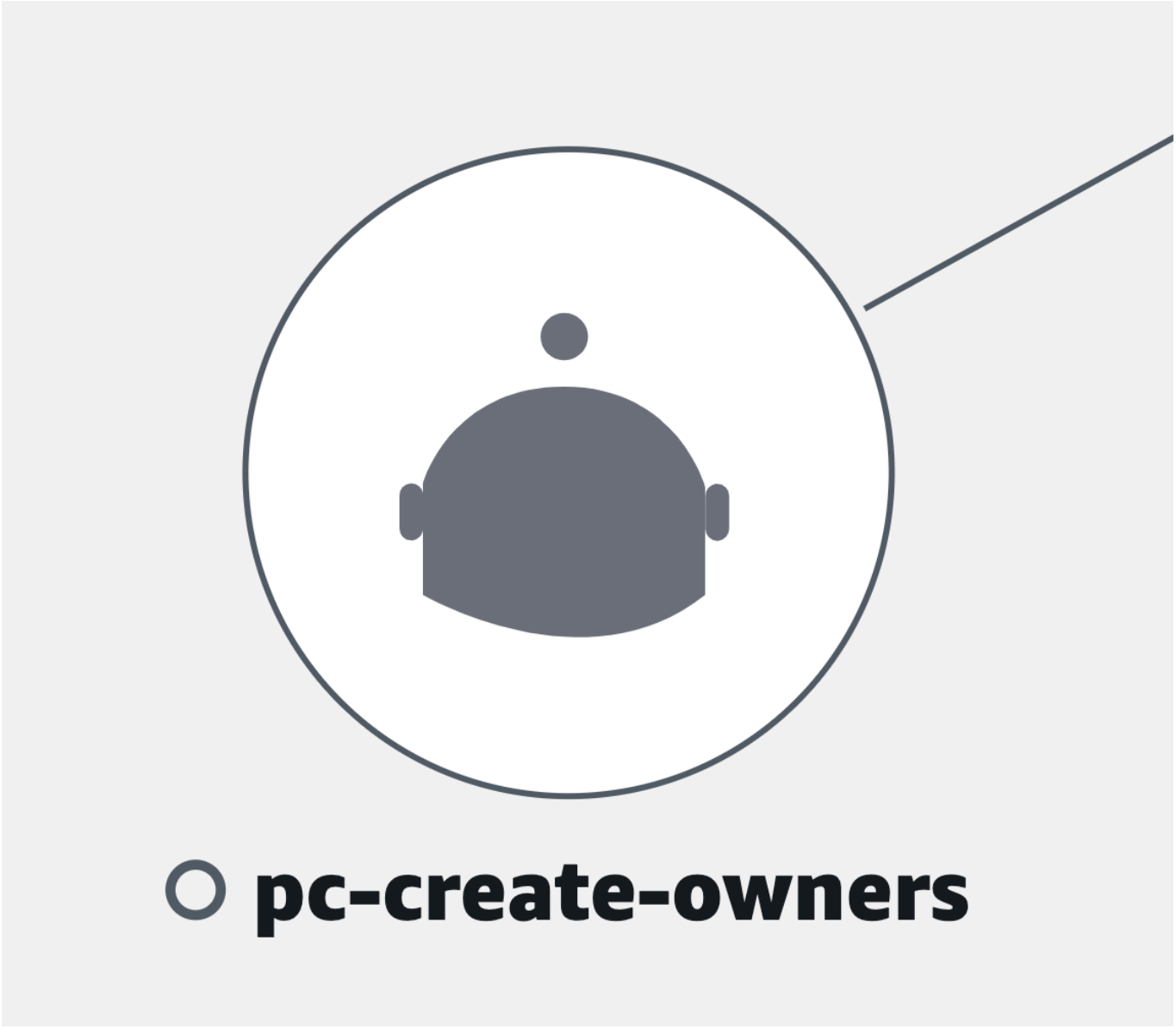
Choisissez un nœud Canary pour ouvrir un volet affichant des informations détaillées sur Canary, comme illustré dans l'image suivante :

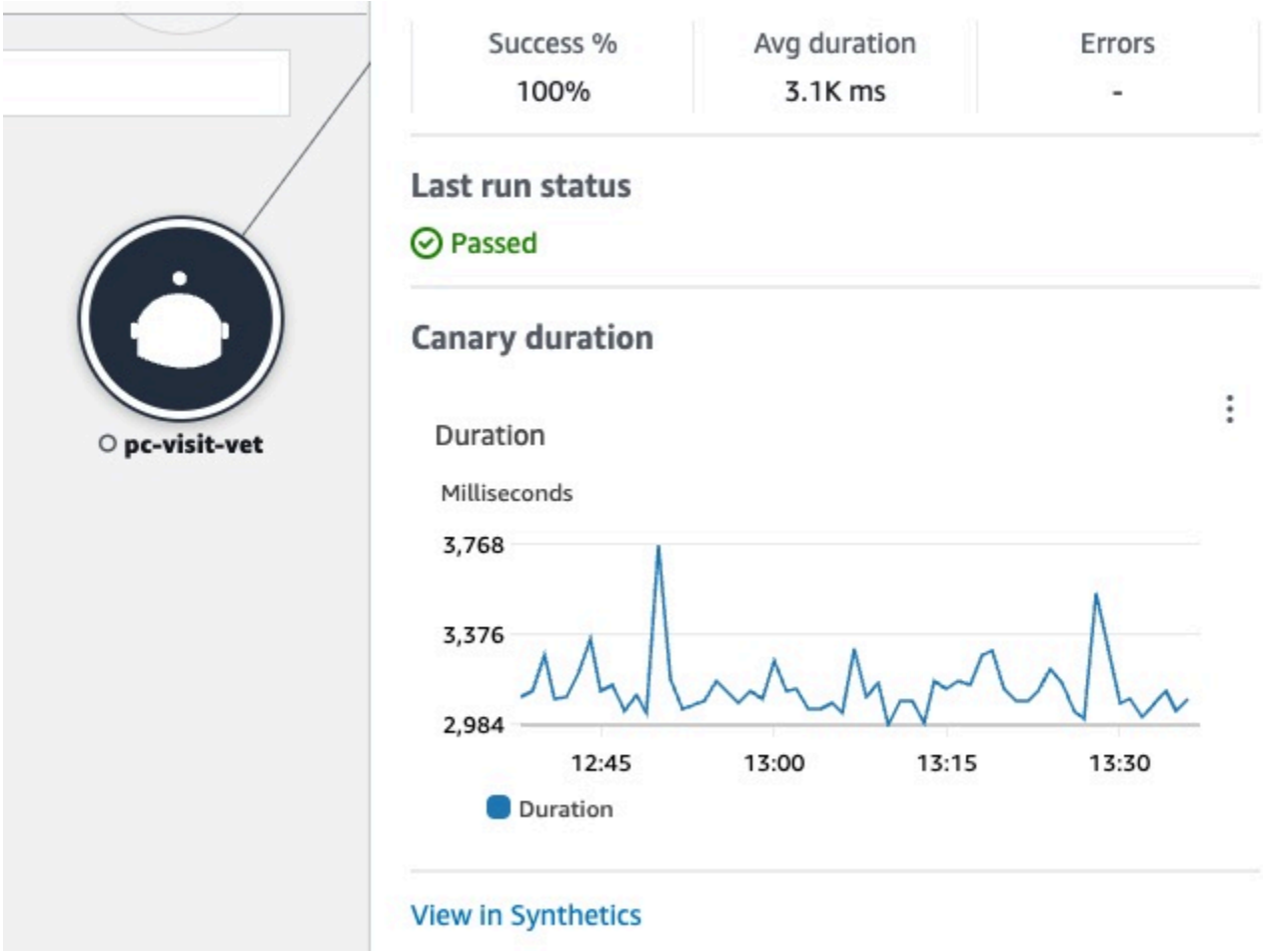


Les canaris sont regroupés par défaut dans une seule icône extensible. Sélectionnez l'icône (+), comme indiqué dans l'image précédente, pour développer le groupe et voir ses différents éléments.

Les icônes suivantes représentent des exemples de clients dans la carte des services :

- Un canari synthétique —


pc-create-owners



Dans le volet dédié aux nœuds Canary, vous pouvez voir ce qui suit :

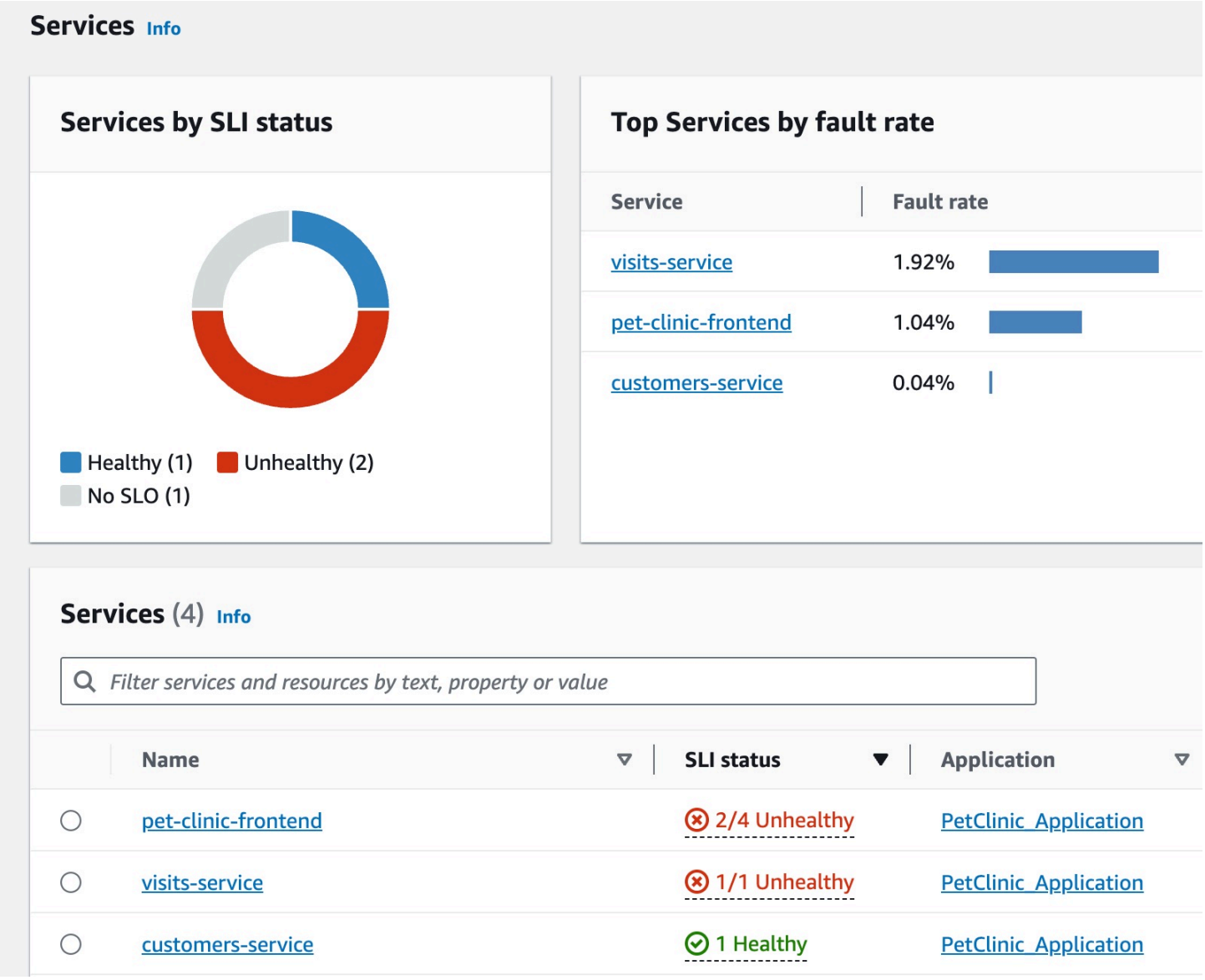
- Métriques relatives au pourcentage de réussite, à la durée moyenne et aux erreurs.
- Statut d’informations sur la dernière exécution canary.
- Un graphique indiquant la durée d’exécution canary. Passez le pointeur de la souris sur une série de graphiques pour afficher une fenêtre contextuelle contenant plus d'informations.
- Un lien pour afficher les détails du canari dans CloudWatch Synthetics.

Exemple : utilisation d’Application Signals pour résoudre un problème d’état de fonctionnement

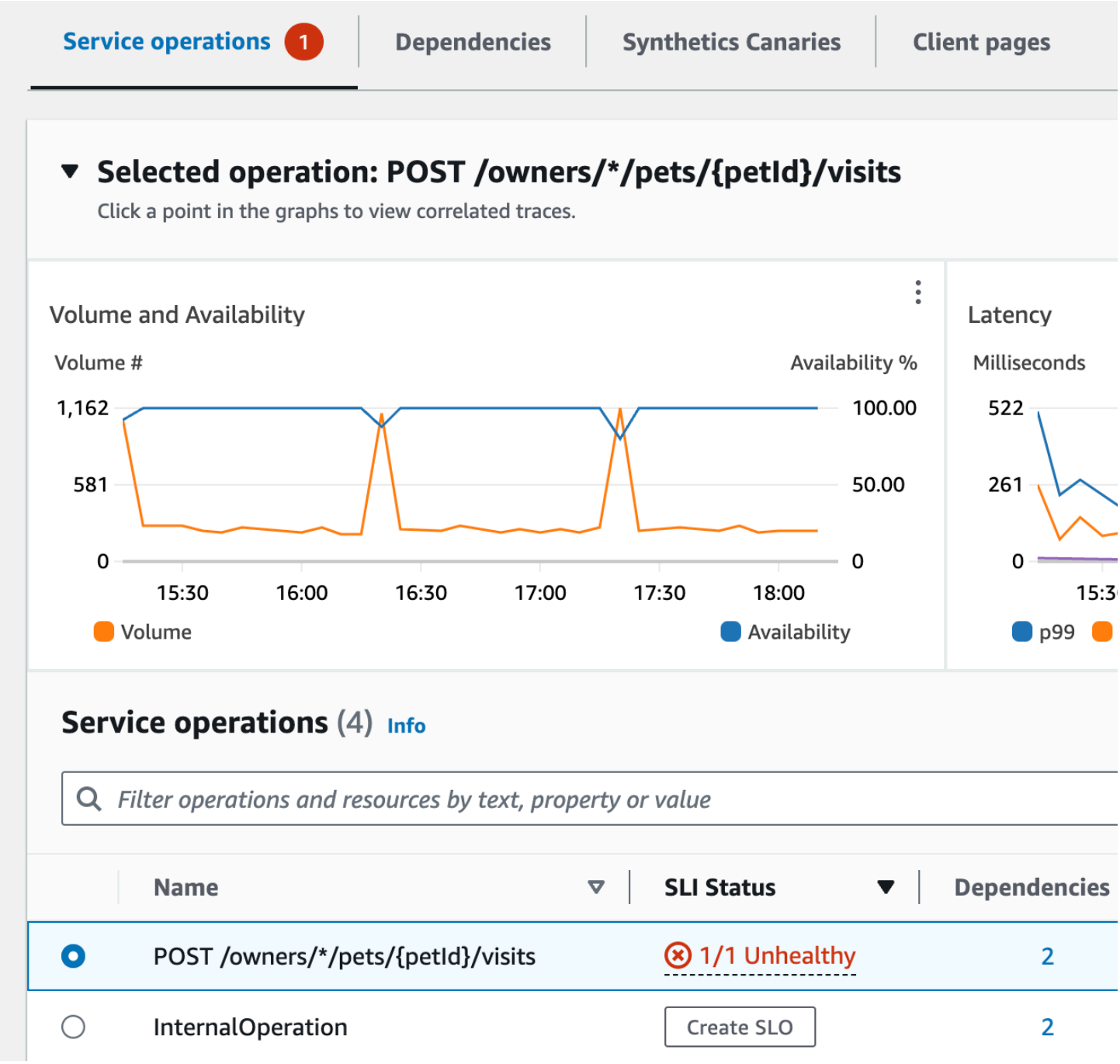
⚠ Application Signals est en version préliminaire pour Amazon CloudWatch et est susceptible d'être modifiée.

Le scénario suivant fournit un exemple de la manière dont Application Signals peuvent être utilisés pour surveiller vos services et identifier les problèmes de qualité de service. Effectuez une analyse approfondie pour identifier les causes profondes potentielles et prendre les mesures nécessaires pour résoudre le problème. Cet exemple se concentre sur une application de clinique pour animaux de compagnie composée de plusieurs microservices qui font appel, par Services AWS exemple, à DynamoDB.

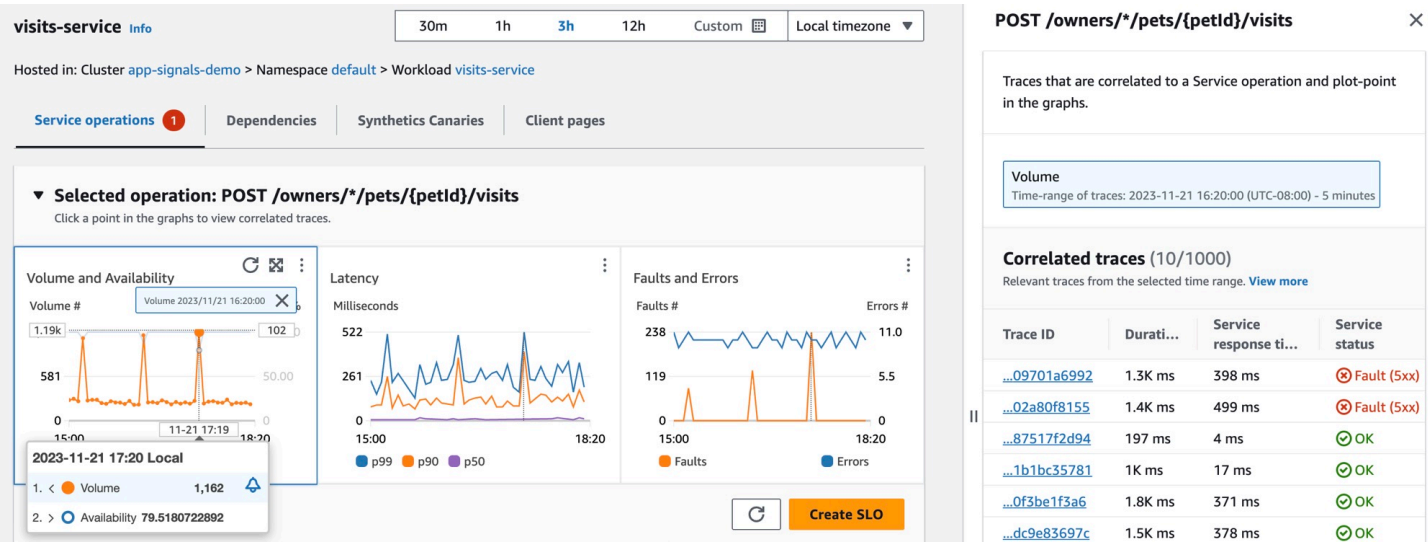
Jane fait partie d'une DevOps équipe qui supervise la santé opérationnelle d'une application de clinique pour animaux de compagnie. L'équipe de Jane s'engage à faire en sorte que l'application soit hautement disponible et réactive. Ils utilisent des objectifs de niveau de service (SLO) pour mesurer les performances des applications par rapport à ces engagements métier. Elle reçoit une alerte concernant plusieurs indicateurs de niveau de service (SLI) non sains. Elle ouvre la CloudWatch console et accède à la page Services, où elle constate que plusieurs services ne fonctionnent pas correctement.



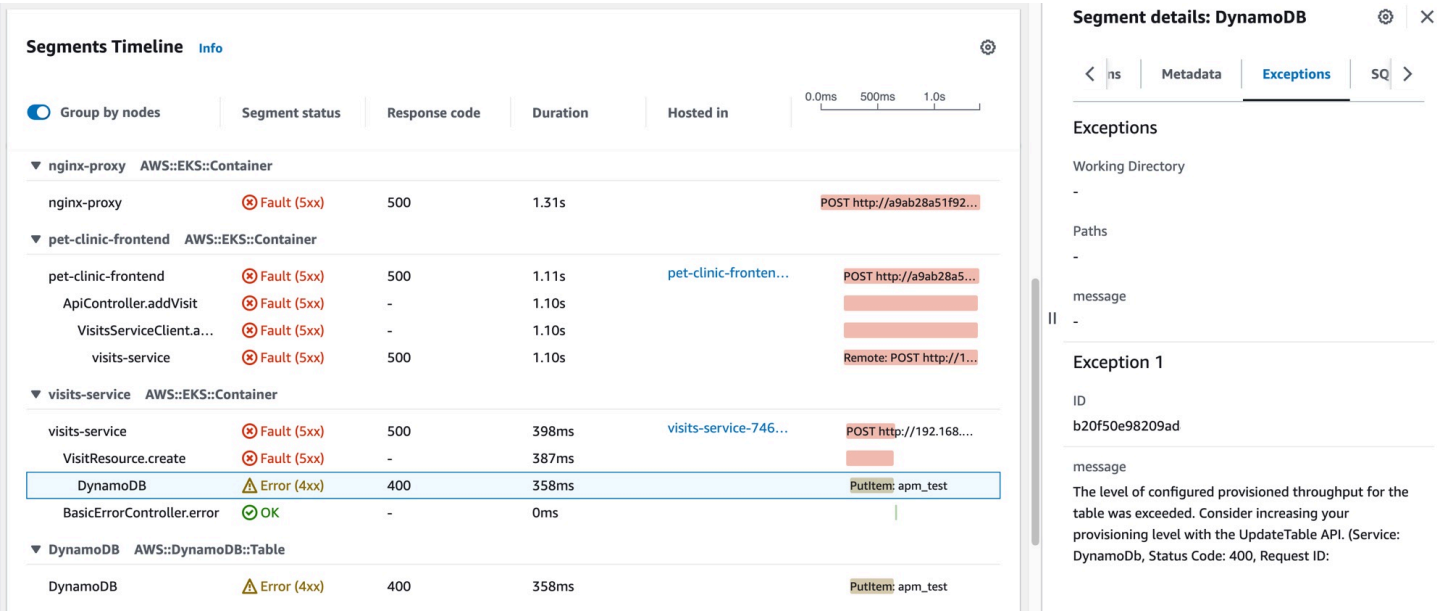
En haut de la page, Jane constate que `visits-service` est le service qui enregistre le plus grand nombre de défaillances. Elle sélectionne le lien dans le graphique, qui ouvre la page Détails du service correspondant. Elle constate qu'il y a une opération non saine dans le tableau des opérations du service. Elle sélectionne cette opération et constate dans le graphique du volume et de la disponibilité qu'il existe des pics de volume d'appels périodiques qui semblent être liés à des baisses de disponibilité.



Afin d'examiner de plus près les baisses de disponibilité des services, Jane sélectionne l'un des points de données de disponibilité dans le graphique. Un tiroir s'ouvre et affiche les suivis X-Ray corrélés au point de données sélectionné. Elle constate qu'il existe de multiples suivis contenant des défaillances.



Jane sélectionne l'un des suivis corrélés présentant un état de défaut, ce qui ouvre la page détaillée de suivi X-Ray pour le suivi sélectionné. Jane fait défiler la page jusqu'à la section Chronologie des segments et suit le chemin des appels jusqu'à ce qu'elle constate que les appels à une table DynamoDB renvoient des erreurs. Elle sélectionne le segment DynamoDB et accède à l'onglet Exceptions du tiroir de droite.



Jane constate qu'une ressource DynamoDB est mal configurée, ce qui entraîne des erreurs lors des pics de demandes des clients. Le niveau de débit provisionné de la table DynamoDB est régulièrement dépassé, ce qui entraîne des problèmes de disponibilité des services et des SLI non sains. Sur la base de ces informations, son équipe est en mesure de configurer un niveau supérieur de débit provisionné et de garantir une haute disponibilité de l'application.

Métriques d’application standard collectées

⚠ Application Signals est en version préliminaire. Si vous avez des commentaires concernant cette fonctionnalité, vous pouvez nous contacter à l'adresse app-signals-feedback@amazon .com.

Application Signals collecte des métriques d’application standard à partir des services qu’il découvre. Ces indicateurs concernent les aspects les plus critiques des performances d’un service : latence, défaillances et erreurs. Ils peuvent vous aider à identifier les problèmes, à surveiller les tendances en matière de performances et à optimiser les ressources afin d’améliorer l’expérience utilisateur globale.

Le tableau suivant répertorie les métriques collectées par Application Signals. Ces métriques sont envoyées CloudWatch dans l'espace de `AppSignals` noms.

Métrique	Description
`Latency`	Le délai avant le début du transfert de données ne commence qu’une fois la demande effectuée. Unités : millisecondes
`Faults`	Nombre d'erreurs HTTP 5XX côté serveur et d'erreurs d'état du OpenTelemetry span. Unités : aucune
`Errors`	Nombre d’erreurs HTTP 4XX côté client. Ces erreurs sont considérées comme des erreurs de requête qui ne sont pas dues à des problèmes de service. Par conséquent, la métrique `Availability` affichée sur les tableaux de bord d’Application Signals ne considère pas ces erreurs comme des défauts de service. Unités : aucune

La `Availability` métrique affichée sur les tableaux de bord des signaux d'application est calculée sous la forme (1 - **`Faults`** /Total) *100. Les réponses réussies sont toutes les réponses sans erreur 5XX. Les réponses 4XX sont considérées comme réussies lorsqu'Application Signals évalue `Availability`.

Dimensions collectées et combinaisons de dimensions

Les dimensions suivantes sont définies pour chacune des métriques d'application standard. Pour plus d'informations sur les dimensions, veuillez consulter la rubrique Dimensions.

Différentes dimensions sont collectées pour les métriques de service et les métriques de dépendance. Dans les services découverts par Application Signals, lorsque le microservice A appelle le microservice B, le microservice B répond à la requête. Dans ce cas, le microservice A émet des métriques de dépendance et le microservice B émet des métriques de service. Lorsqu'un client appelle le microservice A, le microservice A répond à la requête et émet des métriques de service.

Dimension pour les métriques de service

Les dimensions suivantes sont collectées pour les métriques de service.

Dimension	Description
`Service`	Nom du service.
`Operation`	Nom de l'opération d'API ou de toute autre activité.
`HostedIn.EKS.Cluster`	Nom du cluster Amazon EKS dans lequel les services sont exécutés. Cette dimension n'est collectée que si les services sont exécutés sur Amazon EKS.
`HostedIn.K8s.Namespace`	Nom de l'espace de noms Kubernetes dans lequel les services sont exécutés. Cette dimension n'est collectée que si les services sont exécutés sur Amazon EKS.
`HostedIn.Environment`	Nom défini par l'utilisateur de l'environnement dans lequel les services sont exécutés.

Dimension	Description
	Cette dimension n’est collectée que si les services s’exécutent dans un environnement autre qu’Amazon EKS.

Lorsque vous consultez ces mesures dans la CloudWatch console, vous pouvez choisir de les afficher avec les combinaisons de dimensions suivantes.

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`

Pour les plateformes autres qu’Amazon EKS, vous pouvez également consulter les métriques de service avec les combinaisons de dimensions suivantes.

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`

Dimensions pour les métriques de dépendance

Les dimensions suivantes sont collectées pour les métriques de dépendance.

Dimension	Description
`Service`	Nom du service.
`Operation`	Nom de l’opération d’API ou de toute autre activité.
`RemoteService`	Nom du service distant invoqué.
`RemoteOperation`	Nom de l’opération d’API invoquée.
`HostedIn.EKS.Cluster`	Nom du cluster Amazon EKS dans lequel les services sont exécutés. Cette dimension n’est collectée que si les services sont exécutés sur Amazon EKS.
`HostedIn.K8s.Namespace`	Nom de l’espace de noms Kubernetes dans lequel les services sont exécutés.

Dimension	Description
	Cette dimension n’est collectée que si les services sont exécutés sur Amazon EKS.
`K8s.RemoteNamespace`	Nom de l’espace de noms Kubernetes dans lequel les services de dépendance sont exécutés. Cette dimension n’est collectée que si les services sont exécutés sur Amazon EKS.
`RemoteTarget`	Nom de la ressource invoquée par les appels distants. Cette dimension n’a aucune valeur si les appels distants ne sont pas dirigés vers des ressources spécifiques. Cette dimension n’est collectée que si les services sont exécutés sur Amazon EKS.
`HostedIn.Environment`	Nom défini par l’utilisateur de l’environnement dans lequel les services sont exécutés. Cette dimension n’est collectée que si les services s’exécutent dans un environnement autre qu’Amazon EKS.

Lorsque vous consultez ces mesures dans la CloudWatch console, vous pouvez choisir de les afficher avec les combinaisons de dimensions suivantes.

Exécution sur n’importe quelle plateforme

- `RemoteService`

Exécution sur des clusters Amazon EKS

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`

- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.EKS.Cluster`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `RemoteService`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `K8s.RemoteNamespace`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.EKS.Cluster`, `HostedIn.K8s.Namespace`, `RemoteService`, `RemoteOperation`

Exécution sur des plateformes autres que les clusters Amazon EKS

- `Service`, `Operation`, `HostedIn.Environment`
- `Service`, `HostedIn.Environment`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `Operation`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,
- `Service`, `HostedIn.Environment`, `RemoteService`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`, `RemoteTarget`
- `Service`, `HostedIn.Environment`, `RemoteService`, `RemoteOperation`,

Utiliser une surveillance synthétique

Vous pouvez utiliser Amazon CloudWatch Synthetics pour créer des canaries, des scripts configurables qui s'exécutent selon un calendrier, afin de surveiller vos points de terminaison et vos API. Les scripts Canary suivent les mêmes chemins et effectuent les mêmes actions qu'un client, ce qui vous permet de vérifier continuellement l'expérience de votre client, y compris en l'absence

de trafic de clients sur vos applications. En utilisant les scripts Canary, vous pouvez découvrir les problèmes avant vos clients.

Les scripts Canary sont écrits dans Node.js ou Python. Ils créent des fonctions Lambda dans votre compte qui utilisent Node.js ou Python comme cadre. Les scripts Canary fonctionnent sur les protocoles HTTP et HTTPS. Les canaris utilisent des couches Lambda qui contiennent la bibliothèque CloudWatch Synthetics. La bibliothèque contient la version NodeJS de CloudWatch Synthetics pour les canaris NodeJS et la version Python de Synthetics pour les canaris Python. CloudWatch Les couches appartiennent au compte de CloudWatch service Synthetics. Les bibliothèques ne transmettent ni ne stockent jamais d'informations sur les clients. Toutes les données du client sont stockées uniquement dans le compte client.

Les scripts Canary offrent un accès programmatique à un navigateur Google Chrome sans tête via Puppeteer ou Selenium Webdriver. Pour de plus amples informations sur Puppeteer, consultez Puppeteer. Pour de plus amples informations sur Selenium, consultez www.selenium.dev/.

Les scripts Canary vérifient la disponibilité et la latence de vos points de terminaison, et peuvent stocker des données de temps de chargement et des captures d'écran de l'interface utilisateur. Ils surveillent vos API REST, vos URL et le contenu de votre site Web, et peuvent vérifier les modifications non autorisées apportées par des opérations de hameçonnage, l'injection de code et le scripting intersites.

CloudWatch Synthetics est intégré à Application Signals, qui permet de découvrir et de surveiller les services de votre application, vos clients, les canaries de Synthetics et les dépendances des services. Utilisez Application Signals pour consulter une liste ou une carte visuelle de vos services, consulter les métriques d'intégrité en fonction de vos objectifs de niveau de service (SLO) et effectuer une analyse descendante pour voir les suivis X-Ray corrélés afin de résoudre les problèmes de manière plus détaillée. Pour voir vos scripts canary dans Application Signals, activez le suivi actif X-Ray. Vos scripts canary sont affichés sur la Carte des services associée à vos services et sur la page Détails du service auxquels ils font appel.

Pour une démonstration vidéo des scripts canary, consultez les liens suivants :

- Présentation d'Amazon CloudWatch Synthetics
- Démo d'Amazon CloudWatch Synthetics
- Créez des canaris à l'aide d'Amazon Synthetics CloudWatch
- Surveillance visuelle avec Amazon CloudWatch Synthetics

Vous pouvez exécuter un script Canary une fois ou selon un horaire régulier. Les scripts Canary peuvent être exécutés aussi souvent qu'une fois par minute. Vous pouvez utiliser à la fois les expressions cron et rate pour planifier les scripts Canary.

Pour de plus amples informations sur les problèmes de sécurité à prendre en compte avant de créer et d'exécuter des scripts Canary, veuillez consulter Considérations de sécurité pour les scripts Canary Synthetics.

Par défaut, les canaris créent plusieurs CloudWatch métriques dans l'espace de `CloudWatchSynthetics` noms. Ces métriques ont `CanaryName` comme dimension. Les scripts Canary qui utilisent la fonction `executeStep()` ou `executeHttpStep()` de la bibliothèque de fonctions ont également `StepName` comme dimension. Pour de plus amples informations sur la bibliothèque de fonctions des scripts Canary, veuillez consulter Fonctions de bibliothèque disponibles pour les scripts Canary.

CloudWatch Synthetics s'intègre parfaitement à la X-Ray Trace Map, qui fournit end-to-end une vue d' CloudWatch ensemble AWS X-Ray de vos services afin de vous aider à identifier plus efficacement les problèmes de performance et à identifier les utilisateurs concernés. Les canaris que vous créez avec CloudWatch Synthetics apparaissent sur la carte de traçage. Pour plus d'informations, veuillez consulter la rubrique Carte de suivi X-Ray.

CloudWatch Synthetics est actuellement disponible dans toutes les régions AWS commerciales et les régions. GovCloud

> **Note**
>
> En Asie-Pacifique (Osaka), AWS PrivateLink il n'est pas pris en charge. En Asie-Pacifique (Jakarta), AWS PrivateLink et X-Ray ne sont pas pris en charge.

Rubriques

- Exécution d'un script Canary sur un VPC
- Chiffrement des artefacts de script Canary
- Affichage des politiques et détails sur les scripts Canary
- CloudWatch statistiques publiées par canaries
- Modification ou suppression d'un canary
- Démarrage, arrêt, suppression ou mise à jour de l'exécution de plusieurs canaris
- Surveiller les événements liés aux canaris avec Amazon EventBridge

Rôles et autorisations requis pour les CloudWatch canaris

Les utilisateurs qui créent et gèrent les scripts canary, ainsi que les scripts canary eux-mêmes, doivent disposer de certaines autorisations.

Rôles et autorisations requis pour les utilisateurs qui gèrent les CloudWatch canaris

Pour afficher les détails d'un script canary et les résultats des exécutions de scripts canary, vous devez être connecté en tant qu'utilisateur disposant des politiques `CloudWatchSyntheticsFullAccess` ou `CloudWatchSyntheticsReadOnlyAccess` attachées. Pour lire toutes les données Synthetics dans la console, vous avez également besoin des stratégies `CloudWatchReadOnlyAccess` et `AmazonS3ReadOnlyAccess`. Pour afficher le code source utilisé par les scripts Canary, vous avez également besoin de la stratégie `AWSLambda_ReadOnlyAccess`.

Pour créer des scripts canary, vous devez être connecté en tant qu'utilisateur disposant de la politique `CloudWatchSyntheticsFullAccess` ou d'un ensemble similaire d'autorisations. Pour créer des rôles IAM pour les scripts Canary, vous avez également besoin de la déclaration de stratégie en ligne suivante :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
```

```
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

> **Important**
>
> Le fait d'accorder à un utilisateur `iam:AttachRolePolicy` les autorisations `iam:CreateRoleiam:CreatePolicy`,, et lui donne un accès administratif complet à votre AWS compte. Par exemple, un utilisateur disposant de ces autorisations peut créer une stratégie disposant d'autorisations complètes pour toutes les ressources et peut attacher cette stratégie à n'importe quel rôle. Sélectionnez attentivement les personnes auxquelles vous accordez ces autorisations.

Pour plus d'informations sur l'association des politiques et l'octroi d'autorisations aux utilisateurs, consultez Modification des autorisations pour un utilisateur IAM et Pour intégrer une politique en ligne pour un utilisateur ou un rôle.

Rôles et autorisations requis pour les scripts Canary

Chaque script canary doit être associé à un rôle IAM auquel certaines autorisations sont attachées. Lorsque vous créez un canari à l'aide de la CloudWatch console, vous pouvez choisir que CloudWatch Synthetics crée un rôle IAM pour le canari. Si vous le faites, le rôle disposera des autorisations nécessaires.

Si vous souhaitez créer le rôle IAM vous-même ou créer un rôle IAM que vous pouvez utiliser lorsque vous utilisez la AWS CLI ou des API permettant de créer un script canary, le rôle doit contenir les autorisations répertoriées dans cette section.

Tous les rôles IAM pour les scripts canary doivent inclure la déclaration de politique d'approbation suivante.

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

```
            "Principal": {
                "Service": "lambda.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

De plus, le rôle IAM du script canary nécessite l'une des déclarations suivantes.

Canary de base qui n'utilise pas AWS KMS ou n'a pas besoin d'un accès Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/s3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/s3/bucket/that/contains/canary/results"

            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
```

```
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
               "StringEquals": {
                "cloudwatch:namespace": "CloudWatchSynthetics"
               }
            }
        }
    ]
}
```

Canary qui crypte AWS KMS les artefacts Canary mais n'a pas besoin d'un accès Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
```

```
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

Canary qui n'utilise pas Amazon VPC AWS KMS mais qui a besoin d'un accès à Amazon VPC

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
```

```
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       }
    ]
}
```

Canary qui crypte AWS KMS les artefacts Canary et qui a également besoin d'un accès Amazon VPC

Si vous mettez à jour un script canary non VPC pour qu'il commence à utiliser un VPC, vous devez mettre à jour le rôle du script canary pour inclure les autorisations d'interface réseau répertoriées dans la politique suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::path/to/your/S3/bucket/canary/results/folder"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::name/of/the/S3/bucket/that/contains/canary/results"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:canary_region_name:canary_account_id:log-group:/aws/
lambda/cwsyn-canary_name-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListAllMyBuckets",
                "xray:PutTraceSegments"
            ],
```

```
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CloudWatchSynthetics"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": [
                "*"
            ]
       },
       {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource":
 "arn:aws:kms:KMS_key_region_name:KMS_key_account_id:key/KMS_key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": [
                        "s3.region_name_of_the_canary_results_S3_bucket.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

AWS politiques gérées pour CloudWatch Synthetics

Pour ajouter des autorisations à des utilisateurs, des groupes et des rôles, il est plus facile d'utiliser des politiques gérées par AWS que d'écrire des politiques vous-même. Il faut du temps et de l'expertise pour créer des politiques gérées par le client IAM qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votre AWS compte. Pour de plus amples informations sur les politiques gérées par AWS , veuillez consulter Stratégies gérées par AWS Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services modifient occasionnellement les autorisations dans une politique gérée par AWS . Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée.

CloudWatch Mises à jour des politiques gérées par Synthetics AWS

Consultez les détails des mises à jour apportées aux politiques AWS gérées pour CloudWatch Synthetics 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 Historique du CloudWatch document.

Modification	Description	Date
Actions redondantes supprimées de CloudWatchSyntheticsFullAccess	CloudWatch Synthetics a supprimé les actions `lambda:GetLayerVersionByArn` et CloudWatchSyntheticsFullAccessde `s3:PutBucketEncryption` la politique car ces actions étaient redondantes par rapport aux autres autorisations de la politique . Les actions supprimées ne fournissaient aucune autorisation et il n'y a aucune modificat	12 mars 2021

Modification	Description	Date
	ion nette des autorisations accordées par la stratégie.	
CloudWatch Synthetics a commencé à suivre les changements	CloudWatch Synthetics a commencé à suivre les modifications apportées à ses politiques gérées. AWS	10 mars 2021

CloudWatchSyntheticsFullAccess

Voici le contenu de la politique `CloudWatchSyntheticsFullAccess` :

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "synthetics:*"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:CreateBucket",
            "s3:PutEncryptionConfiguration"
         ],
         "Resource":[
            "arn:aws:s3:::cw-syn-results-*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles",
            "s3:ListAllMyBuckets",
            "s3:GetBucketLocation",
            "xray:GetTraceSummaries",
            "xray:BatchGetTraces",
```

```
            "apigateway:GET"
         ],
         "Resource":"*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject",
            "s3:ListBucket"
         ],
         "Resource":"arn:aws:s3:::cw-syn-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObjectVersion"
         ],
         "Resource":"arn:aws:s3:::aws-synthetics-library-*"
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ],
         "Condition":{
            "StringEquals":{
               "iam:PassedToService":[
                  "lambda.amazonaws.com",
                  "synthetics.amazonaws.com"
               ]
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "iam:GetRole"
         ],
         "Resource":[
            "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
         ]
```

```
        },
        {
           "Effect":"Allow",
           "Action":[
              "cloudwatch:GetMetricData",
              "cloudwatch:GetMetricStatistics"
           ],
           "Resource":"*"
        },
        {
           "Effect":"Allow",
           "Action":[
              "cloudwatch:PutMetricAlarm",
              "cloudwatch:DeleteAlarms"
           ],
           "Resource":[
              "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
           ]
        },
        {
           "Effect":"Allow",
           "Action":[
              "cloudwatch:DescribeAlarms"
           ],
           "Resource":[
              "arn:aws:cloudwatch:*:*:alarm:*"
           ]
        },
        {
           "Effect":"Allow",
           "Action":[
              "lambda:CreateFunction",
              "lambda:AddPermission",
              "lambda:PublishVersion",
              "lambda:UpdateFunctionConfiguration",
              "lambda:GetFunctionConfiguration"
           ],
           "Resource":[
              "arn:aws:lambda:*:*:function:cwsyn-*"
           ]
        },
        {
           "Effect":"Allow",
           "Action":[
```

```
            "lambda:GetLayerVersion",
            "lambda:PublishLayerVersion"
         ],
         "Resource":[
            "arn:aws:lambda:*:*:layer:cwsyn-*",
            "arn:aws:lambda:*:*:layer:Synthetics:*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:ListTopics"
         ],
         "Resource":[
            "*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sns:CreateTopic",
            "sns:Subscribe",
            "sns:ListSubscriptionsByTopic"
         ],
         "Resource":[
            "arn:*:sns:*:*:Synthetics-*"
         ]
      }
   ]
}
```

CloudWatchSyntheticsReadOnlyAccess

Voici le contenu de la politique `CloudWatchSyntheticsReadOnlyAccess` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Limiter un utilisateur pour afficher des canarys spécifiques

Vous pouvez limiter la capacité d'un utilisateur à afficher des informations sur les canarys, de sorte qu'il ne puisse voir que les informations sur les canarys que vous spécifiez. Pour ce faire, utilisez une politique IAM avec une déclaration `Condition` semblable à ce qui suit et attachez cette politique à un utilisateur ou à un rôle IAM.

L'exemple suivant montre comment limiter l'utilisateur à afficher uniquement des informations sur `name-of-allowed-canary-1` et `name-of-allowed-canary-2`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "synthetics:Names": [
                        "name-of-allowed-canary-1",
                        "name-of-allowed-canary-2"
                    ]
                }
            }
```

```
        }
    ]
}
```

CloudWatch Synthetics permet de répertorier jusqu'à cinq éléments de la gamme. `synthetics:Names`

Vous pouvez également créer une politique qui utilise un * comme caractère générique dans les noms canary qui doivent être autorisés, comme dans l'exemple suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "synthetics:DescribeCanaries",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringLike": {
                    "synthetics:Names": [
                        "my-team-canary-*"
                    ]
                }
            }
        }
    ]
}
```

Tout utilisateur connecté avec l'une de ces politiques en pièce jointe ne peut pas utiliser la CloudWatch console pour consulter les informations Canary. Ils peuvent consulter les informations relatives aux canaris uniquement pour les canaris autorisés par la politique et uniquement à l'aide de l'DescribeCanariesAPI ou de la commande AWS CLI describe-canaries.

Création d'un Canary

> **Important**
>
> Assurez-vous que vous utilisez des scripts Canary Synthetics afin de surveiller uniquement les points de terminaison et les API dont vous êtes propriétaire ou pour lesquels vous

disposez d'une autorisation. Selon les paramètres de fréquence des scripts Canary, ces points de terminaison peuvent connaître un trafic accru.

Lorsque vous utilisez la CloudWatch console pour créer un canari, vous pouvez utiliser un plan fourni par CloudWatch pour créer votre canari ou vous pouvez écrire votre propre script. Pour plus d'informations, consultez Utilisation des modèles de scripts Canary.

Vous pouvez également créer un canari en utilisant AWS CloudFormation si vous utilisez votre propre script pour le canari. Pour plus d'informations, consultez AWS::Synthetics::Canaryle guide de AWS CloudFormation l'utilisateur.

Si vous écrivez votre propre script, vous pouvez utiliser plusieurs fonctions intégrées par CloudWatch Synthetics dans une bibliothèque. Pour plus d'informations, consultez Versions d'exécution Synthetics.

> **Note**
>
> Lorsque vous créez un canari, l'une des couches créées est une couche Synthetics précédée de. `Synthetics` Cette couche appartient au compte de service Synthetics et contient le code d'exécution.

Pour créer un script Canary

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Synthetics Canaries.
3. Choisissez Create Canary (Créer un script Canary).
4. Sélectionnez l'une des méthodes suivantes :
   - Pour baser votre script Canary sur un modèle de script, choisissez Use a blueprint (Utiliser un modèle), puis le type de script Canary que vous souhaitez créer. Pour de plus amples informations sur les activités de chaque type de modèle, veuillez consulter Utilisation des modèles de scripts Canary.
   - Pour télécharger votre propre script Node.js afin de créer un script Canary personnalisé, choisissez Upload a script (Charger un script).

Vous pouvez ensuite faire glisser votre script dans la zone Script ou choisir Browse files (Parcourir les fichiers) pour accéder au script dans votre système de fichiers.

- Pour importer votre script à partir d'un compartiment S3, choisissez Import from S3 (Importer à partir de S3). Sous Source location (Emplacement source), saisissez le chemin d'accès complet à votre script Canary ou choisissez Browse S3 (Parcourir S3).

  Vous devez disposer d'autorisations `s3:GetObject` et `s3:GetObjectVersion` pour le compartiment S3 que vous utilisez. Le bucket doit se trouver dans la même AWS région que celle où vous créez le canari.

5. Sous Nom, entrez le nom du script Canary. Le nom est utilisé sur de nombreuses pages, donc nous vous recommandons de lui donner un nom descriptif qui le distingue des autres scripts Canary.
6. Sous Application or endpoint URL (URL de l'application ou du point de terminaison), entrez l'URL que vous souhaitez tester avec le script Canary. Cette URL doit inclure le protocole (par exemple, https://).

   Si vous voulez que le script Canary teste un point de terminaison sur un VPC, vous devez également saisir des informations sur votre VPC plus loin dans cette procédure.
7. Si vous utilisez votre propre script pour le script Canary, sous le Lambda handler (Gestionnaire Lambda), saisissez le point d'entrée à partir duquel le script Canary doit commencer. Si vous utilisez une exécution antérieure à `syn-nodejs-puppeteer-3.4` ou à `syn-python-selenium-1.1`, la chaîne que vous saisissez doit se terminer par `.handler`. Si vous utilisez `syn-nodejs-puppeteer-3.4`, `syn-python-selenium-1.1` ou une exécution ultérieure, cette restriction ne s'applique pas.
8. Si vous utilisez des variables d'environnement dans votre script, choisissez Environment variables (Variables d'environnement), puis spécifiez une valeur pour chaque variable d'environnement définie dans votre script. Pour plus d’informations, consultez Variables d’environnement.
9. Sous Schedule (Planification), choisissez si vous souhaitez exécuter ce script Canary une seule fois, l'exécuter en continu à l'aide d'une expression rate ou le planifier à l'aide d'une expression cron.

   - Lorsque vous utilisez la CloudWatch console pour créer un canari qui fonctionne en continu, vous pouvez choisir un débit compris entre une fois par minute et une fois par heure.

- Pour plus d'informations sur l'écriture d'une expression cron pour la planification d'un script Canary, consultez Planification d'exécutions de scripts Canary à l'aide de cron.

10. (Facultatif) Pour définir une valeur de délai d'attente pour le canary, choisissez Configuration supplémentaire puis spécifiez la valeur du délai d'attente. Ne pas dépasser 15 secondes pour permettre les démarrages à froid Lambda et le temps nécessaire pour démarrer l'instrumentation canary.
11. Sous Conservation des données, spécifiez la durée de conservation des informations sur les exécutions de scripts Canary échouées et réussies. La plage va de 1 à 455 jours.

    Ce paramètre affecte uniquement les données que CloudWatch Synthetics stocke et affiche dans la console. Cela n'affecte pas les données stockées dans vos compartiments Amazon S3, ni les journaux ou les mesures publiés par Canary.
12. Sous Data Storage (Stockage de données), sélectionnez le compartiment S3 à utiliser pour stocker les données provenant des exécutions des tests du script Canary. Le nom du compartiment ne peut pas contenir de point (.). Si vous laissez ce champ vide, un compartiment S3 par défaut est utilisé ou créé.

    Si vous utilisez l'exécution `syn-nodejs-puppeteer-3.0` ou version ultérieure, lorsque vous saisissez l'URL du compartiment dans la zone de texte, vous pouvez spécifier un compartiment dans la région actuelle ou dans une autre région. Si vous utilisez une version d'exécution antérieure, le compartiment doit se trouver dans la région actuelle.
13. (Facultatif) Par défaut, les canaris stockent leurs artefacts sur Amazon S3, et les artefacts sont chiffrés au repos à l'aide d'une AWS clé gérée AWS KMS . Vous pouvez utiliser une autre option de chiffrement en choisissant Additional configuration (Configuration supplémentaire) dans la section Data Storage (Stockage de données). Vous pouvez ensuite choisir le type de clé à utiliser pour le chiffrement. Pour plus d’informations, consultez Chiffrement des artefacts de script Canary.
14. Sous Access permissions (Autorisations d'accès), choisissez si vous souhaitez créer un rôle IAM pour exécuter le script Canary ou utiliser un rôle existant.

    Si CloudWatch Synthetics crée le rôle, il inclut automatiquement toutes les autorisations nécessaires. Si vous souhaitez créer le rôle vous-même, consultez Rôles et autorisations requis pour les scripts Canary pour plus d'informations sur les autorisations nécessaires.

Si vous utilisez la CloudWatch console pour créer un rôle pour un canari lorsque vous créez le canari, vous ne pouvez pas réutiliser le rôle pour d'autres canaris, car ces rôles sont spécifiques à un seul canari. Vous pouvez utiliser un rôle existant, si vous avez créé manuellement un rôle qui fonctionne pour plusieurs scripts Canary.

Pour utiliser un rôle existant, vous devez avoir l'autorisation `iam:PassRole` de transmettre ce rôle à Synthetics et à Lambda. Vous devez également avoir l'autorisation `iam:GetRole`.

15. (Facultatif) Sous Alarmes, indiquez si vous souhaitez que des CloudWatch alarmes par défaut soient créées pour ce canari. Si vous choisissez de créer des alertes, elles sont créées selon la convention de nom suivante :`Synthetics-Alarm-`*`canaryName`*`-`*`index`*

    `index` est un numéro représentant chaque alerte créée pour ce script Canary. La première alerte a un index de 1, la deuxième alerte a un index de 2, et ainsi de suite.

16. (Facultatif) Pour que ce script Canary teste un point de terminaison figurant sur un VPC, choisissez VPC settings (Paramètres VPC) et procédez comme suit :

    a. Sélectionnez le VPC qui héberge le point de terminaison.

    b. Sélectionnez un ou plusieurs sous-réseaux sur votre VPC. Vous devez sélectionner un sous-réseau privé, car une instance Lambda ne peut pas être configurée pour s'exécuter dans un sous-réseau public lorsqu'une adresse IP ne peut pas être attribuée à l'instance Lambda lors de l'exécution. Pour de plus amples informations, veuillez consulter Configuration d'une fonction Lambda pour accéder aux ressources d'un VPC.

    c. Sélectionnez un ou plusieurs groupes de sécurité sur votre VPC.

    Si le point de terminaison se trouve sur un VPC, vous devez autoriser votre Canary à envoyer des informations à Amazon S3 CloudWatch et à Amazon S3. Pour plus d’informations, consultez Exécution d'un script Canary sur un VPC.

17. (Facultatif) Sous Tags (Balises), ajoutez une ou plusieurs paires clé/valeur comme balises pour ce script Canary. Les balises peuvent vous aider à identifier et à organiser vos AWS ressources et à suivre vos AWS coûts. Pour plus d’informations, consultez Marquer vos ressources Amazon CloudWatch .

18. (Facultatif) Sous Active tracing (Suivi actif), choisissez d'activer ou non le suivi X-Ray actif pour ce script Canary. Cette option est disponible uniquement si le script Canary utilise la version d'exécution syn-nodejs-2.0 ou ultérieure. Pour plus d’informations, consultez Scripts Canary et suivi X-Ray.

Ressources créées pour les scripts Canary

Lorsque vous créez un script Canary, les ressources suivantes sont créées :

- Un rôle IAM portant le nom `CloudWatchSyntheticsRole-`*`canary-name`*`-`*`uuid`* (si vous utilisez la CloudWatch console pour créer le canari et que vous spécifiez qu'un nouveau rôle sera créé pour le canari)
- Une stratégie IAM portant le nom `CloudWatchSyntheticsPolicy-`*`canary-name`*`-`*`uuid`*.
- Un compartiment S3 portant le nom `cw-syn-results-`*`accountID`*`-`*`region`*.
- Des alertes portant le nom `Synthetics-Alarm-`*`MyCanaryName`* (si vous voulez créer des alertes pour le script Canary)
- Fonctions et couches Lambda, si vous utilisez un modèle pour créer le script Canary. Ces ressources ont le préfixe `cwsyn-`*`MyCanaryName`*.
- CloudWatch Enregistre les groupes de journaux avec le nom`/aws/lambda/cwsyn-`*`MyCanaryName`*`-`*`randomId`*.

Utilisation des modèles de scripts Canary

Cette section fournit des détails sur chacun des modèles de scripts Canary et les tâches auxquelles chaque modèle est le mieux adapté. Des plans sont fournis pour les types de scripts Canary suivants :

- Moniteur de pulsations
- Script Canary d'API
- Vérificateur des liens cassés
- Surveillance visuelle
- Enregistreur de scripts Canary
- Workflow GUI

Lorsque vous utilisez un plan pour créer un canari, lorsque vous remplissez les champs de la CloudWatch console, la zone éditeur de script de la page affiche le canari que vous créez sous la forme d'un script Node.js. Vous pouvez également modifier votre script Canary dans cette zone pour le personnaliser davantage.

Surveillance des pulsations

Les scripts de pulsations chargent l'URL spécifiée et stockent une capture d'écran de la page et un fichier d'archive HTTP (fichier HAR). Ils stockent également des journaux des URL consultées.

Vous pouvez utiliser les fichiers HAR pour afficher des données de performances détaillées sur les pages web. Vous pouvez analyser la liste des demandes web et détecter des problèmes liés aux performances, notamment le temps de chargement d'un élément.

Si votre script Canary utilise la version d'exécution `syn-nodejs-puppeteer-3.1` ou ultérieure, vous pouvez utiliser le modèle de surveillance des pulsations pour surveiller plusieurs URL et voir l'état, la durée, les captures d'écran associées et la raison de l'échec de chaque URL dans le résumé des étapes du rapport d'exécution du script Canary.

Script Canary d'API

Les scripts Canary d'API peuvent tester les fonctions de base de lecture et d'écriture d'une API REST. REST, qui signifie transfert d'état représentatif, est un ensemble de règles que les développeurs suivent lors de la création d'une API. L'une de ces règles stipule qu'un lien vers une URL spécifique doit renvoyer un élément de données.

Les scripts Canary peuvent fonctionner avec n'importe quelle API et tester tous les types de fonctionnalités. Chaque script Canary peut effectuer plusieurs appels d'API.

Dans les scripts Canary qui utilisent la version d'exécution `syn-nodejs-2.2` ultérieure, le modèle de script Canary d'API prend en charge les scripts Canary à plusieurs étapes qui contrôlent vos API en tant qu'étapes HTTP. Vous pouvez tester plusieurs API dans un seul script Canary. Chaque étape est une demande distincte qui peut accéder à des URL différentes, utiliser des en-têtes différents et des règles différentes pour déterminer si les en-têtes et les corps de réponse sont capturés. En ne capturant pas les en-têtes et le corps de la réponse, vous pouvez empêcher l'enregistrement de données sensibles.

Chaque demande dans un script Canary d'API comprend les informations suivantes :

- Le point de terminaison, qui est l'URL que vous demandez.
- La méthode, qui est le type de demande envoyé au serveur. Les API REST prennent en charge les opérations GET (lecture), POST (écriture), PUT (mise à jour), PATCH (mise à jour) et DELETE (suppression).

- Les en-têtes, qui fournissent des informations à la fois au client et au serveur. Ils sont utilisés pour l'authentification et fournir des informations sur le contenu du corps. Pour une liste d'en-têtes valides, consultez En-têtes HTTP.
- Les données (ou le corps) qui contiennent des informations à envoyer au serveur. Utilisées uniquement pour les demandes POST, PUT, PATCH ou DELETE.

Le modèle de script Canary d'API prend en charge les méthodes GET et POST. Lorsque vous utilisez ce modèle, vous devez spécifier des en-têtes. Par exemple, vous pouvez spécifier **`Authorization`** en tant que Key (Clé) et spécifier les données d'autorisation nécessaires en tant que Value (Valeur) pour cette clé.

Si vous testez une demande POST, vous spécifiez également le contenu à publier dans le champ Data (Données).

Intégration à API Gateway

Le modèle d'API est intégré à Amazon API Gateway. Cela vous permet de sélectionner une API API Gateway et un stage à partir du même AWS compte et de la même région que le Canary, ou de télécharger un modèle Swagger depuis API Gateway pour la surveillance des API entre comptes et entre régions. Vous pouvez ensuite choisir le reste des détails dans la console pour créer le script Canary, au lieu de les saisir à partir de zéro. Pour plus d'informations sur API Gateway, consultez Qu'est-ce qu'Amazon API Gateway ?.

Utilisation d'une API privée

Vous pouvez créer un script canary utilisant une API privée dans Amazon API Gateway. Pour de plus amples informations, veuillez consulter Création d'une API privée dans Amazon API Gateway ?

Vérificateur des liens cassés

Le vérificateur de liens rompus collecte tous les liens à l'intérieur de l'URL que vous testez en utilisant `document.getElementsByTagName('a')`. Il teste uniquement le nombre de liens que vous spécifiez et l'URL elle-même est considérée comme le premier lien. Par exemple, si vous voulez vérifier tous les liens d'une page qui en contient cinq, vous devez spécifier que le script Canary doit suivre six liens.

Les scripts Canary du vérificateur de liens rompus créés à l'aide de l'exécution `syn-nodejs-2.0-beta` ou version ultérieure prennent en charge les fonctions supplémentaires suivantes :

- Fourniture d'un rapport qui inclut les liens qui ont été vérifiés, le code d'état, la raison de l'échec (le cas échéant) et les captures d'écran des pages source et de destination.
- Lors de l'affichage des résultats de scripts Canary, vous pouvez utiliser un filtre pour afficher uniquement les liens rompus, puis réparer ces liens en fonction de la raison de l'échec.
- Cette version effectue pour chaque lien des captures d'écran annotées de la page source et met en surbrillance l'ancre où le lien a été trouvé. Les composants masqués ne sont pas annotés.
- Vous pouvez configurer cette version pour effectuer des captures d'écran des pages source et de destination, des pages source uniquement ou des pages de destination uniquement.
- Cette version résout un problème dans la version précédente qui provoquait l'arrêt du script Canary après le premier lien rompu, même lorsque d'autres liens sont récupérés à partir de la première page.

Si vous souhaitez mettre à jour un script Canary existant utilisant `syn-1.0` afin qu'il utilise la nouvelle exécution, vous devez supprimer et recréer le script Canary. La mise à jour d'un script Canary existant vers la nouvelle exécution ne rend pas ces fonctions disponibles.

Un script Canary de vérification des liens rompus détecte les types d'erreurs de liens suivants :

- 404 Page introuvable
- Nom d'hôte non valide
- URL incorrecte. Par exemple, il manque un crochet dans l'URL, elle a des barres obliques supplémentaires ou elle utilise le mauvais protocole.
- Code de réponse HTTP non valide.
- Le serveur hôte renvoie des réponses vides sans contenu ni code de réponse.
- Les requêtes HTTP expirent constamment pendant l'exécution du script Canary.
- L'hôte coupe systématiquement les connexions parce qu'il est mal configuré ou trop occupé.

Modèle de surveillance visuelle

Le modèle de surveillance visuelle inclut un code permettant de comparer les captures d'écran prises lors de l'exécution d'un script Canary aux captures d'écran prises lors de l'exécution d'un script Canary de référence. Si l'écart entre les deux captures d'écran dépasse un pourcentage de seuil, le script Canary échoue. La surveillance visuelle est prise en charge dans les canaris utilisant syn-puppeteer-node-3.2 et versions ultérieures. Elle n'est actuellement pas prise en charge dans les scripts Canary exécutant Python et Selenium.

Le modèle de surveillance visuelle inclut la ligne de code suivante dans le script Canary de modèle par défaut, cette ligne activant la surveillance visuelle.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

La première fois que le script Canary s'exécute avec succès après l'ajout de cette ligne au script, il utilise les captures d'écran prises au cours de cette exécution comme référence pour les comparaisons. Après cette première exécution de Canary, vous pouvez utiliser la CloudWatch console pour modifier le Canary afin d'effectuer l'une des opérations suivantes :

- Définir la prochaine exécution du script Canary comme nouvelle référence.
- Dessiner des limites sur la capture d'écran de référence actuelle pour désigner les zones de la capture d'écran à ignorer lors des comparaisons visuelles.
- Empêcher une capture d'écran d'être utilisée pour la surveillance visuelle.

Pour plus d'informations sur l'utilisation de la CloudWatch console pour modifier un canari, consultezModification ou suppression d'un canary.

Vous pouvez également modifier le canary run utilisé comme référence en utilisant les `lastrun` paramètres `nextrun` or ou en spécifiant un identifiant Canary Run dans l'UpdateCanaryAPI.

Lorsque vous utilisez le modèle de surveillance visuelle, vous saisissez l'URL à laquelle vous souhaitez effectuer la capture d'écran et vous spécifiez un seuil de différence sous forme de pourcentage. Après l'exécution de référence, les futures exécutions du script Canary qui détectent une différence visuelle supérieure à ce seuil déclenchent un échec du script Canary. Après l'exécution de référence, vous pouvez également modifier le script Canary pour « dessiner » des limites sur la capture d'écran de référence afin d'ignorer des zones pendant la surveillance visuelle.

La fonction de surveillance visuelle est alimentée par la boîte à outils logicielle ImageMagick open source. Pour plus d'informations, consultez ImageMagick.

Enregistreur de scripts Canary

Avec le plan Canary Recorder, vous pouvez utiliser le CloudWatch Synthetics Recorder pour enregistrer vos actions de clic et de saisie sur un site Web et générer automatiquement un script Node.js qui peut être utilisé pour créer un Canary suivant les mêmes étapes. Le CloudWatch Synthetics Recorder est une extension Google Chrome fournie par Amazon.

Crédits : The CloudWatch Synthetics Recorder est basé sur l'enregistreur Headless.

Pour plus d’informations, consultez Utilisation de l' CloudWatch enregistreur Synthetics pour Google Chrome.

Générateur de flux de travail GUI

Le modèle de générateur de flux de travail GUI vérifie que des actions peuvent être effectuées sur votre page web. Par exemple, si vous avez une page web avec un formulaire de connexion, le script Canary peut remplir les champs utilisateur et mot de passe et envoyer le formulaire pour vérifier le bon fonctionnement de la page web.

Lorsque vous utilisez un modèle pour créer ce type de script Canary, vous spécifiez les actions que ce dernier doit effectuer sur la page web. Les actions que vous pouvez utiliser sont les suivantes :

- Click (Cliquer) : sélectionne l'élément que vous spécifiez et simule un utilisateur qui clique sur l'élément ou le choisit.

  Pour spécifier l'élément dans un script Node.js, utilisez `[id=]` ou `a[class=]`.

  Pour spécifier l'élément dans un script Python, utilisez `xpath //*[@id=]` ou `//*[@class=]`.
- Verify selector (Vérifier le sélecteur) : vérifie que l'élément spécifié existe sur la page web. Ce test est utile pour vérifier qu'une action précédente a conduit les éléments corrects à remplir la page.

  Pour spécifier l'élément à vérifier dans un script Node.js, utilisez `[id=]` ou `a[class=]`.

  Pour spécifier l'élément à vérifier dans un script Python, utilisez `xpath //*[@id=]` ou `//*[class=]`.
- Verify text (Vérifier le texte) : vérifie que la chaîne spécifiée est contenue dans l'élément cible. Ce test est utile pour vérifier qu'une action précédente a provoqué l'affichage du bon texte.

  Pour spécifier l'élément dans un script Node.js, utilisez un format tel que `div[@id=]//h1`, parce que cette action utilise la fonction `waitForXPath` dans Puppeteer.

  Pour spécifier l'élément dans un script Python, utilisez un format xpath tel que`//*[@id=]` ou //*[@class=], parce que cette action utilise la fonction `implicitly_wait` dans Selenium.
- Input text (Saisir du texte) : écrit le texte spécifié dans l'élément cible.

  Pour spécifier l'élément à vérifier dans un script Node.js, utilisez `[id=]` ou `a[class=]`.

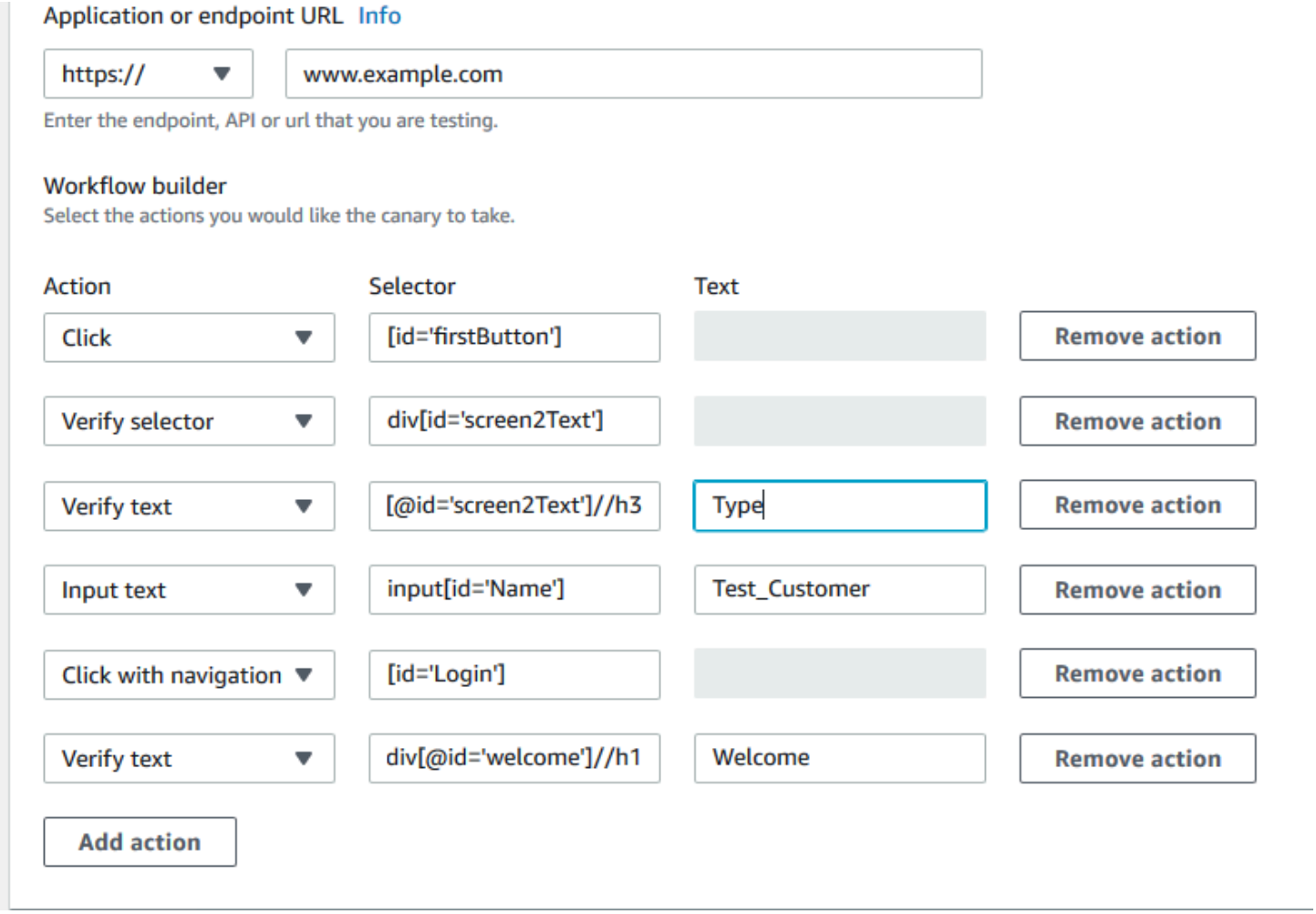
  Pour spécifier l'élément à vérifier dans un script Python, utilisez `xpath //*[@id=]` ou `//*[@class=]`.

- Click with navigation (Cliquer avec la navigation) : attend que la page soit entièrement chargée après avoir choisi l'élément spécifié. Cela s'avère très utile lorsque vous devez recharger la page.

  Pour spécifier l'élément dans un script Node.js, utilisez `[id=]` ou `a[class=]`.

  Pour spécifier l'élément dans un script Python, utilisez `xpath //*[@id=]` ou `//*[@class=]`.

Par exemple, le modèle suivant utilise Node.js. Il clique sur le firstButton dans l'URL spécifiée, il vérifie que le sélecteur attendu s'affiche avec le texte attendu, il saisit le nom `Test_Customer` dans le champ Name (Nom), il clique sur le bouton Login (Connexion) et il vérifie ensuite que la connexion a réussi en recherchant le texte Welcome (Bienvenue) sur la page suivante.



Les scripts Canary du flux de travail GUI qui utilisent les exécutions suivantes fournissent également un résumé des étapes exécutées pour chaque exécution de script Canary. Vous pouvez utiliser les captures d'écran et le message d'erreur associés à chaque étape pour trouver la cause racine de l'échec.

- `syn-nodejs-2.0` ou version ultérieure
- `syn-python-selenium-1.0` ou version ultérieure

Utilisation de l' CloudWatch enregistreur Synthetics pour Google Chrome

Amazon fournit un enregistreur de CloudWatch synthetics pour vous aider à créer des canaris plus facilement. L'enregistreur est une extension Google Chrome.

L'enregistreur enregistre vos actions de clic et de saisie sur un site web et génère automatiquement un script Node.js qui peut être utilisé pour créer un script Canary qui suit les mêmes étapes.

Une fois que vous avez commencé à enregistrer, le CloudWatch Synthetics Recorder détecte vos actions dans le navigateur et les convertit en script. Si nécessaire, vous pouvez mettre en pause et reprendre l'enregistrement. Lorsque vous arrêtez l'enregistrement, l'enregistreur produit un script Node.js de vos actions, que vous pouvez facilement copier avec le bouton Copy to Clipboard (Copier dans le presse-papier). Vous pouvez ensuite utiliser ce script pour créer un canari dans CloudWatch Synthetics.

Crédits : The CloudWatch Synthetics Recorder est basé sur l'enregistreur Headless.

Installation de l' CloudWatch extension Synthetics Recorder pour Google Chrome

Pour utiliser le CloudWatch Synthetics Recorder, vous pouvez commencer à créer un canari et choisir le plan Canary Recorder. Si vous le faites alors que vous n'avez pas encore téléchargé l'enregistreur, la console CloudWatch Synthetics fournit un lien pour le télécharger.

Vous pouvez également suivre ces étapes pour télécharger et installer l'enregistreur directement.

Pour installer l'enregistreur CloudWatch Synthetics

1. À l'aide de Google Chrome, rendez-vous sur ce site Web : https://chrome.google.com/webstore/detail/ cloudwatch-synthetics-rec /bhdnlmmgiplmbcdmkkdfplenecpegfno
2. Choisissez Ajouter à Chrome, puis Ajouter l'extension.

Utilisation de l' CloudWatch enregistreur Synthetics pour Google Chrome

Pour utiliser le CloudWatch Synthetics Recorder afin de créer un canari, vous pouvez choisir Create Canary dans CloudWatch la console, puis choisir Utiliser un plan, Canary Recorder. Pour plus d’informations, consultez Création d'un Canary.

Vous pouvez également utiliser l'enregistreur pour enregistrer des étapes sans les utiliser immédiatement pour créer un script Canary.

Pour utiliser le CloudWatch Synthetics Recorder pour enregistrer vos actions sur un site Web

1. Accédez à la page que vous souhaitez contrôler.
2. Choisissez l'icône des extensions Chrome, puis CloudWatchSynthetics Recorder.
3. Choisissez Start Recording (Démarrer l'enregistrement).
4. Réalisez les étapes que vous souhaitez enregistrer. Pour suspendre l'enregistrement, choisissez Pause.
5. Lorsque vous avez terminé d'enregistrer le flux de travail, sélectionnez Stop recording (Arrêter l'enregistrement).
6. Choisissez Copy to clipboard (Copier dans le presse-papiers) pour copier le script généré dans votre presse-papier. Ou, si vous voulez recommencer, choisissez New recording (Nouvel enregistrement).
7. Pour créer un script Canary avec le script copié, vous pouvez coller votre script copié dans l'éditeur en ligne du modèle de l'enregistreur ou l'enregistrer dans un compartiment Amazon S3 et l'importer à partir de là.
8. Si vous ne créez pas immédiatement un script Canary, vous pouvez enregistrer votre script enregistré dans un fichier.

Limites connues du CloudWatch Synthetics Recorder

L' CloudWatch enregistreur Synthetics pour Google Chrome présente actuellement les limites suivantes.

- Les éléments HTML qui n'ont pas d'ID utiliseront des sélecteurs CSS. Cela peut interrompre les scripts Canary si la structure de la page web change plus tard. Nous prévoyons de fournir certaines options de configuration (comme l'utilisation de data-id) pour cette limitation dans une future version de l'enregistreur.
- L'enregistreur ne prend pas en charge les actions telles que le double-clic ou le copier/coller, et il ne prend pas en charge les combinaisons de touches telles que CMD+0.
- Pour vérifier la présence d'un élément ou d'un texte sur la page, les utilisateurs doivent ajouter des assertions une fois le script généré. L'enregistreur ne prend pas en charge la vérification d'un élément sans effectuer d'action sur cet élément. Ceci est semblable aux options « Verify text

(Vérifier le texte) » ou « Verify element (Vérifier l'élément) » dans le générateur de flux de travail de script Canary. Nous prévoyons d'ajouter la prise en charge de quelques assertions dans une future version de l'enregistreur.

- L'enregistreur enregistre toutes les actions dans l'onglet où l'enregistrement est lancé. Il n'enregistre pas les fenêtres contextuelles (par exemple, pour permettre le suivi de l'emplacement) ou la navigation vers différentes pages à partir de fenêtres contextuelles.

Versions d'exécution Synthetics

Lorsque vous créez ou mettez à jour un script Canary, vous choisissez une version d'environnement d'exécution Synthetics pour le script Canary. Une exécution Synthetics est une combinaison du code Synthetics qui appelle votre gestionnaire de scripts et des couches Lambda de dépendances groupées.

CloudWatch Synthetics prend actuellement en charge les environnements d'exécution qui utilisent Node.js pour les scripts et le framework Puppeteer, ainsi que les environnements d'exécution qui utilisent Python pour les scripts et Selenium Webdriver pour le framework.

Nous vous recommandons de toujours utiliser la version d'environnement d'exécution la plus récente pour vos scripts Canary, afin de pouvoir utiliser les dernières fonctionnalités et mises à jour apportées à la bibliothèque Synthetics.

Lorsque vous créez un canari, l'une des couches créées est une couche Synthetics précédée de. `Synthetics` Cette couche appartient au compte de service Synthetics et contient le code d'exécution.

> **Note**
>
> Chaque fois que vous mettez à niveau un script canary pour utiliser une nouvelle version de l'exécution Synthetics, toutes les fonctions de la bibliothèque Synthetics utilisées par votre script canary sont également automatiquement mises à niveau vers la version de NodeJS prise en charge par l'exécution Synthetics.

Rubriques

- CloudWatch Politique de support de Synthetics Runtime
- Versions d'exécution utilisant Node.js et Puppeteer

- Versions d'exécution utilisant Python et Selenium Webdriver

CloudWatch Politique de support de Synthetics Runtime

Les versions d'environnement d'exécution Synthetics sont soumises à des mises à jour de sécurité et de maintenance. Lorsqu'un composant quelconque d'une version d'exécution n'est plus pris en charge, cette version d'exécution Synthetics devient obsolète.

Vous ne pouvez pas créer des scripts Canary à l'aide de versions d'environnement d'exécution obsolètes. Les scripts Canary qui utilisent des environnements d'exécution obsolètes continuent à fonctionner. Vous pouvez arrêter, démarrer et supprimer ces scripts Canary. Vous pouvez mettre à jour un script Canary existant qui utilise une version d'exécution obsolète en mettant à jour le script Canary pour utiliser une version d'exécution prise en charge.

CloudWatch Synthetics vous avertit par e-mail si certains de vos canaris utilisent des runtimes dont la dépréciation est prévue dans les 60 prochains jours. Nous vous recommandons de migrer vos scripts Canary vers une version d'exécution prise en charge pour bénéficier des nouvelles fonctionnalités et des améliorations de la sécurité et des performances incluses dans les versions plus récentes.

Comment mettre à jour un canary vers une nouvelle version d'exécution ?

Vous pouvez mettre à jour la version d'exécution d'un Canary à l'aide de la CloudWatch console AWS CloudFormation, du SDK AWS CLI ou du AWS SDK. Lorsque vous utilisez la CloudWatch console, vous pouvez mettre à jour jusqu'à cinq canaris à la fois en les sélectionnant dans la page de liste des canaris, puis en choisissant Actions, Update Runtime.

Vous pouvez vérifier la mise à niveau en clonant d'abord le Canary à l'aide de la CloudWatch console et en mettant à jour sa version d'exécution. Cela crée un autre script Canary qui est un clone de votre script Canary d'origine. Une fois que vous avez vérifié votre script Canary avec la nouvelle version d'exécution, vous pouvez mettre à jour la version d'exécution de votre script Canary d'origine et supprimer le clone de script Canary.

Vous pouvez également mettre à jour plusieurs scripts Canary à l'aide d'un script de mise à niveau. Pour plus d’informations, consultez Script de mise à niveau de l'exécution d'un script Canary.

Si vous mettez à niveau un script Canary et qu'il échoue, consultez Dépannage d'un script Canary ayant échoué.

Dates d'obsolescence des exécutions

Version d'exécution	Date d'obsolescence
`syn-nodejs-puppeteer-6.1`	8 mars 2024
`syn-nodejs-puppeteer-6.0`	8 mars 2024
`syn-nodejs-puppeteer-5.1`	8 mars 2024
`syn-nodejs-puppeteer-5.0`	8 mars 2024
`syn-nodejs-puppeteer-4.0`	8 mars 2024
`syn-nodejs-puppeteer-3.9`	8 janvier 2024
`syn-nodejs-puppeteer-3.8`	8 janvier 2024
`syn-python-selenium-2.0`	8 mars 2024
`syn-python-selenium-1.3`	8 mars 2024
`syn-python-selenium-1.2`	8 mars 2024
`syn-python-selenium-1.1`	8 mars 2024
`syn-python-selenium-1.0`	8 mars 2024

Version d'exécution	Date d'obsolescence
`syn-nodejs-puppeteer-3.7`	8 janvier 2024
`syn-nodejs-puppeteer-3.6`	8 janvier 2024
`syn-nodejs-puppeteer-3.5`	8 janvier 2024
`syn-nodejs-puppeteer-3.4`	13 novembre 2022
`syn-nodejs-puppeteer-3.3`	13 novembre 2022
`syn-nodejs-puppeteer-3.2`	13 novembre 2022
`syn-nodejs-puppeteer-3.1`	13 novembre 2022
`syn-nodejs-puppeteer-3.0`	13 novembre 2022
`syn-nodejs-2.2`	28 mai 2021
`syn-nodejs-2.1`	28 mai 2021
`syn-nodejs-2.0`	28 mai 2021
`syn-nodejs-2.0-beta`	8 février 2021
`syn-1.0`	28 mai 2021

Script de mise à niveau de l'exécution d'un script Canary

Pour mettre à niveau un script Canary vers une version d'exécution prise en charge, utilisez le script suivant.

```
const AWS = require('aws-sdk');

// You need to configure your AWS credentials and Region.
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
credentials-node.html
//   https://docs.aws.amazon.com/sdk-for-javascript/v3/developer-guide/setting-
region.html

const synthetics = new AWS.Synthetics();

const DEFAULT_OPTIONS = {
  /**
   * The number of canaries to upgrade during a single run of this script.
   */
  count: 10,
  /**
   * No canaries are upgraded unless force is specified.
   */
  force: false
};

/**
 * The number of milliseconds to sleep between GetCanary calls when
 * verifying that an update succeeded.
 */
const SLEEP_TIME = 5000;

(async () => {
  try {
    const options = getOptions();

    const versions = await getRuntimeVersions();
    const canaries = await getAllCanaries();
    const upgrades = canaries
      .filter(canary => !versions.isLatestVersion(canary.RuntimeVersion))
      .map(canary => {
        return {
          Name: canary.Name,
          FromVersion: canary.RuntimeVersion,
```

```
          ToVersion: versions.getLatestVersion(canary.RuntimeVersion)
        };
      });

    if (options.force) {
      const promises = [];

      for (const upgrade of upgrades.slice(0, options.count)) {
        const promise = upgradeCanary(upgrade);
        promises.push(promise);
        // Sleep for 100 milliseconds to avoid throttling.
        await usleep(100);
      }

      const succeeded = [];
      const failed = [];
      for (let i = 0; i < upgrades.slice(0, options.count).length; i++) {
        const upgrade = upgrades[i];
        const promise = promises[i];
        try {
          await promise;
          console.log(`The update of ${upgrade.Name} succeeded.`);
          succeeded.push(upgrade.Name);
        } catch (e) {
          console.log(`The update of ${upgrade.Name} failed with error: ${e}`);
          failed.push({
            Name: upgrade.Name,
            Reason: e
          });
        }
      }

      if (succeeded.length) {
        console.group('The following canaries were upgraded successfully.');
        for (const name of succeeded) {
          console.log(name);
        }
        console.groupEnd()
      } else {
        console.log('No canaries were upgraded successfully.');
      }

      if (failed.length) {
        console.group('The following canaries were not upgraded successfully.');
```

```
        for (const failure of failed) {
          console.log('\x1b[31m', `${failure.Name}: ${failure.Reason}`, '\x1b[0m');
        }
        console.groupEnd();
      }
    } else {
      console.log('Run with --force [--count <count>] to perform the first <count>
 upgrades shown. The default value of <count> is 10.')
      console.table(upgrades);
    }
  } catch (e) {
    console.error(e);
  }
})();

function getOptions() {
  const force = getFlag('--force', DEFAULT_OPTIONS.force);
  const count = getOption('--count', DEFAULT_OPTIONS.count);
  return { force, count };

  function getFlag(key, defaultValue) {
    return process.argv.includes(key) || defaultValue;
  }
  function getOption(key, defaultValue) {
    const index = process.argv.indexOf(key);
    if (index < 0) {
      return defaultValue;
    }
    const value = process.argv[index + 1];
    if (typeof value === 'undefined' || value.startsWith('-')) {
      throw `The ${key} option requires a value.`;
    }
    return value;
  }
}

function getAllCanaries() {
  return new Promise((resolve, reject) => {
    const canaries = [];

    synthetics.describeCanaries().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
```

```
        if (data === null) {
          resolve(canaries);
        } else {
          canaries.push(...data.Canaries);
        }
      }
    });
  });
}

function getRuntimeVersions() {
  return new Promise((resolve, reject) => {
    const jsVersions = [];
    const pythonVersions = [];
    synthetics.describeRuntimeVersions().eachPage((err, data) => {
      if (err) {
        reject(err);
      } else {
        if (data === null) {
          jsVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          pythonVersions.sort((a, b) => a.ReleaseDate - b.ReleaseDate);
          resolve({
            isLatestVersion(version) {
              const latest = this.getLatestVersion(version);
              return latest === version;
            },
            getLatestVersion(version) {
              if (jsVersions.some(v => v.VersionName === version)) {
                return jsVersions[jsVersions.length - 1].VersionName;
              } else if (pythonVersions.some(v => v.VersionName === version)) {
                return pythonVersions[pythonVersions.length - 1].VersionName;
              } else {
                throw Error(`Unknown version ${version}`);
              }
            }
          });
        } else {
          for (const version of data.RuntimeVersions) {
            if (version.VersionName === 'syn-1.0') {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-2.')) {
              jsVersions.push(version);
            } else if (version.VersionName.startsWith('syn-nodejs-puppeteer-')) {
              jsVersions.push(version);
```

```
            } else if (version.VersionName.startsWith('syn-python-selenium-')) {
              pythonVersions.push(version);
            } else {
              throw Error(`Unknown version ${version.VersionName}`);
            }
          }
        }
      }
    });
  });
}

async function upgradeCanary(upgrade) {
  console.log(`Upgrading canary ${upgrade.Name} from ${upgrade.FromVersion} to
 ${upgrade.ToVersion}`);
  await synthetics.updateCanary({ Name: upgrade.Name, RuntimeVersion:
 upgrade.ToVersion }).promise();
  while (true) {
    await usleep(SLEEP_TIME);
    console.log(`Getting the state of canary ${upgrade.Name}`);
    const response = await synthetics.getCanary({ Name: upgrade.Name }).promise();
    const state = response.Canary.Status.State;
    console.log(`The state of canary ${upgrade.Name} is ${state}`);
    if (state === 'ERROR' || response.Canary.Status.StateReason) {
      throw response.Canary.Status.StateReason;
    }
    if (state !== 'UPDATING') {
      return;
    }
  }
}

function usleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
```

Versions d'exécution utilisant Node.js et Puppeteer

La première version d'exécution pour Node.js et Puppeteer était appelée `syn-1.0`. Les versions d'exécution ultérieures ont la convention de dénomination `syn-`*`language`*`-`*`majorversion`*`.`*`minorversion`*. Commençant par `syn-nodejs-puppeteer-3.0`, la convention de dénomination est `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*

Un suffixe `-beta` supplémentaire indique que la version d'exécution est actuellement dans une version préliminaire bêta.

Les versions d'environnement d'exécution dotées du même numéro de version majeure sont rétrocompatibles.

> **Important**
>
> Les versions d'exécution de Synthetics suivantes CloudWatch devraient être obsolètes le 8 mars 2024.
>
> - `syn-nodejs-puppeteer-6.1`
> - `syn-nodejs-puppeteer-6.0`
> - `syn-nodejs-puppeteer-5.1`
> - `syn-nodejs-puppeteer-5.0`
> - `syn-nodejs-puppeteer-4.0`
>
> Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

> **Important**
>
> IMPORTANT : Le AWS SDK inclus pour la dépendance JavaScript v2 sera supprimé et mis à jour pour utiliser le AWS SDK pour JavaScript v3 dans une future version d'exécution. À ce moment-là, vous pourrez mettre à jour les références de votre code canary. Vous pouvez également continuer à référencer et à utiliser le AWS SDK inclus pour la dépendance JavaScript v2 en l'ajoutant en tant que dépendance à votre fichier zip de code source.

Remarques concernant toutes les versions d'exécution

Lorsque vous utilisez la version d'exécution `syn-nodejs-puppeteer-3.0`, assurez-vous que votre script Canary est compatible avec Node.js 12.x. Si vous utilisez une version antérieure d'une version d'exécution `syn-nodejs`, assurez-vous que votre script est compatible avec Node.js 10.x.

Le code Lambda dans un script Canary est configuré pour avoir une mémoire maximale de 1 Go. Chaque exécution d'un script Canary expire après un délai d'attente configuré. Si aucune valeur de délai n'est spécifiée pour un canari, CloudWatch choisit une valeur de délai d'expiration en fonction

de la fréquence du canari. Si vous configurez une valeur de délai d'attente, ne dépassez pas 15 secondes pour permettre les démarrages à froid Lambda et le temps nécessaire pour démarrer l'instrumentation canary.

> **Note**
>
> Les versions d'exécution de Synthetics suivantes CloudWatch sont devenues obsolètes le 8 janvier 2024. Cela est dû au fait que le runtime Lambda Node.js 14 est devenu AWS Lambda obsolète le 4 décembre 2023.
>
> - `syn-nodejs-puppeteer-3.9`
> - `syn-nodejs-puppeteer-3.8`
> - `syn-nodejs-puppeteer-3.7`
> - `syn-nodejs-puppeteer-3.6`
> - `syn-nodejs-puppeteer-3.5`
>
> Les versions d'exécution de Synthetics suivantes CloudWatch sont devenues obsolètes le 13 novembre 2022. Cela est dû au fait que le runtime Lambda Node.js 12 est devenu AWS Lambda obsolète le 14 novembre 2022.
>
> - `syn-nodejs-puppeteer-3.4`
> - `syn-nodejs-puppeteer-3.3`
> - `syn-nodejs-puppeteer-3.2`
> - `syn-nodejs-puppeteer-3.1`
> - `syn-nodejs-puppeteer-3.0`
>
> Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

syn-nodejs-puppeteer-7,0

Le `syn-nodejs-puppeteer-7.0` runtime est la version d'exécution la plus récente du runtime Lambda Node.js 18.x. Il utilise Node.js et Puppeteer.

Dépendances principales :

- Exécution Lambda Node.js 18.x

- Version 21.9.0 de Puppeteer-core
- Version 121.0.6167.139 de Chrome

Taille du code :

La taille du code et des dépendances que vous pouvez empaqueter dans ce runtime est de 80 Mo.

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -7.0 :

- Versions mises à jour des bibliothèques intégrées dans Puppeteer et Chromium — Les dépendances de Puppeteer et Chromium sont mises à jour vers de nouvelles versions.

> **Important**
>
> Le passage de Puppeteer 19.7.0 à Puppeteer 21.9.0 introduit des modifications majeures concernant les tests et les filtres. Pour plus d'informations, consultez les sections BREAKING CHANGES dans puppeteer : v20.0.0 et puppeteer-core : v21.0.0.

Mise à niveau recommandée vers le AWS SDK v3

Le runtime Lambda nodejs18.x ne prend pas en charge le SDK v2. AWS Nous vous recommandons vivement de migrer vers le AWS SDK v3.

syn-nodejs-puppeteer-6,2

Dépendances principales :

- Exécution Lambda Node.js 18.x
- Puppeteer-core version 19.7.0
- Chromium version 111.0.5563.146

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -6.2 :

- Versions mises à jour des bibliothèques groupées dans Chromium
- Surveillance du stockage éphémère : ce moteur d'exécution ajoute une surveillance du stockage éphémère dans les comptes clients.
- Corrections de bugs

syn-nodejs-puppeteer-5,2

Le `syn-nodejs-puppeteer-5.2` runtime est la version d'exécution la plus récente du runtime Lambda Node.js 16.x. Il utilise Node.js et Puppeteer.

Dépendances principales :

- Exécution Lambda Node.js 16.x
- Puppeteer-core version 19.7.0
- Chromium version 111.0.5563.146

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -5.2 :

- Versions mises à jour des bibliothèques groupées dans Chromium
- Corrections de bugs

syn-nodejs-puppeteer-6,1

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 18.x
- Puppeteer-core version 19.7.0
- Chromium version 111.0.5563.146

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -6.1 :

- Améliorations de la stabilité : ajout d'une logique de réessai automatique pour gérer les erreurs de lancement intermittentes de Puppeteer.
- Mises à niveau des dépendances : met à niveau certains packages de dépendance tiers.
- Scripts canary sans autorisation Amazon S3 : corrections de bogues, de sorte que les scripts canary ne disposant d'aucune autorisation Amazon S3 peuvent toujours fonctionner. Ces scripts

canary ne disposant d'aucune autorisation Amazon S3 ne pourront pas télécharger de captures d'écran ou d'autres artefacts sur Amazon S3. Pour plus d'informations sur les autorisations requises pour les scripts canary, veuillez consulter la rubrique Rôles et autorisations requis pour les scripts Canary.

> **Important**
>
> IMPORTANT : Le AWS SDK inclus pour la dépendance JavaScript v2 sera supprimé et mis à jour pour utiliser le AWS SDK pour JavaScript v3 dans une future version d'exécution. À ce moment-là, vous pourrez mettre à jour les références de votre code canary. Vous pouvez également continuer à référencer et à utiliser le AWS SDK inclus pour la dépendance JavaScript v2 en l'ajoutant en tant que dépendance à votre fichier zip de code source.

syn-nodejs-puppeteer-6,0

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 18.x
- Puppeteer-core version 19.7.0
- Chromium version 111.0.5563.146

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -6.0 :

- Mise à niveau de dépendance – La dépendance Node.js est mise à niveau vers la version 18.x.
- Support du mode d'interception : le support du mode d'interception coopératif de Puppeteer a été ajouté à la bibliothèque d'exécution Synthetics Canary.
- Modification du comportement de suivi : modification du comportement de suivi par défaut pour ne suivre que les requêtes fetch et xhr, et ne pas suivre les requêtes de ressources. Vous pouvez activer le suivi des requêtes de ressources en configurant l'option `traceResourceRequests`.

- Mesure de durée affinée — La `Duration` métrique exclut désormais le temps de fonctionnement utilisé par Canary pour télécharger des artefacts, prendre des captures d'écran et générer des CloudWatch métriques. `Duration`les valeurs métriques sont signalées à CloudWatch, et vous pouvez également les consulter dans la console Synthetics.
- Correction de bogue : : vidage de mémoire généré lorsque Chromium se bloque lors d'une exécution de script canary.

> **Important**
>
> IMPORTANT : Le AWS SDK inclus pour la dépendance JavaScript v2 sera supprimé et mis à jour pour utiliser le AWS SDK pour JavaScript v3 dans une future version d'exécution. À ce moment-là, vous pourrez mettre à jour les références de votre code canary. Vous pouvez également continuer à référencer et à utiliser le AWS SDK inclus pour la dépendance JavaScript v2 en l'ajoutant en tant que dépendance à votre fichier zip de code source.

syn-nodejs-puppeteer-5,1

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 16.x
- Puppeteer-core version 19.7.0
- Chromium version 111.0.5563.146

Corrections de bugs dans syn-nodejs-puppeteer -5.1 :

- Correction de bogue – Cette exécution corrige un bogue dans `syn-nodejs-puppeteer-5.0` où les fichiers HAR créés par les scripts canary n'avaient pas d'en-têtes de requête.

syn-nodejs-puppeteer-5,0

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 16.x
- Puppeteer-core version 19.7.0
- Chromium version 111.0.5563.146

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -5.0 :

- Mise à niveau des dépendances – La version Puppeteer-core est mise à jour vers la version 19.7.0. La version Chromium est mise à niveau vers la version 111.0.5563.146.

> **Important**
>
> La nouvelle version de Puppeteer-core n'est pas totalement rétrocompatible avec les versions précédentes de Puppeteer. Certaines modifications apportées à cette version peuvent entraîner l'échec des scripts canary existants qui utilisent des fonctions Puppeteer obsolètes. Pour plus d'informations, consultez les derniers changements pour les versions 19.7.0 à 6.0 de Puppeteer-core dans les journaux des modifications de Puppeteer.

syn-nodejs-puppeteer-4,0

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 16.x
- Puppeteer-core version 5.5.0
- Chromium version 92.0.4512

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -4.0 :

- Mise à niveau de dépendance – La dépendance Node.js est mise à niveau vers la version 16.x.

Runtimes obsolètes pour Node.js et Puppeteer

Les environnements d'exécution suivants pour Node.js et Puppeteer sont devenus obsolètes.

syn-nodejs-puppeteer-3,9

> **Important**
>
> Cette version d'exécution est devenue obsolète le 8 janvier 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Moteur d'exécution Lambda Node.js 14.x
- Puppeteer-core version 5.5.0
- Chromium version 92.0.4512

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.9 :

- Mises à niveau des dépendances : met à niveau certains packages de dépendance tiers.

syn-nodejs-puppeteer-3,8

> **Important**
>
> Cette version d'exécution est devenue obsolète le 8 janvier 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Moteur d'exécution Lambda Node.js 14.x
- Puppeteer-core version 5.5.0
- Chromium version 92.0.4512

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.8 :

- Nettoyage des profils : les profils Chromium sont désormais nettoyés après chaque exécution de canary.

Corrections de bugs dans syn-nodejs-puppeteer -3.8 :

- Correction de bogues : auparavant, les canaris de surveillance visuelle cessaient parfois de fonctionner correctement après une exécution sans captures d'écran. Ceci est maintenant corrigé.

syn-nodejs-puppeteer-3,7

> **Important**
>
> Cette version d'exécution est devenue obsolète le 8 janvier 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Moteur d'exécution Lambda Node.js 14.x
- Puppeteer-core version 5.5.0
- Chromium version 92.0.4512

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.7 :

- Amélioration de la journalisation : le canary chargera les journaux sur Amazon S3 même en cas d'expiration ou de panne.
- Taille de la couche Lambda réduite : la taille de la couche Lambda utilisée pour les scripts canary est réduite de 34 %.

Corrections de bugs dans syn-nodejs-puppeteer -3.7 :

- Correction de bogues : les polices japonaises, chinoises simplifiées et chinoises traditionnelles seront restituées correctement.

syn-nodejs-puppeteer-3,6

> **Important**
>
> Cette version d'exécution est devenue obsolète le 8 janvier 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Moteur d'exécution Lambda Node.js 14.x
- Puppeteer-core version 5.5.0
- Chromium version 92.0.4512

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.6 :

- Horodatages plus précis— L'heure de début et l'heure de fin des scripts Canary sont désormais précises à la milliseconde près.

syn-nodejs-puppeteer-3,5

> **Important**
>
> Cette version d'exécution est devenue obsolète le 8 janvier 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Moteur d'exécution Lambda Node.js 14.x
- Puppeteer-core version 5.5.0
- Chromium version 92.0.4512

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.5 :

- Updated Dependencies (Dépendances mises à jour) : les seules nouvelles fonctions de ce moteur d'exécution sont les dépendances mises à jour.

syn-nodejs-puppeteer-3,4

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 13 novembre 2022. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 12.x
- Puppeteer-core version 5.5.0
- Chromium version 88.0.4298.0

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.4 :

- Fonction de gestion personnalisée : vous pouvez désormais utiliser une fonction de gestion personnalisée pour vos scripts Canary. Les exécutions précédentes nécessitaient que le point d'entrée du script comprenne `.handler`.

  Vous pouvez également placer des scripts Canary dans n'importe quel dossier et transmettre le nom du dossier dans le gestionnaire. Par exemple, `MyFolder/MyScriptFile.functionname` peut être utilisé comme point d'entrée.
- Informations sur les fichiers HAR étendus : vous pouvez désormais voir des demandes erronées, en attente et incomplètes dans les fichiers HAR produits par les scripts canary.

syn-nodejs-puppeteer-3,3

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 13 novembre 2022. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 12.x
- Puppeteer-core version 5.5.0
- Chromium version 88.0.4298.0

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.3 :

- Options supplémentaires pour le chiffrement des artefacts : pour les canaris utilisant ce moteur d'exécution ou une version ultérieure, au lieu d'utiliser une clé AWS gérée pour chiffrer les artefacts stockés dans Amazon S3, vous pouvez choisir d'utiliser une clé gérée par le AWS KMS client ou une clé gérée par Amazon S3. Pour plus d'informations, consultez Chiffrement des artefacts de script Canary.

syn-nodejs-puppeteer-3,2

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 13 novembre 2022. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 12.x
- Puppeteer-core version 5.5.0
- Chromium version 88.0.4298.0

Nouvelles fonctionnalités de la version syn-nodejs-puppeteer -3.2 :

- Surveillance visuelle avec des captures d'écran : les scripts Canary qui utilisent cette exécution ou une version ultérieure peuvent comparer une capture d'écran prise lors d'une exécution à une version de référence de la même capture d'écran. Si les différences entre les captures d'écran dépassent un seuil de pourcentage spécifié, le script Canary échoue. Pour plus d'informations, consultez Surveillance visuelle ou Modèle de surveillance visuelle.

- Nouvelles fonctions concernant les données sensibles : vous pouvez empêcher l'apparition de données sensibles dans les journaux et rapports des scripts Canary. Pour plus d'informations, consultez SyntheticsLogHelper classe.
- Fonction obsolète : la classe `RequestResponseLogHelper` est rendue obsolète au profit de nouvelles options de configuration. Pour plus d'informations, consultez RequestResponseLogHelper classe.

syn-nodejs-puppeteer-3,1

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 13 novembre 2022. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 12.x
- Puppeteer-core version 5.5.0
- Chromium version 88.0.4298.0

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -3.1 :

- Possibilité de configurer CloudWatch les métriques — Avec cet environnement d'exécution, vous pouvez désactiver les métriques dont vous n'avez pas besoin. Sinon, les canaris publient différentes CloudWatch statistiques pour chaque course canari.
- Lien de capture d'écran : vous pouvez lier une capture d'écran à une étape de script Canary une fois l'étape terminée. Pour ce faire, vous devez faire une capture d'écran à l'aide de la méthode takeScreenshot, en utilisant le nom de l'étape à laquelle vous souhaitez associer la capture d'écran. Par exemple, vous pouvez réaliser une étape, ajouter un temps d'attente, puis prendre la capture d'écran.
- Le plan de surveillance du rythme cardiaque peut surveiller plusieurs URL : vous pouvez utiliser le plan de surveillance du rythme cardiaque de la CloudWatch console pour surveiller plusieurs URL et voir le statut, la durée, les captures d'écran associées et la raison de l'échec de chaque URL dans le résumé des étapes du rapport Canary Run.

syn-nodejs-puppeteer-3,0

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 13 novembre 2022. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 12.x
- Puppeteer-core version 5.5.0
- Chromium version 88.0.4298.0

Nouvelles fonctionnalités de la syn-nodejs-puppeteer version -3.0 :

- Dépendances mises à niveau : cette version utilise Puppeteer version 5.5.0, Node.js 12.x et Chromium 88.0.4298.0.
- Accès entre régions au compartiment : vous pouvez désormais spécifier un compartiment S3 dans une autre région comme compartiment où votre script Canary stocke ses fichiers journaux, ses captures d'écran et ses fichiers HAR.
- Nouvelles fonctions disponibles : cette version ajoute des fonctions de bibliothèque pour récupérer le nom du script Canary et la version d'exécution de Synthetics.

  Pour plus d'informations, consultez Classe Synthetics.

syn-nodejs-2.2

Cette section contient des informations sur la version d'exécution `syn-nodejs-2.2`.

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 28 mai 2021. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 10.x
- Puppeteer-core version 3.3.0
- Chromium version 83.0.4103.0

Nouvelles fonctions de syn-nodejs-2.2 :

- Surveillance des scripts Canary en tant qu'étapes HTTP : vous pouvez maintenant tester plusieurs API dans un seul script Canary. Chaque API est testée en tant qu'étape HTTP distincte, et CloudWatch Synthetics surveille le statut de chaque étape à l'aide de métriques et du rapport d'étape de CloudWatch Synthetics. CloudWatch Synthetics `SuccessPercent` crée des statistiques pour `Duration` chaque étape HTTP.

  Cette fonctionnalité est implémentée par la fonction executeHttpStep(StepName, RequestOptions, callback, StepConfig). Pour plus d'informations, consultez executeHttpStep(StepName, RequestOptions, [rappel], [StepConfig]).

  Le modèle de script Canary d'API est mis à jour pour utiliser cette nouvelle fonction.
- Rapports sur les requêtes HTTP : vous pouvez maintenant afficher des rapports détaillés sur les requêtes HTTP qui capturent des détails tels que les en-têtes de demande/réponse, le corps de la réponse, le code d'état, les minutages d'erreur et de performance, le temps de connexion TCP, le temps de liaison TLS, le temps du premier octet et le temps de transfert de contenu. Toutes les requêtes HTTP qui utilisent le module HTTP/HTTPS sous le capot sont capturées ici. Les en-têtes et le corps de la réponse ne sont par défaut pas capturés, mais cette option peut être activée en définissant les options de configuration.
- Configuration globale et au niveau des étapes — Vous pouvez définir des configurations CloudWatch Synthetics au niveau global, qui sont appliquées à toutes les étapes des canaris. Vous pouvez également remplacer ces configurations au niveau de l'étape en appliquant des paires clé/valeur de configuration pour activer ou désactiver certaines options.

  Pour plus d'informations, consultez SyntheticsConfiguration classe.
- Configuration pour continuer après un échec de l'étape : vous pouvez choisir de poursuivre l'exécution des scripts Canary en cas d'échec d'une étape. Pour la fonction `executeHttpStep`, cette option est activée par défaut. Vous pouvez définir cette option une fois au niveau global ou la définir différemment par étape.

syn-nodejs-2.1

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 28 mai 2021. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 10.x
- Puppeteer-core version 3.3.0
- Chromium version 83.0.4103.0

Nouvelles fonctions de syn-nodejs-2.1 :

- Comportement de capture d'écran configurable : offre la possibilité de désactiver la prise de captures d'écran par les scripts Canary d'interface utilisateur. Dans les scripts Canary qui utilisent des versions précédentes des exécutions, les scripts Canary d'interface utilisateur effectuent toujours des captures d'écran avant et après chaque étape. Avec `syn-nodejs-2.1`, ce comportement peut être configuré. La désactivation des captures d'écran peut réduire vos coûts de stockage Amazon S3 et vous aider à vous conformer aux réglementations HIPAA. Pour plus d'informations, consultez SyntheticsConfiguration classe.
- Personnalisation des paramètres de lancement de Google Chrome : vous pouvez désormais configurer les arguments utilisés lorsqu'un script Canary lance une fenêtre de navigateur Google Chrome. Pour plus d'informations, consultez launch(options).

Il peut y avoir une légère augmentation de la durée des scripts Canary lors de l'utilisation de syn-nodejs-2.0 ou d'une version ultérieure, par rapport aux versions antérieures des exécutions des scripts Canary.

syn-nodejs-2.0

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 28 mai 2021. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 10.x
- Puppeteer-core version 3.3.0
- Chromium version 83.0.4103.0

Nouvelles fonctions de syn-nodejs-2.0 :

- Dépendances mises à niveau : cette version d'exécution utilise Puppeteer-core version 3.3.0 et Chromium version 83.0.4103.0
- Prise en charge du suivi actif X-Ray. Lorsque le suivi est activé sur un canary, des traces X-Ray sont envoyées pour tous les appels effectués par le canari qui utilisent le navigateur, le AWS SDK ou les modules HTTP ou HTTPS. Les scripts canary dont le suivi est activé apparaissent sur la carte de suivi X-Ray, même s'ils n'envoient pas de demandes à d'autres services ou applications dont le suivi est activé. Pour plus d'informations, consultez Scripts Canary et suivi X-Ray.
- Rapports synthétiques — Pour chaque course à Canary, CloudWatch Synthetics crée un rapport `SyntheticsReport-PASSED.json` nommé `SyntheticsReport-FAILED.json` ou qui enregistre des données telles que l'heure de début, l'heure de fin, le statut et les échecs. Il enregistre également l'état PASSED/FAILED de chaque étape du script Canary ainsi que les échecs et les captures d'écran réalisées pour chaque étape.
- Rapport du vérificateur des liens rompus : la nouvelle version du vérificateur de liens rompus inclus dans cette exécution crée un rapport qui inclut les liens qui ont été vérifiés, le code d'état, la raison de l'échec (le cas échéant) et les captures d'écran de la page source et de destination.
- Nouvelles CloudWatch métriques — Synthetics publie les métriques `2xx` nommées`4xx`,`5xx`, `RequestFailed` et dans l'espace de noms. `CloudWatchSynthetics` Ces métriques indiquent le nombre d'échecs 200, 400, 500 et de demande dans les exécutions de scripts Canary. Avec cette version d'exécution, ces métriques sont signalées uniquement pour les scripts Canary d'interface utilisateur et ne sont pas signalées pour les scripts Canary d'API. Elles sont également signalées pour les scripts Canary d'API commençant par la version d'exécution `syn-nodejs-puppeteeer-2.2`.
- Fichiers HAR triables : vous pouvez maintenant trier vos fichiers HAR par code d'état, taille de la demande et durée.
- Horodatage des métriques : CloudWatch les métriques sont désormais signalées en fonction de l'heure d'invocation de Lambda plutôt que de l'heure de fin de l'exécution de Canary.

Corrections de bogues dans syn-nodejs-2.0 :

- Correction du problème lors duquel les erreurs de téléchargement d'artefacts des scripts Canary n'étaient pas signalées. Ces erreurs apparaissent maintenant comme des erreurs d'exécution.
- Correction du problème lors duquel des demandes redirigées (3xx) étaient incorrectement journalisées en tant qu'erreurs.
- Correction du problème lors duquel les captures d'écran étaient numérotées à partir de 0. Elles devraient maintenant commencer par 1.
- Correction du problème lors duquel des captures d'écran étaient brouillées pour les polices chinoises et japonaises.

Il peut y avoir une légère augmentation de la durée des scripts Canary lors de l'utilisation de syn-nodejs-2.0 ou d'une version ultérieure, par rapport aux versions antérieures des exécutions des scripts Canary.

syn-nodejs-2.0-beta

> **Important**
>
> Cette version d'exécution a été rendue obsolète le 8 février 2021. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Exécution Lambda Node.js 10.x
- Puppeteer-core version 3.3.0
- Chromium version 83.0.4103.0

Nouvelles fonctions de syn-nodejs-2.0-beta :

- Dépendances mises à niveau : cette version d'exécution utilise Puppeteer-core version 3.3.0 et Chromium version 83.0.4103.0
- Rapports synthétiques — Pour chaque course à Canary, CloudWatch Synthetics crée un rapport `SyntheticsReport-PASSED.json` nommé `SyntheticsReport-FAILED.json` ou qui enregistre des données telles que l'heure de début, l'heure de fin, le statut et les échecs. Il

enregistre également l'état PASSED/FAILED de chaque étape du script Canary ainsi que les échecs et les captures d'écran réalisées pour chaque étape.

- Rapport du vérificateur des liens rompus : la nouvelle version du vérificateur de liens rompus inclus dans cette exécution crée un rapport qui inclut les liens qui ont été vérifiés, le code d'état, la raison de l'échec (le cas échéant) et les captures d'écran de la page source et de destination.
- Nouvelles CloudWatch métriques — Synthetics publie les métriques `2xx` nommées`4xx`,`5xx`, `RequestFailed` et dans l'espace de noms. `CloudWatchSynthetics` Ces métriques indiquent le nombre d'échecs 200, 400, 500 et de demande dans les exécutions de scripts Canary. ACes métriques sont signalées uniquement pour les scripts Canary d'interface utilisateur et ne sont pas signalées pour les scripts Canary d'API.
- Fichiers HAR triables : vous pouvez maintenant trier vos fichiers HAR par code d'état, taille de la demande et durée.
- Horodatage des métriques : CloudWatch les métriques sont désormais signalées en fonction de l'heure d'invocation de Lambda plutôt que de l'heure de fin de l'exécution de Canary.

Corrections de bogues dans syn-nodejs-2.0-beta :

- Correction du problème lors duquel les erreurs de téléchargement d'artefacts des scripts Canary n'étaient pas signalées. Ces erreurs apparaissent maintenant comme des erreurs d'exécution.
- Correction du problème lors duquel des demandes redirigées (3xx) étaient incorrectement journalisées en tant qu'erreurs.
- Correction du problème lors duquel les captures d'écran étaient numérotées à partir de 0. Elles devraient maintenant commencer par 1.
- Correction du problème lors duquel des captures d'écran étaient brouillées pour les polices chinoises et japonaises.

syn-1.0

> **Important**
>
> L'obsolescence de cette version d'exécution est prévue pour le 28 mai 2021. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

La première version d'exécution de Synthetics est `syn-1.0`.

Dépendances principales :

- Exécution Lambda Node.js 10.x
- Puppeteer-core version 1.14.0
- La version Chromium qui correspond à Puppeteer-core 1.14.0

Versions d'exécution utilisant Python et Selenium Webdriver

Les sections suivantes contiennent des informations sur les versions d' CloudWatch exécution de Synthetics pour Python et Selenium Webdriver. Selenium est un outil d'automatisation de navigateur open source. Pour de plus amples informations sur Selenium, consultez www.selenium.dev/.

La convention de dénomination pour ces versions d'exécution est `syn-`*`language`*`-`*`framework`*`-`*`majorversion`*`.`*`minorversion`*.

> **Important**
>
> Les versions d'exécution de Synthetics suivantes CloudWatch devraient être obsolètes le 8 mars 2024.
>
> - `syn-python-selenium-2.0`
> - `syn-python-selenium-1.3`
> - `syn-python-selenium-1.2`
> - `syn-python-selenium-1.1`
> - `syn-python-selenium-1.0`
>
> Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

syn-python-selenium-3,0

La version 3.0 est le dernier runtime CloudWatch Synthetics pour Python et Selenium.

Dépendances principales :

- Python 3.8
- Sélénium 4.15.1
- Version 121.0.6167.139 de Chrome

Nouvelles fonctionnalités de la syn-python-selenium version -3.0 :

- Versions mises à jour des bibliothèques groupées dans Chromium — La dépendance à Chromium est mise à jour vers une nouvelle version.

syn-python-selenium-2,1

Dépendances principales :

- Python 3.8
- Sélénium 4.15.1
- Chromium version 111.0.5563.146

Nouvelles fonctionnalités de la syn-python-selenium version -2.1 :

- Versions mises à jour des bibliothèques groupées dans Chromium — Les dépendances Chromium et Selenium sont mises à jour vers de nouvelles versions.

syn-python-selenium-2,0

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Python 3.8
- Selenium 4.10.0
- Chromium version 111.0.5563.146

Nouvelles fonctionnalités de la syn-python-selenium version -2.0 :

- Dépendances mises à jour – Les dépendances de Chromium et Selenium ont été mises à jour vers les nouvelles versions.

Corrections de bugs dans syn-python-selenium -2.0 :

- Horodatage ajouté – Un horodatage a été ajouté aux journaux de scripts canary.
- Réutilisation des sessions – Un bogue a été corrigé pour empêcher les scripts canary de réutiliser la session de leur précédente exécution.

syn-python-selenium-1,3

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Python 3.8
- Selenium 3.141.0
- Chromium version 92.0.4512.0

Nouvelles fonctionnalités de la version syn-python-selenium -1.3 :

- Horodatages plus précis— L'heure de début et l'heure de fin des scripts Canary sont désormais précises à la milliseconde près.

syn-python-selenium-1,2

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Python 3.8
- Selenium 3.141.0

- Chromium version 92.0.4512.0

- Updated Dependencies (Dépendances mises à jour) : les seules nouvelles fonctions de ce moteur d'exécution sont les dépendances mises à jour.

syn-python-selenium-1,1

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Python 3.8
- Selenium 3.141.0
- Chromium version 83.0.4103.0

Fonctionnalités :

- Fonction de gestion personnalisée : vous pouvez désormais utiliser une fonction de gestion personnalisée pour vos scripts canary. Les exécutions précédentes nécessitaient que le point d'entrée du script comprenne `.handler`.

  Vous pouvez également placer des scripts Canary dans n'importe quel dossier et transmettre le nom du dossier dans le gestionnaire. Par exemple, `MyFolder/MyScriptFile.functionname` peut être utilisé comme point d'entrée.
- Options de configuration pour ajouter des métriques et des configurations d'échec d'étapes : Ces options étaient déjà disponibles dans les exécutions pour les scripts canary Node.js. Pour plus d'informations, consultez SyntheticsConfiguration classe.
- Arguments personnalisés dans Chrome : vous pouvez désormais ouvrir un navigateur en mode navigation privée ou passer en configuration de serveur proxy. Pour plus d'informations, consultez Chrome().
- Compartiments d'artefacts entre Régions : un script Canary peut stocker ses artefacts dans un compartiment Simple Storage Service (Amazon S3) dans une autre Région.

- Corrections de bogue, y compris un correctif pour le problème **`index.py`** : avec les exécutions précédentes, un fichier Canary nommé `index.py` a provoqué des exceptions car il entrait en conflit avec le nom du fichier de bibliothèque. Le problème a été résolu.

syn-python-selenium-1,0

> **Important**
>
> La mise hors service de cette version d'exécution est prévue pour le 8 mars 2024. Pour plus d'informations, consultez CloudWatch Politique de support de Synthetics Runtime.

Dépendances principales :

- Python 3.8
- Selenium 3.141.0
- Chromium version 83.0.4103.0

Fonctionnalités :

- Prise en charge de Selenium : vous pouvez écrire des scripts Canary à l'aide du cadre de test Selenium. Vous pouvez importer vos scripts Selenium d'autres sources dans CloudWatch Synthetics avec un minimum de modifications, et ils fonctionneront avec les services. AWS

Écriture d'un script Canary

Les sections suivantes expliquent comment écrire un script Canary et comment intégrer un script Canary à d'autres AWS services ainsi qu'à des dépendances et bibliothèques externes.

Rubriques

- Écriture d'un script Canary Node.js
- Écriture d'un script Canary Python
- Modification d'un script Selenium existant pour l'utiliser en tant que script Canary Synthetics
- Modification d'un script Puppeteer Synthetics existant pour authentifier des certificats non standard

Écriture d'un script Canary Node.js

Rubriques

Création d'un CloudWatch canari Synthetics à partir de zéro

Voici un exemple minimal de script Canary Synthetics. Ce script est transmis comme une exécution réussie et renvoie une chaîne. Pour voir à quoi ressemble un script Canary qui a échoué, remplacez `let fail = false;` par `let fail = true;`.

Vous devez définir une fonction de point d'entrée pour le script Canary. Pour voir comment les fichiers sont téléchargés à l'emplacement Amazon S3 spécifié en tant que `ArtifactS3Location` du script Canary, créez-les dans le dossier /tmp. Une fois le script exécuté, le statut de réussite/d'échec et les métriques de durée sont publiés sur CloudWatch et les fichiers situés sous /tmp sont téléchargés dans S3.

```
const basicCustomEntryPoint = async function () {

    // Insert your code here

    // Perform multi-step pass/fail check

    // Log decisions made and results to /tmp

    // Be sure to wait for all your code paths to complete
    // before returning control back to Synthetics.
    // In that way, your canary will not finish and report success
    // before your code has finished executing

    // Throw to fail, return to succeed
    let fail = false;
    if (fail) {
        throw "Failed basicCanary check.";
```

```
    }

    return "Successfully completed basicCanary checks.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Ensuite, nous allons développer le script pour utiliser la journalisation Synthetics et passer un appel à l'aide du SDK. AWS À des fins de démonstration, ce script créera un client Amazon DynamoDB et réalisera un appel à l'API ListTables DynamoDB. Il enregistre la réponse à la demande et les journaux réussissent ou échouent selon le succès de la demande.

```
const log = require('SyntheticsLogger');
const AWS = require('aws-sdk');
// Require any dependencies that your script needs
// Bundle additional files and dependencies into a .zip file with folder structure
// nodejs/node_modules/additional files and folders

const basicCustomEntryPoint = async function () {

    log.info("Starting DynamoDB:listTables canary.");

    let dynamodb = new AWS.DynamoDB();
    var params = {};
    let request = await dynamodb.listTables(params);
    try {
        let response = await request.promise();
        log.info("listTables response: " + JSON.stringify(response));
    } catch (err) {
        log.error("listTables error: " + JSON.stringify(err), err.stack);
        throw err;
    }

    return "Successfully completed DynamoDB:listTables canary.";
};

exports.handler = async () => {
    return await basicCustomEntryPoint();
};
```

Empaqueter vos fichiers Canary Node.js

Si vous chargez vos scripts canary à l'aide d'un emplacement Simple Storage Service (Amazon S3), votre fichier zip doit inclure votre script sous la structure de dossier suivante : `nodejs/node_modules/`*`myCanaryFilename.js file`*.

Si vous avez plusieurs fichiers `.js` ou une dépendance dont dépend votre script, vous pouvez les regrouper dans un seul fichier ZIP contenant la structure de dossiers `nodejs/node_modules/`*`myCanaryFilename.js file and other folders and files`*. Si vous utilisez `syn-nodejs-puppeteer-3.4` ou une version ultérieure, vous pouvez éventuellement placer vos fichiers canary dans un autre dossier et créer votre structure de dossiers comme suit : `nodejs/node_modules/`*`myFolder`*`/`*`myCanaryFilename.js file and other folders and files`*.

Nom du gestionnaire

Assurez-vous de définir le point d'entrée (gestionnaire) de votre script Canary de sorte que `myCanaryFilename.functionName` corresponde au nom du fichier du point d'entrée de votre script. Si vous utilisez une exécution antérieure à `syn-nodejs-puppeteer-3.4`, alors `functionName` doit être `handler`. Si vous utilisez `syn-nodejs-puppeteer-3.4` ou une version ultérieure, vous pouvez choisir n'importe quel nom de fonction comme gestionnaire. Si vous utilisez `syn-nodejs-puppeteer-3.4` ou une version ultérieure, vous pouvez également stocker le script canary dans un dossier séparé tel que `nodejs/node_modules/myFolder/my_canary_filename`. Si vous le stockez dans un dossier séparé, spécifiez ce chemin dans le point d'entrée de votre script, tel que `myFolder/my_canary_filename.functionName`.

Modification d'un script Puppeteer existant pour l'utiliser en tant que script Canary Synthetics

Cette section explique comment prendre des scripts Puppeteer, pour ensuite les modifier afin qu'ils s'exécutent en tant que scripts Canary Synthetics. Pour de plus amples informations sur Puppeteer, consultez Puppeteer API v1.14.0.

Nous allons commencer avec cet exemple de script Puppeteer :

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await page.screenshot({path: 'example.png'});
```

```
  await browser.close();
})();
```

Les étapes de conversion sont les suivantes :

- Créez et exportez une fonction `handler`. Le gestionnaire est la fonction de point d'entrée du script. Si vous utilisez une exécution antérieure à `syn-nodejs-puppeteer-3.4`, la fonction du gestionnaire doit être nommée `handler`. Si vous utilisez `syn-nodejs-puppeteer-3.4` ou une version ultérieure, la fonction peut porter n'importe quel nom, mais il doit être identique à celui utilisé dans le script. De plus, si vous utilisez `syn-nodejs-puppeteer-3.4` ou une version ultérieure, vous pouvez stocker vos scripts dans n'importe quel dossier et spécifier ce dossier dans le nom du gestionnaire.

  ```
  const basicPuppeteerExample = async function () {};

  exports.handler = async () => {
      return await basicPuppeteerExample();
  };
  ```

- Utilisez la dépendance `Synthetics`.

  ```
  var synthetics = require('Synthetics');
  ```

- Utilisez la fonction `Synthetics.getPage` pour obtenir un objet `Page` Puppeteer.

  ```
  const page = await synthetics.getPage();
  ```

  L'objet page renvoyé par la fonction Synthetics.getPage présente les événements page.on, `request`, `response` et `requestfailed` instrumentés pour la journalisation. Synthetics configure également la génération de fichiers HAR pour les demandes et les réponses sur la page, et ajoute l'ARN du script Canary aux en-têtes des demandes sortantes de l'agent utilisateur sur la page.

Le script est maintenant prêt à être exécuté en tant que script Canary Synthetics. Voici le script mis à jour :

```
var synthetics = require('Synthetics');  // Synthetics dependency

const basicPuppeteerExample = async function () {
```

```
    const page = await synthetics.getPage(); // Get instrumented page from Synthetics
    await page.goto('https://example.com');
    await page.screenshot({path: '/tmp/example.png'}); // Write screenshot to /tmp
 folder
};

exports.handler = async () => {  // Exported handler function
    return await basicPuppeteerExample();
};
```

Variables d’environnement

Vous pouvez utiliser des variables d'environnement lorsque vous créez des scripts Canary. Cela vous permet d'écrire un seul script Canary, puis d'utiliser ce script avec des valeurs différentes pour créer rapidement plusieurs scripts Canary ayant une tâche similaire.

Par exemple, supposons que votre organisation dispose de points de terminaison tels que `prod`, `dev` et `pre-release` pour les différentes étapes de votre développement logiciel et que vous devez créer des scripts Canary pour tester chacun de ces points de terminaison. Vous pouvez écrire un seul script Canary qui teste votre logiciel, puis spécifier différentes valeurs pour la variable d'environnement de point de terminaison lorsque vous créez chacun des trois scripts Canary. Ensuite, lorsque vous créez un script Canary, vous spécifiez le script et les valeurs à utiliser pour les variables d'environnement.

Les noms des variables d'environnement peuvent contenir des lettres, des chiffres et le caractère de soulignement. Ils doivent commencer par une lettre et comporter au moins deux caractères. La taille totale de vos variables d'environnement ne peut pas dépasser 4 Ko. Vous ne pouvez pas spécifier de variables d'environnement réservées Lambda comme noms pour vos variables d'environnement. Pour plus d'informations sur les variables d'environnement réservées, consultez Variables d'environnement d'exécution.

> **Important**
>
> Les clés et les valeurs des variables d'environnement ne sont pas chiffrées. Ne stockez pas d'informations sensibles dans celles-ci.

L'exemple de script suivant utilise deux variables d'environnement. Ce script est destiné à un script Canary qui vérifie si une page web est disponible. Il utilise des variables d'environnement pour paramétrer à la fois l'URL qu'il vérifie et le niveau de journal Synthetics qu'il utilise CloudWatch .

La fonction suivante définit `LogLevel` sur la valeur de la variable d'environnement `LOG_LEVEL`.

```
 synthetics.setLogLevel(process.env.LOG_LEVEL);
```

Cette fonction définit `URL` sur la valeur de la variable d'environnement `URL`.

```
const URL = process.env.URL;
```

C'est le script complet. Lorsque vous créez un script Canary à l'aide de ce script, vous spécifiez les valeurs pour les variables d'environnement `LOG_LEVEL` et `URL`.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadEnvironmentVariable = async function () {

    // Setting the log level (0-3)
    synthetics.setLogLevel(process.env.LOG_LEVEL);
    // INSERT URL here
    const URL = process.env.URL;

    let page = await synthetics.getPage();
    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
    }
    //Wait for page to render.
    //Increase or decrease wait time based on endpoint being monitored.
    await page.waitFor(15000);
    await synthetics.takeScreenshot('loaded', 'loaded');
    let pageTitle = await page.title();
    log.info('Page title: ' + pageTitle);
    log.debug('Environment variable:' + process.env.URL);

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};
```

```
exports.handler = async () => {
    return await pageLoadEnvironmentVariable();
};
```

Transmission de variables d'environnement à votre script

Pour transmettre des variables d'environnement à votre script lorsque vous créez un script Canary dans la console, spécifiez les clés et les valeurs des variables d'environnement dans la section Environment variables (Variables d'environnement) de la console. Pour plus d'informations, consultez Création d'un Canary.

Pour transmettre des variables d'environnement via l'API AWS CLI, utilisez le `EnvironmentVariables` paramètre de la `RunConfig` section. Voici un exemple de AWS CLI commande qui crée un canari qui utilise deux variables d'environnement avec les clés de `Environment` et`Region`.

```
aws synthetics create-canary --cli-input-json '{
   "Name":"nameofCanary",
   "ExecutionRoleArn":"roleArn",
   "ArtifactS3Location":"s3://cw-syn-results-123456789012-us-west-2",
   "Schedule":{
      "Expression":"rate(0 minute)",
      "DurationInSeconds":604800
   },
   "Code":{
      "S3Bucket": "canarycreation",
      "S3Key": "cwsyn-mycanaryheartbeat-12345678-d1bd-1234-
abcd-123456789012-12345678-6a1f-47c3-b291-123456789012.zip",
      "Handler":"pageLoadBlueprint.handler"
   },
   "RunConfig": {
      "TimeoutInSeconds":60,
      "EnvironmentVariables": {
         "Environment":"Production",
         "Region": "us-west-1"
      }
   },
   "SuccessRetentionPeriodInDays":13,
   "FailureRetentionPeriodInDays":13,
   "RuntimeVersion":"syn-nodejs-2.0"
}'
```

Intégrer votre Canary à d'autres AWS services

Tous les canaris peuvent utiliser la bibliothèque du AWS SDK. Vous pouvez utiliser cette bibliothèque lorsque vous écrivez votre canari pour intégrer le canari à d'autres AWS services.

Pour ce faire, vous devez ajouter le code suivant à votre script canary. Pour ces exemples, AWS Secrets Manager il est utilisé comme service auquel le canari s'intègre.

- Importez le AWS SDK.

```
const AWS = require('aws-sdk');
```

- Créez un client pour le AWS service auquel vous souhaitez procéder à l'intégration.

```
const secretsManager = new AWS.SecretsManager();
```

- Utilisez le client pour effectuer des appels d'API vers ce service.

```
var params = {
  SecretId: secretName
};
return await secretsManager.getSecretValue(params).promise();
```

L'extrait de code de script Canary suivant illustre une intégration à Secrets Manager de manière plus détaillée.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const AWS = require('aws-sdk');
const secretsManager = new AWS.SecretsManager();

const getSecrets = async (secretName) => {
    var params = {
        SecretId: secretName
    };
    return await secretsManager.getSecretValue(params).promise();
}

const secretsExample = async function () {
    let URL = "<URL>";
```

```
    let page = await synthetics.getPage();

    log.info(`Navigating to URL: ${URL}`);
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});

    // Fetch secrets
    let secrets = await getSecrets("secretname")

    /**
    * Use secrets to login.
    *
    * Assuming secrets are stored in a JSON format like:
    * {
    *   "username": "<USERNAME>",
    *   "password": "<PASSWORD>"
    * }
    **/
    let secretsObj = JSON.parse(secrets.SecretString);
    await synthetics.executeStep('login', async function () {
        await page.type(">USERNAME-INPUT-SELECTOR<", secretsObj.username);
        await page.type(">PASSWORD-INPUT-SELECTOR<", secretsObj.password);

        await Promise.all([
          page.waitForNavigation({ timeout: 30000 }),
          await page.click(">SUBMIT-BUTTON-SELECTOR<")
        ]);
    });

    // Verify login was successful
    await synthetics.executeStep('verify', async function () {
        await page.waitForXPath(">SELECTOR<", { timeout: 30000 });
    });
};

exports.handler = async () => {
    return await secretsExample();
};
```

Utilisation forcée d'une adresse IP statique par le script Canary

Vous pouvez configurer un script Canary afin qu'il utilise une adresse IP statique.

Pour forcer un script Canary à utiliser une adresse IP statique

1. Créez un nouveau VPC. Pour plus d'informations, consultez Utilisation de DNS avec votre VPC.
2. Créez une nouvelle passerelle Internet. Pour plus d'informations, consultez Ajout d'une passerelle Internet à votre VPC.
3. Créez un sous-réseau public dans votre nouveau VPC.
4. Ajoutez une nouvelle table de routage au VPC.
5. Ajoutez un routage dans la nouvelle table de routage, qui va de `0.0.0.0/0` à la passerelle Internet.
6. Associez la nouvelle table de routage au sous-réseau public.
7. Créez une adresse IP Elastic. Pour plus d'informations, consultez Adresses IP élastiques.
8. Créez une nouvelle passerelle NAT et affectez-la au sous-réseau public et à l'adresse IP Elastic.
9. Créez un sous-réseau privé dans le VPC.
10. Ajoutez un routage à la table de routage par défaut du VPC, qui va de `0.0.0.0/0` à la passerelle NAT.
11. Créez votre script Canary.

Écriture d'un script Canary Python

Ce script est transmis comme une exécution réussie et renvoie une chaîne. Pour voir à quoi ressemble un script Canary en échec, remplacez fail = False par fail = True

```
def basic_custom_script():
    # Insert your code here
    # Perform multi-step pass/fail check
    # Log decisions made and results to /tmp
    # Be sure to wait for all your code paths to complete
    # before returning control back to Synthetics.
    # In that way, your canary will not finish and report success
    # before your code has finished executing
    fail = False
    if fail:
        raise Exception("Failed basicCanary check.")
    return "Successfully completed basicCanary checks."
def handler(event, context):
    return basic_custom_script()
```

Empaqueter vos fichiers Python Canary

Si vous avez plusieurs fichiers .py ou une dépendance à votre script, vous pouvez les regrouper dans un seul fichier ZIP. Si vous utilisez l'exécution `syn-python-selenium-1.1`, le fichier ZIP doit contenir votre fichier .py canary principal dans un dossier `python`, tel que `python/my_canary_filename.py`. Si vous utilisez `syn-python-selenium-1.1` ou une version ultérieure, vous pouvez éventuellement utiliser un autre dossier, tel que `python/myFolder/my_canary_filename.py`.

Ce fichier ZIP doit contenir tous les dossiers et fichiers nécessaires, mais les autres fichiers n'ont pas besoin de se trouver dans le dossier `python`.

Assurez-vous de définir le point d'entrée de votre script Canary de sorte que `my_canary_filename.functionName` corresponde au nom du fichier et de la fonction du point d'entrée de votre script. Si vous utilisez une exécution `syn-python-selenium-1.0`, alors `functionName` doit être `handler`. Si vous utilisez `syn-python-selenium-1.1` ou une version ultérieure, cette restriction du nom du gestionnaire ne s'applique pas et vous pouvez également stocker le script Canary dans un dossier séparé tel que `python/myFolder/my_canary_filename.py`. Si vous le stockez dans un dossier séparé, spécifiez ce chemin dans le point d'entrée de votre script, tel que `myFolder/my_canary_filename.functionName`.

Modification d'un script Selenium existant pour l'utiliser en tant que script Canary Synthetics

Vous pouvez rapidement modifier un script existant pour Python et Selenium pour l'utiliser comme script Canary. Pour de plus amples informations sur Selenium, consultez www.selenium.dev/.

Pour cet exemple, nous commencerons par le script Selenium suivant :

```
from selenium import webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

basic_selenium_script()
```

Les étapes de conversion sont les suivantes.

Pour convertir un script Selenium à utiliser comme script Canary

1. Modifier l'instruction `import` pour utiliser Selenium à partir du module `aws_synthetics` :

   ```
   from aws_synthetics.selenium import synthetics_webdriver as webdriver
   ```

   Le module Selenium de `aws_synthetics` garantit que le Canary peut émettre des métriques et des journaux, générer un fichier HAR et fonctionner avec d'autres fonctionnalités de CloudWatch Synthetics.

2. Créez une fonction de gestionnaire et appelez votre méthode Selenium. Le gestionnaire est la fonction de point d'entrée du script.

   Si vous utilisez `syn-python-selenium-1.0`, la fonction du gestionnaire doit être nommée `handler`. Si vous utilisez `syn-python-selenium-1.1` ou une version ultérieure, la fonction peut porter n'importe quel nom, mais il doit être identique à celui utilisé dans le script. De plus, si vous utilisez `syn-python-selenium-1.1` ou une version ultérieure, vous pouvez stocker vos scripts dans n'importe quel dossier et spécifier ce dossier dans le nom du gestionnaire.

   ```
   def handler(event, context):
       basic_selenium_script()
   ```

Le script est maintenant mis à jour pour devenir un canari CloudWatch Synthetics. Voici le script mis à jour :

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver

def basic_selenium_script():
    browser = webdriver.Chrome()
    browser.get('https://example.com')
    browser.save_screenshot('loaded.png')

def handler(event, context):
    basic_selenium_script()
```

Modification d'un script Puppeteer Synthetics existant pour authentifier des certificats non standard

L'un des principaux cas d'utilisation des canaris Synthetics est de vous permettre de surveiller vos propres terminaux. Si vous souhaitez surveiller un point de terminaison qui n'est pas prêt pour

le trafic externe, cette surveillance peut parfois signifier que vous ne disposez pas d'un certificat approprié signé par une autorité de certification tierce de confiance.

Les deux solutions possibles à ce scénario sont les suivantes :

- Pour authentifier un certificat client, consultez Comment valider l'authentification à l'aide d'Amazon CloudWatch Synthetics — Partie 2.
- Pour authentifier un certificat auto-signé, consultez Comment valider l'authentification à l'aide de certificats auto-signés dans Amazon Synthetics CloudWatch

Vous n'êtes pas limité à ces deux options lorsque vous utilisez les canaris CloudWatch Synthetics. Vous pouvez étendre ces fonctionnalités et ajouter votre logique métier en étendant le code Canary.

> **Note**
>
> L'indicateur est activé par défaut pour les canaris Synthetics exécutés sur des environnements d'exécution Python. Ces canaris ne devraient donc pas `--ignore-certificate-errors` avoir de problèmes pour accéder à des sites dont les configurations de certificats ne sont pas standard.

Fonctions de bibliothèque disponibles pour les scripts Canary

CloudWatch Synthetics inclut plusieurs classes et fonctions intégrées que vous pouvez appeler lorsque vous écrivez des scripts Node.js à utiliser comme canaris.

Certaines de ces classes et fonctions s'appliquent à la fois aux scripts Canary d'interface utilisateur et d'API. D'autres s'appliquent uniquement aux scripts Canary de l'interface utilisateur. Un script Canary d'interface utilisateur est un script Canary qui utilise la fonction `getPage()` et qui utilise Puppeteer comme pilote web pour naviguer entre les pages web et interagir avec elles.

> **Note**
>
> Chaque fois que vous mettez à niveau un script canary pour utiliser une nouvelle version de l'exécution Synthetics, toutes les fonctions de la bibliothèque Synthetics utilisées par votre script canary sont également automatiquement mises à niveau vers la version de NodeJS prise en charge par l'exécution Synthetics.

Rubriques

Fonctions de bibliothèque disponibles pour les scripts Canary Node.js

Cette section répertorie les fonctions de bibliothèque disponibles pour les scripts Canary Node.js

Rubriques

Classes et fonctions de bibliothèque Node.js qui s'appliquent à tous les scripts Canary

Les fonctions de la bibliothèque CloudWatch Synthetics suivantes pour Node.js sont utiles pour tous les canaris.

Rubriques

Classe Synthetics

Les fonctions suivantes pour tous les scripts Canary sont dans la classe Synthetics.

addExecutionError(Message d'erreur, par exemple) ;

`errorMessage` décrit l'erreur et `ex` est l'exception rencontrée.

Vous pouvez utiliser `addExecutionError` pour définir les erreurs d'exécution pour votre script Canary. Ce code fait échouer le script Canary sans interrompre l'exécution du script. Cela n'a pas non plus d'impact sur vos métriques `successPercent`.

Vous ne devriez suivre les erreurs comme des erreurs d'exécution que si elles ne sont pas importantes pour indiquer le succès ou l'échec de votre script Canary.

L'exemple suivant illustre l'utilisation de `addExecutionError`. Vous surveillez la disponibilité de votre point de terminaison et vous prenez des captures d'écran après le chargement de la page. Étant donné que l'échec de la prise d'une capture d'écran ne détermine pas la disponibilité du point de terminaison, vous pouvez détecter toutes les erreurs rencontrées lors de la prise de captures d'écran et les ajouter en tant qu'erreurs d'exécution. Vos métriques de disponibilité indiqueront toujours que le point de terminaison est opérationnel, mais le statut de votre script Canary indiquera qu'il a échoué. L'exemple de bloc de code suivant détecte une telle erreur et l'ajoute en tant qu'erreur d'exécution.

```
try {
        await synthetics.takeScreenshot(stepName, "loaded");
    } catch(ex) {
        synthetics.addExecutionError('Unable to take screenshot ', ex);
    }
```

getCanaryName();

Renvoie le nom du script Canary.

getCanaryArn();

Renvoie l'ARN du script canary.

getCanaryUserAgentString();

Renvoie l'agent utilisateur personnalisé du script canary.

getRuntimeVersion();

Cette fonction est disponible dans la version d'exécution `syn-nodejs-puppeteer-3.0` et versions ultérieures. Elle renvoie la version d'exécution Synthetics du script Canary. Par exemple, la valeur renvoyée peut être `syn-nodejs-puppeteer-3.0`.

getLogLevel();

Récupère le niveau de journalisation actuel pour la bibliothèque Synthetics. Les valeurs possibles sont les suivantes :

- `0` : débogage

- 1 : informations
- 2 : avertissement
- 3 : erreur

Exemple :

```
let logLevel = synthetics.getLogLevel();
```

setLogLevel();

Définit le niveau de journalisation pour la bibliothèque Synthetics. Les valeurs possibles sont les suivantes :

- `0` : débogage
- 1 : informations
- 2 : avertissement
- 3 : erreur

Exemple :

```
synthetics.setLogLevel(0);
```

SyntheticsConfiguration classe

Cette classe est disponible uniquement dans la version d'exécution `syn-nodejs-2.1` ou version ultérieure.

Vous pouvez utiliser la SyntheticsConfiguration classe pour configurer le comportement des fonctions de la bibliothèque Synthetics. Par exemple, vous pouvez utiliser cette classe pour configurer la fonction `executeStep()` pour ne pas prendre de captures d'écran.

Vous pouvez définir CloudWatch des configurations Synthetics au niveau global, qui sont appliquées à toutes les étapes des canaris. Vous pouvez également remplacer ces configurations au niveau de l'étape en appliquant des paires clé/valeur.

Vous pouvez transmettre des options au niveau de l'étape. Pour obtenir des exemples, veuillez consulter async ExecuteStep (StepName, [StepConfig]) ; functionToExecute et executeHttpStep(StepName, RequestOptions, [rappel], [StepConfig])

Définitions des fonctions :

setConfig(options)

`options` est un objet, qui est un ensemble d'options configurables pour votre script Canary. Les sections suivantes expliquent les champs possibles dans *`options`*.

setConfig(options) pour tous les scripts Canary

Pour les scripts Canary utilisant `syn-nodejs-puppeteer-3.2` ou version ultérieure, les (options) pour setConfig peut inclure les paramètres suivants :

- `includeRequestHeaders` (booléen) : indique si les en-têtes de demande doivent être inclus dans le rapport. La valeur par défaut est `false`.
- `includeResponseHeaders` (booléen) : indique si les en-têtes de réponse doivent être inclus dans le rapport. La valeur par défaut est `false`.
- `restrictedHeaders` (tableau) : une liste de valeurs d'en-tête à ignorer, si les en-têtes sont inclus. Cela s'applique à la fois aux en-têtes de demande et de réponse. Par exemple, vous pouvez masquer vos informations d'identification en passant includeRequestHeadersas `true` et RestrictedHeaders as. `['Authorization']`
- `includeRequestBody` (booléen) : indique si le corps de requête doit être inclus dans le rapport. L'argument par défaut est `false`.
- `includeResponseBody` (booléen) : indique si le corps de réponse doit être inclus dans le rapport. L'argument par défaut est `false`.

SetConfig (options) concernant les métriques CloudWatch

Pour les scripts Canary utilisant `syn-nodejs-puppeteer-3.1` ou version ultérieure, les (options) pour setConfig peuvent inclure les paramètres booléens suivants qui déterminent les métriques publiées par le script Canary. La valeur par défaut de chacune de ces options est `true`. Les options qui commencent par `aggregated` déterminent si la métrique est émise sans la dimension `CanaryName`. Vous pouvez utiliser ces métriques pour afficher les résultats agrégés de tous vos scripts Canary. Les autres options déterminent si la métrique est émise avec la dimension `CanaryName`. Vous pouvez utiliser ces métriques pour afficher les résultats de chaque script Canary individuel.

Pour une liste des CloudWatch métriques émises par les canaris, voirCloudWatch statistiques publiées par canaries.

- `failedCanaryMetric` (booléen) : indique s'il faut émettre la métrique `Failed` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `failedRequestsMetric` (booléen) : indique s'il faut émettre la métrique `Failed requests` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `_2xxMetric` (booléen) : indique s'il faut émettre la métrique `2xx` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `_4xxMetric` (booléen) : indique s'il faut émettre la métrique `4xx` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `_5xxMetric` (booléen) : indique s'il faut émettre la métrique `5xx` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `stepDurationMetric` (booléen) : indique s'il faut émettre la métrique `Step duration` (avec les dimensions `CanaryName` `StepName`) pour ce script Canary. L'argument par défaut est `true`.
- `stepSuccessMetric` (booléen) : indique s'il faut émettre la métrique `Step success` (avec les dimensions `CanaryName` `StepName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregatedFailedCanaryMetric` (booléen) : indique s'il faut émettre la métrique `Failed` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregatedFailedRequestsMetric` (booléen) : indique s'il faut émettre la métrique `Failed Requests` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregated2xxMetric` (booléen) : indique s'il faut émettre la métrique `2xx` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregated4xxMetric` (booléen) : indique s'il faut émettre la métrique `4xx` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregated5xxMetric` (booléen) : indique s'il faut émettre la métrique `5xx` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `visualMonitoringSuccessPercentMetric` (booléen) : indique s'il faut émettre la métrique `visualMonitoringSuccessPercent` pour ce script Canary. L'argument par défaut est `true`.
- `visualMonitoringTotalComparisonsMetric` (booléen) : indique s'il faut émettre la métrique `visualMonitoringTotalComparisons` pour ce script Canary. L'argument par défaut est `false`.
- `stepsReport` (booléen) : indique s'il faut créer un rapport avec le résumé de l'exécution d'une étape. L'argument par défaut est `true`.

- `includeUrlPassword` (booléen) : indique s'il faut inclure un mot de passe qui apparaît dans l'URL. Par défaut, les mots de passe qui apparaissent dans les URL sont effacés des journaux et des rapports afin d'empêcher la divulgation de données sensibles. L'argument par défaut est `false`.
- `restrictedUrlParameters` (tableau) : liste de paramètres de chemin d'URL ou de requête à effacer. Cette option s'applique aux URL apparaissant dans les journaux, les rapports et les erreurs. Le paramètre est sensible à la casse. Vous pouvez utiliser un astérisque (*) en tant que valeur pour effacer toutes les valeurs de paramètre de chemin d'URL et de requête. La valeur par défaut est un tableau vide.
- `logRequest` (booléen) : indique s'il faut journaliser chaque demande dans les journaux des scripts Canary. Pour les scripts Canary d'interface utilisateur, cette option journalise chaque demande envoyée par le navigateur. L'argument par défaut est `true`.
- `logResponse` (booléen) : indique s'il faut journaliser chaque réponse dans les journaux des scripts Canary. Pour les scripts Canary d'interface utilisateur, cette option journalise chaque réponse reçue par le navigateur. L'argument par défaut est `true`.
- `logRequestBody` (booléen) : indique si les corps de requête doivent être journalisés avec les requêtes dans les journaux des scripts Canary. Cette configuration s'applique uniquement si `logRequest` est `true`. L'argument par défaut est `false`.
- `logResponseBody` (booléen) : indique si les corps de réponse doivent être journalisés avec les réponses dans les journaux des scripts Canary. Cette configuration s'applique uniquement si `logResponse` est `true`. L'argument par défaut est `false`.
- `logRequestHeaders` (booléen) : indique si les en-têtes de requête doivent être journalisés avec les requêtes dans les journaux des scripts Canary. Cette configuration s'applique uniquement si `logRequest` est `true`. L'argument par défaut est `false`.

  Notez que `includeRequestHeaders` active les en-têtes dans les artefacts.
- `logResponseHeaders` (booléen) : indique si les en-têtes de réponse doivent être journalisés avec les réponses dans les journaux des scripts Canary. Cette configuration s'applique uniquement si `logResponse` est `true`. L'argument par défaut est `false`.

  Notez que `includeResponseHeaders` active les en-têtes dans les artefacts.

> **Note**
>
> Les métriques `Duration` et `SuccessPercent` sont toujours émises pour chaque script Canary, à la fois avec et sans la métrique `CanaryName`.

Méthodes pour activer ou désactiver les métriques

disableAggregatedRequestMétriques ()

Désactive l'émission par le script Canary de toutes les métriques de demande émises sans dimension `CanaryName`.

disableRequestMetrics()

Désactive toutes les métriques de demande, y compris les métriques par script Canary et les métriques agrégées pour tous les scripts Canary.

disableStepMetrics()

Désactive toutes les métriques d'étapes, y compris les métriques de succès des étapes et les métriques de durée des étapes.

enableAggregatedRequestMétriques ()

Active l'émission par le script Canary de toutes les métriques de demande émises sans dimension `CanaryName`.

enableRequestMetrics()

Active toutes les métriques de demande, y compris les métriques par script Canary et les métriques agrégées pour tous les scripts Canary.

enableStepMetrics()

Active toutes les métriques d'étapes, y compris les métriques de succès des étapes et les métriques de durée des étapes.

get2xxMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `2xx` avec la dimension `CanaryName`.

get4xxMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `4xx` avec la dimension `CanaryName`.

get5xxMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `5xx` avec la dimension `CanaryName`.

getAggregated2xxMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `2xx` sans dimension.

getAggregated4xxMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `4xx` sans dimension.

getAggregatedFailedCanaryMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `Failed` sans dimension.

getAggregatedFailedRequestsMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `Failed requests` sans dimension.

getAggregated5xxMetric()

Renvoie une valeur indiquant si le script Canary émet une métrique `5xx` sans dimension.

getFailedCanaryMétrique ()

Renvoie une valeur indiquant si le script Canary émet une métrique `Failed` avec la dimension `CanaryName`.

getFailedRequestsMétrique ()

Renvoie une valeur indiquant si le script Canary émet une métrique `Failed requests` avec la dimension `CanaryName`.

getStepDurationMétrique ()

Renvoie une valeur indiquant si le script Canary émet une métrique `Duration` avec la dimension `CanaryName` pour ce script Canary.

getStepSuccessMétrique ()

Renvoie une valeur indiquant si le script Canary émet une métrique `StepSuccess` avec la dimension `CanaryName` pour ce script Canary.

with2xxMetric(_2xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `2xx` avec la dimension `CanaryName` pour ce script Canary.

with4xxMetric(_4xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `4xx` avec la dimension `CanaryName` pour ce script Canary.

with5xxMetric(_5xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `5xx` avec la dimension `CanaryName` pour ce script Canary.

withAggregated2xxMetric(aggregated2xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `2xx` sans dimension pour ce script Canary.

withAggregated4xxMetric(aggregated4xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `4xx` sans dimension pour ce script Canary.

withAggregated5xxMetric(aggregated5xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `5xx` sans dimension pour ce script Canary.

withAggregatedFailedCanaryMetric(aggregatedFailedCanaryMétrique)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed` sans dimension pour ce script Canary.

withAggregatedFailedRequestsMetric(aggregatedFailedRequestsMétrique)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed requests` sans dimension pour ce script Canary.

withFailedCanaryMétrique (failedCanaryMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed` avec la dimension `CanaryName` pour ce script Canary.

withFailedRequestsMétrique (failedRequestsMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed requests` avec la dimension `CanaryName` pour ce script Canary.

withStepDurationMétrique (stepDurationMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Duration` avec la dimension `CanaryName` pour ce script Canary.

withStepSuccessMétrique (stepSuccessMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `StepSuccess` avec la dimension `CanaryName` pour ce script Canary.

Méthodes pour activer ou désactiver d'autres fonctions

withHarFile()

Accepte un argument booléen qui spécifie s'il faut créer un fichier HAR pour ce script Canary.

withStepsReport()

Accepte un argument booléen qui spécifie s'il faut créer un rapport avec un résumé de l'exécution des étapes pour ce script Canary.

withIncludeUrlMot de passe ()

Accepte un argument booléen qui spécifie s'il faut inclure dans les journaux et les rapports les mots de passe qui apparaissent dans les URL.

withRestrictedUrlParamètres ()

Accepte un tableau des paramètres de chemin d'URL ou de requête à effacer. Cette option s'applique aux URL apparaissant dans les journaux, les rapports et les erreurs. Vous pouvez utiliser un

astérisque (*) en tant que valeur pour effacer toutes les valeurs de paramètre de chemin d'URL et de requête.

withLogRequest()

Accepte un argument booléen qui spécifie si chaque demande doit être journalisée dans les journaux du script Canary.

withLogResponse()

Accepte un argument booléen qui spécifie si chaque réponse doit être journalisée dans les journaux du script Canary.

withLogRequestCorps ()

Accepte un argument booléen qui spécifie si chaque corps de requête doit être journalisé dans les journaux du script Canary.

withLogResponseCorps ()

Accepte un argument booléen qui spécifie si chaque corps de réponse doit être journalisé dans les journaux du script Canary.

withLogRequestEn-têtes ()

Accepte un argument booléen qui spécifie si chaque en-tête de demande doit être journalisé dans les journaux du script Canary.

withLogResponseEn-têtes ()

Accepte un argument booléen qui spécifie si chaque en-tête de réponse doit être journalisé dans les journaux du script Canary.

getHarFile()

Renvoie une valeur indiquant si le script Canary crée un fichier HAR.

getStepsReport()

Renvoie une valeur indiquant si le script Canary génère un rapport avec un résumé de l'exécution des étapes.

getIncludeUrlMot de passe ()

Renvoie une valeur indiquant si le script Canary inclut dans les journaux et les rapports les mots de passe qui apparaissent dans les URL.

getRestrictedUrlParamètres ()

Renvoie une valeur indiquant si le script Canary supprime les paramètres de chemin d'URL ou de requête.

getLogRequest()

Renvoie une valeur indiquant si le script Canary journalise chaque demande dans les journaux du script Canary.

getLogResponse()

Renvoie une valeur indiquant si le script Canary journalise chaque réponse dans les journaux du script Canary.

getLogRequestCorps ()

Renvoie une valeur indiquant si le script Canary journalise chaque corps de requête dans les journaux du script Canary.

getLogResponseCorps ()

Renvoie une valeur indiquant si le script Canary journalise chaque corps de réponse dans les journaux du script Canary.

getLogRequestEn-têtes ()

Renvoie une valeur indiquant si le script Canary journalise chaque en-tête de demande dans les journaux du script Canary.

getLogResponseEn-têtes ()

Renvoie une valeur indiquant si le script Canary journalise chaque en-tête de réponse dans les journaux du script Canary.

Fonctions pour tous les scripts canary

- `withIncludeRequestHeaders`(includeRequestHeaders)
- `withIncludeResponseHeaders`(includeResponseHeaders)
- `withRestrictedHeaders`(restrictedHeaders)

- `withIncludeRequestBody`(includeRequestBody)
- `withIncludeResponseBody`(includeResponseBody)
- `enableReportingOptions`() — Active toutes les options de reporting-- includeRequestHeadersincludeResponseHeaders, includeRequestBody, et includeResponseBody,.
- `disableReportingOptions`() — Désactive toutes les options de création de rapports-- includeRequestHeadersincludeResponseHeaders, includeRequestBody, et includeResponseBody,.

setConfig(options) pour les scripts Canary d'interface utilisateur

Pour les scripts Canary d'interface utilisateur, setConfig peut inclure les paramètres booléens suivants :

- `continueOnStepFailure` (booléen) : indique s'il faut poursuivre l'exécution du script Canary après l'échec d'une étape (cela fait référence à la fonction executeStep). Si une étape échoue, l'exécution du script Canary sera toujours marquée comme ayant échoué. L'argument par défaut est `false`.
- `harFile` (booléen) : indique s'il faut créer un fichier HAR. L'argument par défaut est `True`.
- `screenshotOnStepStart` (booléen) : indique s'il faut prendre une capture d'écran avant de commencer une étape.
- `screenshotOnStepSuccess` (booléen) : indique s'il faut prendre une capture d'écran après la réussite d'une étape.
- `screenshotOnStepFailure` (booléen) : indique s'il faut prendre une capture d'écran après l'échec d'une étape.

Méthodes pour activer ou désactiver les captures d'écran

disableStepScreenshots()

Désactive toutes les options de capture d'écran (screenshotOnStepdémarrage, screenshotOnStep réussite et screenshotOnStep échec).

enableStepScreenshots()

Active toutes les options de capture d'écran (screenshotOnStepdémarrage, screenshotOnStep réussite et screenshotOnStep échec). Par défaut, toutes ces méthodes sont activées.

getScreenshotOnStepFailure()

Renvoie une valeur indiquant si le script Canary prend une capture d'écran après l'échec d'une étape.

getScreenshotOnStepStart()

Renvoie une valeur indiquant si le script Canary prend une capture d'écran avant de commencer une étape.

getScreenshotOnStepSuccess()

Renvoie une valeur indiquant si le script Canary prend une capture d'écran après la réussite d'une étape.

withScreenshotOnStepStart(screenshotOnStepDémarrer)

Accepte un argument booléen qui indique s'il faut prendre une capture d'écran avant de commencer une étape.

withScreenshotOnStepSuccess(screenshotOnStepSuccès)

Accepte un argument booléen qui indique s'il faut prendre une capture d'écran après la réussite d'une étape.

withScreenshotOnStepFailure(screenshotOnStepÉchec)

Accepte un argument booléen qui indique s'il faut prendre une capture d'écran après l'échec d'une étape.

Utilisation dans les scripts canary d'interface utilisateur

Tout d'abord, importez la dépendance Synthetics et récupérez la configuration.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();
```

Ensuite, définissez la configuration de chaque option en appelant la méthode setConfig à l'aide de l'une des options suivantes.

```
// Set configuration values
    synConfig.setConfig({
        screenshotOnStepStart: true,
        screenshotOnStepSuccess: false,
        screenshotOnStepFailure: false
    });
```

Ou

```
synConfig.withScreenshotOnStepStart(false).withScreenshotOnStepSuccess(true).withScreenshotOnSt
```

Pour désactiver toutes les captures d'écran, utilisez la fonction disableStepScreenshots () comme dans cet exemple.

```
synConfig.disableStepScreenshots();
```

Vous pouvez activer et désactiver des captures d'écran à tout moment dans le code. Par exemple, pour désactiver les captures d'écran pour une seule étape, désactivez-les avant d'exécuter cette étape, puis activez-les après l'étape.

setConfig(options) pour les scripts Canary d'API

Pour les scripts Canary d'API, setConfig peut inclure les paramètres booléens suivants :

- `continueOnHttpStepFailure`(booléen) — S'il faut continuer à exécuter le script Canary après l'échec d'une étape HTTP (cela fait référence à la executeHttpStepfonction). Si une étape échoue, l'exécution du script Canary sera toujours marquée comme ayant échoué. L’argument par défaut est `true`.

Surveillance visuelle

La surveillance visuelle compare les captures d'écran prises lors de l'exécution d'un script Canary aux captures d'écran prises lors de l'exécution d'un script Canary de référence. Si l'écart entre les deux captures d'écran dépasse un pourcentage de seuil, le script Canary échoue et les zones présentant des différences sont mises en évidence en couleur dans le rapport d'exécution du script Canary. La surveillance visuelle est prise en charge dans les canaris utilisant syn-puppeteer-node-3.2 et versions ultérieures. Elle n'est actuellement pas prise en charge dans les scripts Canary exécutant Python et Selenium.

Pour activer la surveillance visuelle, ajoutez la ligne de code suivante au script Canary. Pour en savoir plus, consultez SyntheticsConfiguration classe.

```
syntheticsConfiguration.withVisualCompareWithBaseRun(true);
```

La première fois que le script Canary s'exécute avec succès après l'ajout de cette ligne au script, il utilise les captures d'écran prises au cours de cette exécution comme référence pour les comparaisons. Après cette première exécution de Canary, vous pouvez utiliser la CloudWatch console pour modifier le Canary afin d'effectuer l'une des opérations suivantes :

- Définir la prochaine exécution du script Canary comme nouvelle référence.
- Dessiner des limites sur la capture d'écran de référence actuelle pour désigner les zones de la capture d'écran à ignorer lors des comparaisons visuelles.
- Empêcher une capture d'écran d'être utilisée pour la surveillance visuelle.

Pour plus d'informations sur l'utilisation de la CloudWatch console pour modifier un canari, consultezModification ou suppression d'un canary.

Autres options pour la surveillance visuelle

Configuration synthétique. withVisualVarianceThresholdPercentage(Pourcentage souhaité)

Définissez le pourcentage d'écart acceptable entre les captures d'écran lors des comparaisons visuelles.

Configuration synthétique. withVisualVarianceHighlightHexColor(» #fafa00 «)

Définissez la couleur de surbrillance qui désigne les zones d'écart lorsque vous consultez les rapports d'exécution des scripts Canary qui utilisent la surveillance visuelle.

Configuration synthétique. withFailCanaryRunOnVisualVariance(FailCanary)

Définissez si le script Canary échoue ou non lorsqu'il y a une différence visuelle supérieure au seuil. La valeur par défaut est de faire échouer le script Canary.

Enregistreur Synthetics

SyntheticsLogger écrit les déconnexions à la fois dans la console et dans un fichier journal local au même niveau de journal. Ce fichier journal est écrit dans les deux emplacements seulement si le

niveau de journalisation est égal ou inférieur au niveau de journalisation souhaité de la fonction de journalisation qui a été appelée.

Les instructions de journalisation dans le fichier journal local sont précédées de « DEBUG: », « INFO: », etc., pour respecter le niveau de journalisation de la fonction appelée.

Vous pouvez utiliser le SyntheticsLogger, en supposant que vous souhaitiez exécuter la bibliothèque Synthetics au même niveau de journalisation que votre journalisation Synthetics Canary.

SyntheticsLogger Il n'est pas nécessaire d'utiliser le pour créer un fichier journal qui est téléchargé vers votre emplacement de résultats S3. Vous pouvez créer à la place un autre fichier journal dans le dossier `/tmp`. Tous les fichiers créés sous le dossier `/tmp` sont chargés vers l'emplacement des résultats dans S3 en tant qu'artefacts.

Pour utiliser l'enregistreur de la bibliothèque Synthetics :

```
const log = require('SyntheticsLogger');
```

Définitions de fonctions utiles :

log.debug(***message***, ***ex***);

Paramètres : *`message`* est le message à consigner. *`ex`* est l'exception, le cas échéant, à consigner.

Exemple :

```
log.debug("Starting step - login.");
```

log.error(***message***, ***ex***);

Paramètres : *`message`* est le message à consigner. *`ex`* est l'exception, le cas échéant, à consigner.

Exemple :

```
try {
  await login();
catch (ex) {
  log.error("Error encountered in step - login.", ex);
}
```

log.info(***message***, ***ex***);

Paramètres : *`message`* est le message à consigner. *`ex`* est l'exception, le cas échéant, à consigner.

Exemple :

```
log.info("Successfully completed step - login.");
```

log.log(***`message`***, ***`ex`***);

Ceci est un alias pour `log.info`.

Paramètres : *`message`* est le message à consigner. *`ex`* est l'exception, le cas échéant, à consigner.

Exemple :

```
 log.log("Successfully completed step - login.");
```

log.warn(***`message`***, ***`ex`***);

Paramètres : *`message`* est le message à consigner. *`ex`* est l'exception, le cas échéant, à consigner.

Exemple :

```
log.warn("Exception encountered trying to publish CloudWatch Metric.", ex);
```

SyntheticsLogHelper classe

La classe `SyntheticsLogHelper` est disponible dans l'exécution `syn-nodejs-puppeteer-3.2` et versions ultérieures. Il est déjà initialisé dans la bibliothèque CloudWatch Synthetics et est configuré avec la configuration Synthetics. Vous pouvez l'ajouter en tant que dépendance dans votre script. Cette classe vous permet de nettoyer les URL, les en-têtes et les messages d'erreur pour effacer les informations sensibles.

> **Note**
>
> Synthetics nettoie toutes les URL et tous les messages d'erreur qu'il journalise avant de les inclure dans les journaux, les rapports, les fichiers HAR et les erreurs d'exécution de script Canary en fonction du paramètre de configuration Synthetics `restrictedUrlParameters`. Vous devez utiliser `getSanitizedUrl` ou `getSanitizedErrorMessage` uniquement si vous journalisez des URL ou des erreurs dans votre script. Synthetics ne stocke pas d'artefacts de script Canary, sauf pour les erreurs de script Canary levées par le script.

> Les artefacts d'exécution de script Canary sont stockés sur votre compte client. Pour plus d'informations, consultez Considérations de sécurité pour les scripts Canary Synthetics.

getSanitizedUrl(url, StepConfig = nul)

Cette fonction est disponible dans `syn-nodejs-puppeteer-3.2` et versions ultérieures. Elle renvoie des chaînes d'URL nettoyées en fonction de la configuration. Vous pouvez choisir d'effacer les paramètres d'URL sensibles tels que les mots de passe et access_token en définissant la propriété `restrictedUrlParameters`. Par défaut, les mots de passe dans les URL sont effacés. Vous pouvez activer les mots de passe d'URL si nécessaire en définissant `includeUrlPassword` sur true (vrai).

Cette fonction lève une erreur si l'URL transmise n'est pas une URL valide.

Paramètres

- *`url`* est une chaîne et est l'URL à nettoyer.
- *`stepConfig`* (facultatif) remplace la configuration Synthetics globale pour cette fonction. Si `stepConfig` n'est pas transmis, la configuration globale est utilisée pour nettoyer l'URL.

Exemple :

Cet exemple utilise l'exemple d'URL suivant : `https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`. Dans cet exemple, `access_token` contient vos informations sensibles qui ne doivent pas être journalisées. Notez que les services Synthetics ne stockent pas d'artefacts d'exécution de script Canary. Les artefacts tels que les journaux, les captures d'écran et les rapports sont tous stockés dans un compartiment Amazon S3 sur votre compte client.

La première étape consiste à définir la configuration Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();
```

```
// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

Ensuite, nettoyez et journalisez l'URL

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200');
```

Ceci journalise ce qui suit dans votre journal de script Canary.

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

Vous pouvez remplacer la configuration Synthetics d'une URL en transmettant un paramètre facultatif contenant des options de configuration Synthetics, comme dans l'exemple suivant.

```
const urlConfig = {
   restrictedUrlParameters = ['*']
};
const sanitizedUrl = synthetics.getSanitizedUrl('https://example.com/learn/home?
access_token=12345&token_type=Bearer&expires_in=1200', urlConfig);
logger.info('My example url is: ' + sanitizedUrl);
```

L'exemple précédent efface tous les paramètres de requête et est journalisé comme suit :

```
My example url is: https://example.com/learn/home?
access_token=REDACTED&token_type=REDACTED&expires_in=REDACTED
```

getSanitizedErrorMessage

Cette fonction est disponible dans `syn-nodejs-puppeteer-3.2` et versions ultérieures. Elle renvoie des chaînes d'erreur nettoyées en nettoyant toutes les URL présentes en fonction de la configuration Synthetics. Vous pouvez choisir de remplacer la configuration Synthetics globale lorsque vous appelez cette fonction en transmettant un paramètre `stepConfig` facultatif.

Paramètres

- *erreur* est l'erreur à nettoyer. Il peut s'agir d'un objet Error ou d'une chaîne.
- *stepConfig* (facultatif) remplace la configuration Synthetics globale pour cette fonction. Si `stepConfig` n'est pas transmis, la configuration globale est utilisée pour nettoyer l'URL.

Exemple :

Cet exemple utilise l'erreur suivante : `Failed to load url: https://example.com/learn/home?access_token=12345&token_type=Bearer&expires_in=1200`

La première étape consiste à définir la configuration Synthetics.

```
// Import Synthetics dependency
const synthetics = require('Synthetics');

// Import Synthetics logger for logging url
const log = require('SyntheticsLogger');

// Get Synthetics configuration
const synConfig = synthetics.getConfiguration();

// Set restricted parameters
synConfig.setConfig({
   restrictedUrlParameters: ['access_token'];
});
```

Ensuite, nettoyez et journalisez le message d'erreur

```
// Import SyntheticsLogHelper dependency
const syntheticsLogHelper = require('SyntheticsLogHelper');

try {
   // Your code which can throw an error containing url which your script logs
} catch (error) {
    const sanitizedErrorMessage = synthetics.getSanitizedErrorMessage(errorMessage);
    logger.info(sanitizedErrorMessage);
}
```

Ceci journalise ce qui suit dans votre journal de script Canary.

```
Failed to load url: https://example.com/learn/home?
access_token=REDACTED&token_type=Bearer&expires_in=1200
```

getSanitizedHeaders(en-têtes, StepConfig=NULL)

Cette fonction est disponible dans `syn-nodejs-puppeteer-3.2` et versions ultérieures. Elle renvoie les en-têtes nettoyés en fonction de la propriété `restrictedHeaders` de `syntheticsConfiguration`. Les en-têtes spécifiés dans la propriété `restrictedHeaders` sont effacés des journaux, des fichiers HAR et des rapports.

Paramètres

- *`en-têtes`* est un objet contenant les en-têtes à nettoyer.
- *`stepConfig`* (facultatif) remplace la configuration Synthetics globale pour cette fonction. Si `stepConfig` n'est pas transmis, la configuration globale est utilisée pour nettoyer les en-têtes.

Classes et fonctions de bibliothèque Node.js qui s'appliquent uniquement aux scripts Canary d'interface utilisateur

Les fonctions de bibliothèque CloudWatch Synthetics suivantes pour Node.js ne sont utiles que pour les canaris de l'interface utilisateur.

Rubriques

- Classe Synthetics
- BrokenLinkCheckerReport classe
- SyntheticsLink classe

Classe Synthetics

Les fonctions suivantes sont dans la classe Synthetics.

asynchrone addUserAgent (page, userAgentString) ;

Cette fonction s'ajoute *`userAgentString`* à l'en-tête de l'agent utilisateur de la page spécifiée.

Exemple :

```
await synthetics.addUserAgent(page, "MyApp-1.0");
```

Il en résulte que l'en-tête d'agent utilisateur de la page est défini sur *`browsers-user-agent-header-value`*`MyApp-1.0`

async ExecuteStep (StepName, [StepConfig]) ; functionToExecute

Exécute l'étape fournie, en l'enveloppant avec la journalisation start/pass/fail, les captures d'écran start/pass/fail et les métriques de durée et pass/fail.

> **Note**
>
> Si vous utilisez l'exécution `syn-nodejs-2.1` ou version ultérieure, vous pouvez configurer s'il faut prendre des captures d'écran et quand. Pour plus d'informations, consultez SyntheticsConfiguration classe.

La fonction `executeStep` effectue également les opérations suivantes :

- Elle consigne le fait que l'étape a commencé.
- Prend une capture d'écran appelée `<stepName>-starting`.
- Elle démarre un minuteur.
- Elle exécute la fonction fournie.
- Si la fonction a un retour normal, elle est considérée comme une réussite. Si la fonction lève une exception, elle est considérée comme un échec.
- Elle arrête le minuteur.
- Elle consigne le fait que l'étape a réussi ou échoué.
- Prend une capture d'écran appelée `<stepName>-succeeded` ou `<stepName>-failed`.
- Elle émet la métrique `stepName SuccessPercent`, 100 pour succès ou 0 pour échec.
- Elle émet la métrique `stepName Duration` avec une valeur basée sur les heures de début et de fin de l'étape.
- Enfin, elle retourne ce que `functionToExecute` a retourné ou lève à son tour ce que `functionToExecute` a levé.

Si le script Canary utilise l'exécution `syn-nodejs-2.0` ou version ultérieure, cette fonction ajoute également un résumé de l'exécution des étapes au rapport du script Canary. Le résumé inclut des détails sur chaque étape, tels que l'heure de début, l'heure de fin, le statut (PASSED/FAILED

[RÉUSSITE/ÉCHEC]), le motif de l'échec (en cas d'échec) et les captures d'écran prises lors de l'exécution de chaque étape.

Exemple :

```
await synthetics.executeStep('navigateToUrl', async function (timeoutInMillis = 30000)
 {
           await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});});
```

Réponse :

Retourne ce que `functionToExecute` retourne.

Mises à jour avec syn-nodejs-2.2

En commençant par`syn-nodejs-2.2`, vous pouvez éventuellement passer des configurations d'étape pour remplacer les configurations CloudWatch Synthetics au niveau de l'étape. Pour obtenir la liste des options que vous pouvez transmettre à `executeStep`, consultez SyntheticsConfiguration classe.

L'exemple suivant remplace la configuration par défaut `false` pour `continueOnStepFailure` par `true` et spécifie quand prendre des captures d'écran.

```
var stepConfig = {
    'continueOnStepFailure': true,
    'screenshotOnStepStart': false,
    'screenshotOnStepSuccess': true,
    'screenshotOnStepFailure': false
}

await executeStep('Navigate to amazon', async function (timeoutInMillis = 30000) {
      await page.goto(url, {waitUntil: ['load', 'networkidle0'], timeout:
 timeoutInMillis});
 }, stepConfig);
```

getDefaultLaunchDes options () ;

La `getDefaultLaunchOptions()` fonction renvoie les options de lancement du navigateur utilisées par CloudWatch Synthetics. Pour plus d'informations, voir Launch options type

```
// This function returns default launch options used by Synthetics.
```

```
const defaultOptions = await synthetics.getDefaultLaunchOptions();
```

getPage();

Retourne la page ouverte actuelle en tant qu'objet Puppeteer. Pour de plus amples informations, reportez-vous à l'API Puppeteer v1.14.0.

Exemple :

```
let page = synthetics.getPage();
```

Réponse :

La page (objet Puppeteer) actuellement ouverte dans la session de navigateur en cours.

getRequestResponseLogHelper();

> **Important**
>
> Dans les scripts Canary qui utilisent l'exécution `syn-nodejs-puppeteer-3.2` ou version ultérieure, cette fonction et la classe `RequestResponseLogHelper` sont rendues obsolètes. Toute utilisation de cette fonction entraîne l'apparition d'un avertissement dans les journaux des scripts Canary. Cette fonction sera supprimée dans les futures versions d'exécution. Si vous utilisez cette fonction, utilisez plutôt RequestResponseLogHelper classe.

Utilisez cette fonction comme modèle de générateur pour ajuster les indicateurs de journalisation des demandes et des réponses.

Exemple :

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper().withLogRequestHeaders(fals
```

Réponse :

```
{RequestResponseLogHelper}
```

launch(options)

Les options de cette fonction ne sont disponibles que dans l'exécution `syn-nodejs-2.1` ou version ultérieure.

Cette fonction est utilisée uniquement pour les scripts Canary d'interface utilisateur. Elle ferme le navigateur existant et en lance un nouveau.

> **Note**
>
> CloudWatch Synthetics lance toujours un navigateur avant de commencer à exécuter votre script. Vous n'avez pas besoin d'appeler launch(), sauf si vous voulez lancer un nouveau navigateur avec des options personnalisées.

(options) est un ensemble configurable d'options à définir sur le navigateur. Pour plus d'informations, consultez Launch options type.

Si vous appelez cette fonction sans options, Synthetics lance un navigateur avec des arguments par défaut, `executablePath` et `defaultViewport`. La fenêtre d'affichage par défaut dans CloudWatch Synthetics est 1920 x 1080.

Vous pouvez remplacer les paramètres de lancement utilisés par CloudWatch Synthetics et transmettre des paramètres supplémentaires lors du lancement du navigateur. Par exemple, l'extrait de code suivant lance un navigateur avec des arguments par défaut et un chemin d'accès au fichier exécutable par défaut, mais avec une fenêtre d'affichage de 800 x 600.

```
await synthetics.launch({
        defaultViewport: {
            "deviceScaleFactor": 1,
            "width": 800,
            "height": 600
    }});
```

L'exemple de code suivant ajoute un nouveau `ignoreHTTPSErrors` paramètre aux paramètres de lancement de CloudWatch Synthetics :

```
await synthetics.launch({
        ignoreHTTPSErrors: true
 });
```

Vous pouvez désactiver la sécurité Web en ajoutant un `--disable-web-security` indicateur aux arguments dans les paramètres de lancement de CloudWatch Synthetics :

```
// This function adds the --disable-web-security flag to the launch parameters
```

```
const defaultOptions = await synthetics.getDefaultLaunchOptions();
const launchArgs = [...defaultOptions.args, '--disable-web-security'];
await synthetics.launch({
     args: launchArgs
  });
```

RequestResponseLogHelper classe

> **Important**
>
> Dans les scripts Canary qui utilisent l'exécution `syn-nodejs-puppeteer-3.2` ou version ultérieure, cette classe est rendue obsolète. Toute utilisation de cette classe entraîne l'apparition d'un avertissement dans les journaux des scripts Canary. Cette fonction sera supprimée dans les futures versions d'exécution. Si vous utilisez cette fonction, utilisez plutôt RequestResponseLogHelper classe.

Elle gère la configuration précise et la création de représentations chaînes des charges utiles de demandes et de réponses.

```
class RequestResponseLogHelper {

    constructor () {
        this.request = {url: true, resourceType: false, method: false, headers: false,
 postData: false};
        this.response = {status: true, statusText: true, url: true, remoteAddress:
 false, headers: false};
    }

    withLogRequestUrl(logRequestUrl);

    withLogRequestResourceType(logRequestResourceType);

    withLogRequestMethod(logRequestMethod);

    withLogRequestHeaders(logRequestHeaders);

    withLogRequestPostData(logRequestPostData);


    withLogResponseStatus(logResponseStatus);
```

```
    withLogResponseStatusText(logResponseStatusText);

    withLogResponseUrl(logResponseUrl);

    withLogResponseRemoteAddress(logResponseRemoteAddress);

    withLogResponseHeaders(logResponseHeaders);
```

Exemple :

```
synthetics.setRequestResponseLogHelper(getRequestResponseLogHelper()
.withLogRequestPostData(true)
.withLogRequestHeaders(true)
.withLogResponseHeaders(true));
```

Réponse :

```
{RequestResponseLogHelper}
```

setRequestResponseLogHelper();

> **Important**
>
> Dans les scripts Canary qui utilisent l'exécution `syn-nodejs-puppeteer-3.2` ou version ultérieure, cette fonction et la classe `RequestResponseLogHelper` sont rendues obsolètes. Toute utilisation de cette fonction entraîne l'apparition d'un avertissement dans les journaux des scripts Canary. Cette fonction sera supprimée dans les futures versions d'exécution. Si vous utilisez cette fonction, utilisez plutôt RequestResponseLogHelper classe.

Utilisez cette fonction comme modèle de générateur pour définir les indicateurs de journalisation de demandes et de réponses.

Exemple :

```
synthetics.setRequestResponseLogHelper().withLogRequestHeaders(true).withLogResponseHeaders(tru
```

Réponse :

```
{RequestResponseLogHelper}
```

async takeScreenshot(name, suffix);

Effectue une capture d'écran (.PNG) de la page actuelle avec un nom et un suffixe (facultatif).

Exemple :

```
await synthetics.takeScreenshot("navigateToUrl", "loaded")
```

Cet exemple effectue et télécharge une capture d'écran nommée `01-navigateToUrl-loaded.png` dans le compartiment S3 du script Canary.

Vous pouvez prendre une capture d'écran pour une étape spécifique d'un script Canary en transmettant `stepName` en tant que premier paramètre. Les captures d'écran sont liées à l'étape du script Canary dans vos rapports afin de vous aider à suivre chaque étape lors du débogage.

CloudWatch Synthetics Canaries prend automatiquement des captures d'écran avant de commencer une étape (`executeStep`la fonction) et une fois l'étape terminée (sauf si vous configurez le canari pour désactiver les captures d'écran). Vous pouvez prendre plus de captures d'écran en transmettant le nom de l'étape dans la fonction `takeScreenshot`.

L'exemple suivant prend une capture d'écran avec `signupForm` comme valeur de la propriété `stepName`. La capture d'écran sera nommée `02-signupForm-address` et sera liée à l'étape nommée `signupForm` dans le rapport du script Canary.

```
await synthetics.takeScreenshot('signupForm', 'address')
```

BrokenLinkCheckerReport classe

Cette classe fournit des méthodes pour ajouter un lien Synthetics. Elle n'est prise en charge que sur les scripts Canary qui utilisent la version `syn-nodejs-2.0-beta` ou ultérieure de l'exécution.

Pour utiliser `BrokenLinkCheckerReport`, incluez les lignes suivantes dans le script :

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');

const brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

Définitions de fonctions utiles :

addLink(***syntheticsLink***, isBroken)

`syntheticsLink` est un objet `SyntheticsLink` représentant un lien. Cette fonction ajoute le lien en fonction du code de statut. Par défaut, elle considère qu'un lien est rompu si le code de statut n'est pas disponible ou s'il est 400 ou plus. Vous pouvez remplacer ce comportement par défaut en transmettant le paramètre facultatif `isBrokenLink` avec une valeur `true` ou `false`.

Cette fonction n'a pas de valeur de retour.

getLinks()

Cette fonction renvoie un tableau d'objets `SyntheticsLink` qui sont inclus dans le rapport du vérificateur de liens rompus.

getTotalBrokenLiens ()

Cette fonction renvoie un nombre représentant le nombre total de liens rompus.

getTotalLinksVérifié ()

Cette fonction renvoie un nombre représentant le nombre total de liens inclus dans le rapport.

Comment utiliser BrokenLinkCheckerReport

L'extrait de code de script Canary suivant illustre un exemple de navigation vers un lien et de l'ajout de celui-ci au rapport du vérificateur de liens rompus.

1. Importez `SyntheticsLink`, `BrokenLinkCheckerReport` et `Synthetics`.

```
const BrokenLinkCheckerReport = require('BrokenLinkCheckerReport');
const SyntheticsLink = require('SyntheticsLink');

// Synthetics dependency
const synthetics = require('Synthetics');
```

2. Pour ajouter un lien au rapport, créez une instance de `BrokenLinkCheckerReport`.

```
let brokenLinkCheckerReport = new BrokenLinkCheckerReport();
```

3. Accédez à l'URL et ajoutez-la au rapport du vérificateur de liens rompus.

```
let url = "https://amazon.com";

let syntheticsLink = new SyntheticsLink(url);
```

```
// Navigate to the url.
let page = await synthetics.getPage();

// Create a new instance of Synthetics Link
let link = new SyntheticsLink(url)

try {
    const response = await page.goto(url, {waitUntil: 'domcontentloaded', timeout:
 30000});
} catch (ex) {
    // Add failure reason if navigation fails.
    link.withFailureReason(ex);
}

if (response) {
    // Capture screenshot of destination page
    let screenshotResult = await synthetics.takeScreenshot('amazon-home', 'loaded');

    // Add screenshot result to synthetics link
    link.addScreenshotResult(screenshotResult);

    // Add status code and status description to the link
    link.withStatusCode(response.status()).withStatusText(response.statusText())
}

// Add link to broken link checker report.
brokenLinkCheckerReport.addLink(link);
```

4. Ajoutez le rapport à Synthetics. Cela crée un fichier JSON nommé `BrokenLinkCheckerReport.json` dans votre compartiment S3 pour chaque exécution de script Canary. Vous pouvez voir un rapport des liens dans la console pour chaque exécution de script Canary ainsi que des captures d'écran, des journaux et des fichiers HAR.

```
await synthetics.addReport(brokenLinkCheckerReport);
```

SyntheticsLink classe

Cette classe fournit des méthodes pour envelopper les informations. Elle n'est prise en charge que sur les scripts Canary qui utilisent la version `syn-nodejs-2.0-beta` ou ultérieure de l'exécution.

Pour utiliser `SyntheticsLink`, incluez les lignes suivantes dans le script :

```
const SyntheticsLink = require('SyntheticsLink');

const syntheticsLink = new SyntheticsLink("https://www.amazon.com");
```

Cette fonction retourne `syntheticsLink`*`Object`*.

Définitions de fonctions utiles :

withUrl(***`url`***)

`url` est une chaîne d'URL. Cette fonction retourne `syntheticsLink`*`Object`*.

withText(***`text`***)

`text` est une chaîne représentant le texte d'ancrage. Cette fonction retourne `syntheticsLink`*`Object`*. Elle ajoute le texte d'ancrage correspondant au lien.

withParentUrl(***`URL du parent`***)

`parentUrl` est une chaîne représentant l'URL parent (page source). Cette fonction retourne `syntheticsLink`*`Object`*.

withStatusCode(***`StatusCode)`***

`statusCode` est une chaîne représentant le code de statut. Cette fonction retourne `syntheticsLink`*`Object`*.

withFailureReason(Raison de ***`l'échec)`***

`failureReason` est une chaîne représentant la raison de l'échec. Cette fonction retourne `syntheticsLink`*`Object`*.

addScreenshotResult(Résultat de la ***`capture`*** d'écran)

`screenshotResult` est un objet. Il s'agit d'une instance de `ScreenshotResult` qui a été retournée par la fonction Synthetics `takeScreenshot`. L'objet inclut les éléments suivants :

- `fileName` : chaîne représentant `screenshotFileName`
- `pageUrl` (facultatif)
- `error` (facultatif)

Classes et fonctions de bibliothèque Node.js qui s'appliquent uniquement aux scripts Canary d'API

Les fonctions de bibliothèque CloudWatch Synthetics suivantes pour Node.js ne sont utiles que pour les canaris d'API.

Rubriques

executeHttpStep(StepName, RequestOptions, [rappel], [StepConfig])

Exécute la requête HTTP fournie en tant qu'étape et publie les métriques `SuccessPercent` (réussite/échec) et `Duration`.

executeHttpSteputilise des fonctions natives HTTP ou HTTPS sous le capot, selon le protocole spécifié dans la demande.

Cette fonction ajoute également un résumé de l'exécution des étapes au rapport du script Canary. Le résumé inclut des détails sur chaque requête HTTP, tels que les suivants :

- L'heure de début
- L'heure de fin
- Le statut (PASSED/FAILED [RÉUSSITE/ÉCHEC])
- La raison de l'échec, le cas échéant
- Les détails de l'appel HTTP tels que les en-têtes de requête/réponse, le corps, le code de statut, le message de statut et les temps de performance.

Paramètres

stepName(***chaîne***)

Spécifie le nom de l'étape. Ce nom est également utilisé pour publier CloudWatch les statistiques de cette étape.

requestOptions(***objet ou chaîne***)

La valeur de ce paramètre peut être une URL, une chaîne d'URL ou un objet. S'il s'agit d'un objet, il doit s'agir d'un ensemble d'options configurables pour effectuer une requête HTTP. Il prend en charge toutes les options dans http.request(options[, callback]) dans la documentation de Node.js.

En plus de ces options Node.js, requestOptions prend en charge le paramètre supplémentaire body. Vous pouvez utiliser le paramètre body pour transmettre des données en tant que corps de requête.

callback(***réponse***)

(Facultatif) Il s'agit d'une fonction utilisateur qui est appelée avec la réponse HTTP. La réponse est du type Class : http. IncomingMessage.

stepConfig(***objet***)

(Facultatif) Utilisez ce paramètre pour remplacer les configurations Synthetics globales par une configuration différente pour cette étape.

Exemples d'utilisation executeHttpStep

Les exemples suivants s'inspirent les uns des autres pour illustrer les différentes utilisations de cette option.

Ce premier exemple configure les paramètres de requête. Vous pouvez transmettre une URL en tant que requestOptions :

```
let requestOptions = 'https://www.amazon.com';
```

Vous pouvez également transmettre un ensemble d'options :

```
let requestOptions = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };
```

L'exemple suivant crée une fonction de rappel qui accepte une réponse. Par défaut, si vous ne spécifiez pas de rappel, CloudWatch Synthetics vérifie que le statut est compris entre 200 et 299 inclus.

```
// Handle validation for positive scenario
    const callback = async function(res) {
        return new Promise((resolve, reject) => {
```

```
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };
```

L'exemple suivant crée une configuration pour cette étape qui remplace la configuration globale de CloudWatch Synthetics. La configuration d'étape dans cet exemple autorise les en-têtes de requête, les en-têtes de réponse, le corps de requête (données de publication) et le corps de réponse dans votre rapport et restreint les valeurs d'en-tête 'X-Amz-Security-Token' et 'Authorization'. Par défaut, ces valeurs ne sont pas incluses dans le rapport pour des raisons de sécurité. Si vous choisissez de les inclure, les données sont stockées uniquement dans votre compartiment S3.

```
// By default headers, post data, and response body are not included in the report for
 security reasons.
// Change the configuration at global level or add as step configuration for individual
 steps
let stepConfig = {
    includeRequestHeaders: true,
    includeResponseHeaders: true,
    restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted header
 values do not appear in report generated.
    includeRequestBody: true,
    includeResponseBody: true
};
```

Ce dernier exemple transmet votre demande à l'étape executeHttpStepet lui donne un nom.

```
await synthetics.executeHttpStep('Verify GET products API', requestOptions, callback,
 stepConfig);
```

Avec cet ensemble d'exemples, CloudWatch Synthetics ajoute les détails de chaque étape de votre rapport et produit des métriques pour chaque étape à l'aide de StepName.

Vous verrez les métriques `successPercent` et `duration` pour l'étape `Verify GET products API`. Vous pouvez contrôler les performances de vos API en contrôlant les métriques des étapes d'appel de vos API.

Pour obtenir un exemple de script complet qui utilise ces fonctions, consultez Script Canary d'API à plusieurs étapes.

Fonctions de bibliothèque disponibles pour les scripts Canary Python utilisant Selenium

Cette section répertorie les fonctions de bibliothèque Selenium disponibles pour les scripts Canary Python.

Rubriques

- Classes et fonctions de bibliothèque Python et Selenium qui s'appliquent à tous les scripts Canary
- Classes et fonctions de bibliothèque Python et Selenium qui s'appliquent aux scripts Canary d'interface utilisateur uniquement

Classes et fonctions de bibliothèque Python et Selenium qui s'appliquent à tous les scripts Canary

Les fonctions de la bibliothèque CloudWatch Synthetics Selenium pour Python suivantes sont utiles pour tous les canaris.

Rubriques

- SyntheticsConfiguration classe
- SyntheticsLogger classe

SyntheticsConfiguration classe

Vous pouvez utiliser la SyntheticsConfiguration classe pour configurer le comportement des fonctions de la bibliothèque Synthetics. Par exemple, vous pouvez utiliser cette classe pour configurer la fonction `executeStep()` pour ne pas prendre de captures d'écran.

Vous pouvez définir CloudWatch des configurations Synthetics au niveau global.

Définitions des fonctions :

set_config(options)

```
from aws_synthetics.common import synthetics_configuration
```

options est un objet, qui est un ensemble d'options configurables pour votre script Canary. Les sections suivantes expliquent les champs possibles dans *options*.

- `screenshot_on_step_start` (booléen) : indique s'il faut prendre une capture d'écran avant de commencer une étape.
- `screenshot_on_step_success` (booléen) : indique s'il faut prendre une capture d'écran après la réussite d'une étape.
- `screenshot_on_step_failure` (booléen) : indique s'il faut prendre une capture d'écran après l'échec d'une étape.

with_screenshot_on_step_start(screenshot_on_step_start)

Accepte un argument booléen qui indique s'il faut prendre une capture d'écran avant de commencer une étape.

with_screenshot_on_step_success(screenshot_on_step_success)

Accepte un argument booléen qui indique s'il faut prendre une capture d'écran après la réussite d'une étape.

with_screenshot_on_step_failure(screenshot_on_step_failure)

Accepte un argument booléen qui indique s'il faut prendre une capture d'écran après l'échec d'une étape.

get_screenshot_on_step_start()

Renvoie une valeur indiquant s'il faut prendre une capture d'écran avant de commencer une étape.

get_screenshot_on_step_success()

Renvoie une valeur indiquant s'il faut prendre une capture d'écran après la réussite d'une étape.

get_screenshot_on_step_failure()

Renvoie une valeur indiquant s'il faut prendre une capture d'écran après l'échec d'une étape.

disable_step_screenshots()

Désactive toutes les options de capture d'écran (get_screenshot_on_step_start, get_screenshot_on_step_success et get_screenshot_on_step_failure).

enable_step_screenshots()

Active toutes les options de capture d'écran (get_screenshot_on_step_start, get_screenshot_on_step_success et get_screenshot_on_step_failure). Par défaut, toutes ces méthodes sont activées.

SetConfig (options) concernant les métriques CloudWatch

Pour les scripts Canary utilisant `syn-python-selenium-1.1` ou version ultérieure, les (options) pour setConfig peuvent inclure les paramètres booléens suivants qui déterminent les métriques publiées par le script Canary. La valeur par défaut de chacune de ces options est `true`. Les options qui commencent par `aggregated` déterminent si la métrique est émise sans la dimension `CanaryName`. Vous pouvez utiliser ces métriques pour afficher les résultats agrégés de tous vos scripts Canary. Les autres options déterminent si la métrique est émise avec la dimension `CanaryName`. Vous pouvez utiliser ces métriques pour afficher les résultats de chaque script Canary individuel.

Pour une liste des CloudWatch métriques émises par les canaris, voirCloudWatch statistiques publiées par canaries.

- `failed_canary_metric` (booléen) : indique s'il faut émettre la métrique `Failed` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `failed_requests_metric` (booléen) : indique s'il faut émettre la métrique `Failed requests` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `2xx_metric` (booléen) : indique s'il faut émettre la métrique `2xx` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `4xx_metric` (booléen) : indique s'il faut émettre la métrique `4xx` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `5xx_metric` (booléen) : indique s'il faut émettre la métrique `5xx` (avec la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `step_duration_metric` (booléen) : indique s'il faut émettre la métrique `Step duration` (avec les dimensions `CanaryName StepName`) pour ce script Canary. L'argument par défaut est `true`.
- `step_success_metric` (booléen) : indique s'il faut émettre la métrique `Step success` (avec les dimensions `CanaryName StepName`) pour ce script Canary. L'argument par défaut est `true`.

- `aggregated_failed_canary_metric` (booléen) : indique s'il faut émettre la métrique `Failed` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregated_failed_requests_metric` (booléen) : indique s'il faut émettre la métrique `Failed Requests` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregated_2xx_metric` (booléen) : indique s'il faut émettre la métrique `2xx` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregated_4xx_metric` (booléen) : indique s'il faut émettre la métrique `4xx` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.
- `aggregated_5xx_metric` (booléen) : indique s'il faut émettre la métrique `5xx` (sans la dimension `CanaryName`) pour ce script Canary. L'argument par défaut est `true`.

with_2xx_metric(2xx_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `2xx` avec la dimension `CanaryName` pour ce script Canary.

with_4xx_metric(4xx_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `4xx` avec la dimension `CanaryName` pour ce script Canary.

with_5xx_metric(5xx_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `5xx` avec la dimension `CanaryName` pour ce script Canary.

withAggregated2xxMetric(aggregated2xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `2xx` sans dimension pour ce script Canary.

withAggregated4xxMetric(aggregated4xxMetric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `4xx` sans dimension pour ce script Canary.

with_aggregated_5xx_metric(aggregated_5xx_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `5xx` sans dimension pour ce script Canary.

with_aggregated_failed_canary_metric(aggregated_failed_canary_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed` sans dimension pour ce script Canary.

with_aggregated_failed_requests_metric(aggregated_failed_requests_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed requests` sans dimension pour ce script Canary.

with_failed_canary_metric(failed_canary_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed` avec la dimension `CanaryName` pour ce script Canary.

with_failed_requests_metric(failed_requests_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Failed requests` avec la dimension `CanaryName` pour ce script Canary.

with_step_duration_metric(step_duration_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `Duration` avec la dimension `CanaryName` pour ce script Canary.

with_step_success_metric(step_success_metric)

Accepte un argument booléen qui spécifie s'il faut émettre une métrique `StepSuccess` avec la dimension `CanaryName` pour ce script Canary.

Méthodes pour activer ou désactiver les métriques

disable_aggregated_request_metrics()

Désactive l'émission par le script Canary de toutes les métriques de demande émises sans dimension `CanaryName`.

disable_request_metrics()

Désactive toutes les métriques de demande, y compris les métriques par script Canary et les métriques agrégées pour tous les scripts Canary.

disable_step_metrics()

Désactive toutes les métriques d'étapes, y compris les métriques de succès des étapes et les métriques de durée des étapes.

enable_aggregated_request_metrics()

Active l'émission par le script Canary de toutes les métriques de demande émises sans dimension `CanaryName`.

enable_request_metrics()

Active toutes les métriques de demande, y compris les métriques par script Canary et les métriques agrégées pour tous les scripts Canary.

enable_step_metrics()

Active toutes les métriques d'étapes, y compris les métriques de succès des étapes et les métriques de durée des étapes.

Utilisation dans les scripts canary d'interface utilisateur

Tout d'abord, importez la dépendance Synthetics et récupérez la configuration. Ensuite, définissez la configuration de chaque option en appelant la méthode setConfig à l'aide de l'une des options suivantes.

```
from aws_synthetics.common import synthetics_configuration

synthetics_configuration.set_config(
     {
        "screenshot_on_step_start": False,
        "screenshot_on_step_success": False,
        "screenshot_on_step_failure": True
     }
)

or
```

Ou

```
synthetics_configuration.with_screenshot_on_step_start(False).with_screenshot_on_step_success(F
```

Pour désactiver toutes les captures d'écran, utilisez la fonction disableStepScreenshots () comme dans cet exemple.

```
synthetics_configuration.disable_step_screenshots()
```

Vous pouvez activer et désactiver des captures d'écran à tout moment dans le code. Par exemple, pour désactiver les captures d'écran pour une seule étape, désactivez-les avant d'exécuter cette étape, puis activez-les après l'étape.

set_config(options) pour les scripts Canary d'interface utilisateur

En commençant par `syn-python-selenium-1.1`, pour les scripts Canary d'interface utilisateur, `set_config` peut inclure les paramètres booléens suivants :

- `continue_on_step_failure` (booléen) : indique s'il faut poursuivre l'exécution du script Canary après l'échec d'une étape (cela fait référence à la fonction executeStep). Si une étape échoue, l'exécution du script Canary sera toujours marquée comme ayant échoué. L'argument par défaut est `false`.

SyntheticsLogger classe

`synthetics_logger` écrit les journaux dans la console et dans un fichier journal local au même niveau de journalisation. Ce fichier journal est écrit dans les deux emplacements seulement si le niveau de journalisation est égal ou inférieur au niveau de journalisation souhaité de la fonction de journalisation qui a été appelée.

Les instructions de journalisation dans le fichier journal local sont précédées de « DEBUG: », « INFO: », etc., pour respecter le niveau de journalisation de la fonction appelée.

L'utilisation de `synthetics_logger` n'est pas nécessaire pour créer un fichier journal qui est téléchargé dans votre emplacement des résultats Amazon S3. Vous pouvez créer à la place un autre fichier journal dans le dossier `/tmp`. Tous les fichiers créés sous le dossier `/tmp` sont téléchargés vers l'emplacement des résultats dans le compartiment S3 en tant qu'artefacts.

Pour utiliser `synthetics_logger` :

```
from aws_synthetics.common import synthetics_logger
```

Définitions de fonctions utiles :

Obtenir le niveau de journalisation :

```
log_level = synthetics_logger.get_level()
```

Définir le niveau de journalisation :

```
synthetics_logger.set_level()
```

Journaliser un message avec un niveau spécifié. Le niveau peut être `DEBUG`, `INFO`, `WARN` ou `ERROR`, comme dans les exemples de syntaxe suivants :

```
synthetics_logger.debug(message, *args, **kwargs)
```

```
synthetics_logger.info(message, *args, **kwargs)
```

```
synthetics_logger.log(message, *args, **kwargs)
```

```
synthetics_logger.warn(message, *args, **kwargs)
```

```
synthetics_logger.error(message, *args, **kwargs)
```

Pour plus d'informations sur les paramètres de débogage, consultez logging.debug dans la documentation Python standard.

Dans ces fonctions de journalisation, le `message` est la chaîne de format du message. Les `args` sont les arguments qui sont fusionnés dans le `msg` à l'aide de l'opérateur de formatage de chaîne.

Il y a trois arguments de mot-clé dans `kwargs` :

- `exc_info` : s'il n'est pas évalué comme false (faux), ajoute des informations d'exception au message de journalisation.
- `stack_info` : la valeur par défaut est false (faux). Si true (vrai), ajoute des informations de pile au message de journalisation, y compris l'appel de journalisation réel.
- `extra` : le troisième argument de mot-clé facultatif, que vous pouvez utiliser pour transmettre un dictionnaire qui est utilisé pour remplir le `__dict__` du `LogRecord` créé pour l'événement de journalisation avec des attributs définis par l'utilisateur.

Exemples :

Journaliser un message avec le niveau DEBUG:

```
synthetics_logger.debug('Starting step - login.')
```

Journaliser un message avec le niveau `INFO`. `logger.log` est un synonyme de `logger.info` :

```
synthetics_logger.info('Successfully completed step - login.')
```

or

```
synthetics_logger.log('Successfully completed step - login.')
```

Journaliser un message avec le niveau `WARN` :

```
synthetics_logger.warn('Warning encountered trying to publish %s', 'CloudWatch Metric')
```

Journaliser un message avec le niveau `ERROR` :

```
synthetics_logger.error('Error encountered trying to publish %s', 'CloudWatch Metric')
```

Journaliser une exception :

```
synthetics_logger.exception(message, *args, **kwargs)
```

Journalise un message avec le niveau `ERROR`. Des informations d'exception sont ajoutées au message de journalisation. Vous ne devriez appeler cette fonction qu'à partir d'un gestionnaire d'exceptions.

Pour plus d'informations sur les paramètres d'exception, consultez logging.exception dans la documentation Python standard.

Le `message` est la chaîne de format du message. Les `args` sont les arguments qui sont fusionnés dans le `msg` à l'aide de l'opérateur de formatage de chaîne.

Il y a trois arguments de mot-clé dans `kwargs` :

- `exc_info` : s'il n'est pas évalué comme false (faux), ajoute des informations d'exception au message de journalisation.
- `stack_info` : la valeur par défaut est false (faux). Si true (vrai), ajoute des informations de pile au message de journalisation, y compris l'appel de journalisation réel.
- `extra` : le troisième argument de mot-clé facultatif, que vous pouvez utiliser pour transmettre un dictionnaire qui est utilisé pour remplir le `__dict__` du `LogRecord` créé pour l'événement de journalisation avec des attributs définis par l'utilisateur.

Exemple :

```
synthetics_logger.exception('Error encountered trying to publish %s', 'CloudWatch
 Metric')
```

Classes et fonctions de bibliothèque Python et Selenium qui s'appliquent aux scripts Canary d'interface utilisateur uniquement

Les fonctions suivantes de la bibliothèque CloudWatch Synthetics Selenium pour Python ne sont utiles que pour les canaris de l'interface utilisateur.

Rubriques

- SyntheticsBrowser classe
- SyntheticsWebDriver classe

SyntheticsBrowser classe

Lorsque vous créez une instance de navigateur en appelant `synthetics_webdriver.Chrome()`, l'instance de navigateur renvoyée est du type `SyntheticsBrowser`. La `SyntheticsBrowser` classe contrôle le ChromeDriver navigateur et permet au script Canary de piloter le navigateur, permettant ainsi au Selenium de WebDriver fonctionner avec Synthetics.

En plus des méthodes Selenium standard, elle fournit également les méthodes suivantes.

set_viewport_size(largeur, hauteur)

Définit la fenêtre d'affichage du navigateur. Exemple :

```
browser.set_viewport_size(1920, 1080)
```

save_screenshot(filename, suffix)

Enregistre les captures d'écran dans le répertoire `/tmp`. Les captures d'écran sont téléchargées à partir de ce répertoire dans le dossier des artefacts des scripts Canary dans le compartiment S3.

filename est le nom de fichier de la capture d'écran et suffix est une chaîne facultative à utiliser pour nommer la capture d'écran.

Exemple :

```
browser.save_screenshot('loaded.png', 'page1')
```

SyntheticsWebDriver classe

Pour utiliser cette classe, utilisez ce qui suit dans votre script :

```
from aws_synthetics.selenium import synthetics_webdriver
```

add_execution_error(errorMessage, ex);

`errorMessage` décrit l'erreur et `ex` est l'exception rencontrée.

Vous pouvez utiliser `add_execution_error` pour définir les erreurs d'exécution pour votre script Canary. Ce code fait échouer le script Canary sans interrompre l'exécution du script. Cela n'a pas non plus d'impact sur vos métriques `successPercent`.

Vous ne devriez suivre les erreurs comme des erreurs d'exécution que si elles ne sont pas importantes pour indiquer le succès ou l'échec de votre script Canary.

L'exemple suivant illustre l'utilisation de `add_execution_error`. Vous surveillez la disponibilité de votre point de terminaison et vous prenez des captures d'écran après le chargement de la page. Étant donné que l'échec de la prise d'une capture d'écran ne détermine pas la disponibilité du point de terminaison, vous pouvez détecter toutes les erreurs rencontrées lors de la prise de captures d'écran et les ajouter en tant qu'erreurs d'exécution. Vos métriques de disponibilité indiqueront toujours que le point de terminaison est opérationnel, mais le statut de votre script Canary indiquera qu'il a échoué. L'exemple de bloc de code suivant détecte une telle erreur et l'ajoute en tant qu'erreur d'exécution.

```
try:
    browser.save_screenshot("loaded.png")
```

```
except Exception as ex:
   self.add_execution_error("Unable to take screenshot", ex)
```

add_user_agent(user_agent_str)

Ajoute la valeur de `user_agent_str` à l'en-tête de l'agent utilisateur du navigateur. Vous devez assigner `user_agent_str` avant de créer l'instance du navigateur.

Exemple :

```
synthetics_webdriver.add_user_agent('MyApp-1.0')
```

execute_step(step_name, function_to_execute)

Traite une fonction. Elle effectue également les opérations suivantes :

- Elle consigne le fait que l'étape a commencé.
- Prend une capture d'écran appelée `<stepName>-starting`.
- Elle démarre un minuteur.
- Elle exécute la fonction fournie.
- Si la fonction a un retour normal, elle est considérée comme une réussite. Si la fonction lève une exception, elle est considérée comme un échec.
- Elle arrête le minuteur.
- Elle consigne le fait que l'étape a réussi ou échoué.
- Prend une capture d'écran appelée `<stepName>-succeeded` ou `<stepName>-failed`.
- Elle émet la métrique `stepName SuccessPercent`, 100 pour succès ou 0 pour échec.
- Elle émet la métrique `stepName Duration` avec une valeur basée sur les heures de début et de fin de l'étape.
- Enfin, elle retourne ce que `functionToExecute` a retourné ou lève à son tour ce que `functionToExecute` a levé.

Exemple :

```
from selenium.webdriver.common.by import By

def custom_actions():
```

```
        #verify contains
        browser.find_element(By.XPATH, "//*[@id=\"id_1\"][contains(text(),'login')]")
        #click a button
        browser.find_element(By.XPATH, '//*[@id="submit"]/a').click()

    await synthetics_webdriver.execute_step("verify_click", custom_actions)
```

Chrome()

Lance une instance du navigateur Chromium et renvoie l'instance créée du navigateur.

Exemple :

```
browser = synthetics_webdriver.Chrome()
browser.get("https://example.com/)
```

Pour lancer un navigateur en mode navigation privée, utilisez ce qui suit :

```
add_argument('—-incognito')
```

Pour ajouter des paramètres proxys, utilisez ce qui suit :

```
add_argument('--proxy-server=%s' % PROXY)
```

Exemple :

```
from selenium.webdriver.chrome.options import Options
chrome_options = Options()
chrome_options.add_argument("—-incognito")
browser = syn_webdriver.Chrome(chrome_options=chrome_options)
```

Planification d'exécutions de scripts Canary à l'aide de cron

L'utilisation d'une expression cron vous donne de la flexibilité lorsque vous planifiez un script Canary. Les expressions cron contiennent cinq ou six champs dans l'ordre indiqué dans le tableau suivant. Les champs sont séparés par des espaces. La syntaxe varie selon que vous utilisez la CloudWatch console pour créer le Canary AWS CLI ou les AWS SDK. Lorsque vous utilisez la console, vous spécifiez uniquement les cinq premiers champs. Lorsque vous utilisez les AWS SDK AWS CLI ou, vous spécifiez les six champs, et vous devez spécifier `*` le `Year` champ.

Champ	Valeurs autorisées	Caractères spéciaux autorisés
Minutes	0-59	, - * /
Heures	0-23	, - * /
D ay-of-month	1-31	, - * ? / L W
Mois	1-12 ou JAN-DEC	, - * /
D ay-of-week	1-7 ou DIM-SAM	, - * ? L #
Année	*	

Caractères spéciaux

- Le caractère spécial , (virgule) inclut plusieurs valeurs dans l'expression d'un champ. Dans le champ Month (Mois), JAN,FEB,MAR englobe janvier, février et mars.
- Le caractère spécial - (tiret) spécifie des plages. Dans le champ Jour, 1-15 englobe les jours 1 à 15 du mois spécifié.
- Le caractère spécial * (astérisque) inclut toutes les valeurs du champ. Dans le champ Hours (Heures), * inclut toutes les heures. Vous ne pouvez pas utiliser * à la fois dans les ay-of-week champs D ay-of-month et D d'une même expression. Si vous l'utilisez dans un champ, vous devez utiliser ? dans l'autre.
- Le caractère spécial / (barre oblique) spécifie les incréments. Dans le champ Minutes, vous pouvez saisir 1/10 pour spécifier toutes les dix minutes, à partir de la première minute de l'heure (par exemple, la 11e, 21e et 31e minute, et ainsi de suite).
- Le caractère spécial ? (point d'interrogation) indique l'un ou l'autre. Si vous entrez 7 dans le ay-of-month champ D et que vous ne vous souciez pas du jour de la semaine le septième, pouvez-vous participer ? dans le ay-of-week champ D.
- Le caractère générique L dans les ay-of-week champs D ay-of-month ou D indique le dernier jour du mois ou de la semaine.
- Le **`W`** caractère générique dans le ay-of-month champ D indique un jour de la semaine. Dans le ay-of-month champ D, **`3W`** indique le jour de la semaine le plus proche du troisième jour du mois.
- Le caractère générique # dans le ay-of-week champ D indique une certaine instance du jour de la semaine spécifié dans un délai d'un mois. Par exemple, 3#2 est le deuxième mardi du mois. Le

3 fait référence au mardi, car c'est le troisième jour de chaque semaine et le 2 fait référence à la deuxième journée de ce type dans le mois.

Limites

- Vous ne pouvez pas spécifier les ay-of-week champs D ay-of-month et D dans la même expression cron. Si vous spécifiez une valeur ou le caractère * (astérisque) dans l'un de ces champs, vous devez utiliser un caractère ? (point d'interrogation) dans l'autre.
- Les expressions cron qui entraînent des fréquences d'une rapidité supérieure à une minute ne sont pas prises en charge.
- Vous ne pouvez pas configurer un script Canary pour qu'il attende plus d'un an avant son exécution. Vous ne pouvez donc spécifier que * dans le champ `Year`.

Exemples

Vous pouvez vous référer aux exemples de chaînes cron suivants lorsque vous créez un script Canary. Les exemples suivants présentent la syntaxe correcte pour utiliser les AWS SDK AWS CLI ou pour créer ou mettre à jour un Canary. Si vous utilisez la CloudWatch console, omettez le final * dans chaque exemple.

Expression	Signification
`0 10 * * ? *`	Exécuter à 10 h 00 (UTC) chaque jour
`15 12 * * ? *`	Exécuter à 12 h 15 (UTC) chaque jour
`0 18 ? * MON-FRI *`	Exécuter à 18 h 00 (UTC) du lundi au vendredi
`0 8 1 * ? *`	Exécuter à 8 h 00 (UTC) le premier jour du mois
`0/10 * ? * MON-SAT *`	Exécuter toutes les 10 minutes du lundi au samedi de chaque semaine
`0/5 8-17 ? * MON-FRI *`	Exécuter toutes les 5 minutes du lundi au vendredi entre 8 h 00 et 17 h 55 (UTC)

Groups

Vous pouvez créer des groupes pour associer les versions Canary entre elles, y compris les versions Canary entre régions. L'utilisation de groupes peut vous aider à gérer et à automatiser vos versions Canary, et vous pouvez également afficher les résultats d'exécution et les statistiques agrégés pour toutes les versions Canary d'un groupe.

Les groupes sont des ressources globales. Lorsque vous créez un groupe, il est répliqué dans toutes les AWS régions qui prennent en charge des groupes, et vous pouvez y ajouter des canaris de n'importe laquelle de ces régions et le visualiser dans n'importe laquelle de ces régions. Bien que le format ARN du groupe reflète le nom de la région dans laquelle elle a été créée, le groupe n'est soumis à aucune région. Cela signifie que vous pouvez placer des versions Canary de plusieurs régions dans le même groupe, puis utiliser ce groupe pour afficher et gérer toutes ces versions Canary dans une seule vue.

Les groupes sont pris en charge dans toutes les régions, à l'exception des régions désactivées par défaut. Pour de plus amples informations sur les régions, veuillez consulter la rubrique Enabling a Region (Activer une région).

Chaque groupe peut contenir jusqu'à 10 versions Canary. Vous pouvez avoir jusqu'à 20 groupes dans votre compte. Chaque script Canary peut appartenir à 10 groupes au maximum.

Pour créer un groupe

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Synthetics Canaries.
3. Choisissez Create Group.
4. Pour Group Name (Nom du groupe), saisissez un nom de groupe.
5. Sélectionnez les versions Canary à associer à ce groupe. Pour sélectionner un script Canary, tapez son nom complet dans Exact canary name (Nom exact du script Canary) et choisissez Search (Rechercher). Activez ensuite la case à cocher en regard du nom du script Canary. S'il y a plusieurs versions Canary portant le même nom dans différentes régions, assurez-vous de sélectionner les versions Canary souhaitées.

   Vous pouvez répéter cette étape pour associer jusqu'à 10 versions Canary au groupe.
6. (Facultatif) Sous Tags (Balises), ajoutez une ou plusieurs paires clé/valeur comme balises pour ce groupe. Les balises peuvent vous aider à identifier et à organiser vos AWS ressources et à

suivre vos AWS coûts. Pour plus d'informations, consultez Marquer vos ressources Amazon CloudWatch .

7. Choisissez Create Group.

Testez un canari localement

Cette section explique comment modifier, tester et déboguer les canaries CloudWatch Synthetics directement dans l'éditeur de code ou Microsoft Visual Studio l'éditeur de code. JetBrains IDE L'environnement de débogage local utilise un conteneur SAM (Serverless Application Model) pour simuler une fonction Lambda afin d'émuler le comportement d'un Canary Synthetics.

> **Note**
>
> Il n'est pas pratique de réaliser des canaris de débogage locaux qui reposent sur une surveillance visuelle. La surveillance visuelle repose sur la capture de captures d'écran de base lors d'une première exécution, puis sur la comparaison de ces captures d'écran avec les captures d'écran des exécutions suivantes. Dans un environnement de développement local, les exécutions ne sont ni stockées ni suivies, et chaque itération est une exécution indépendante et autonome. En l'absence d'historique des canaris, il n'est pas pratique de déboguer des canaris qui dépendent d'une surveillance visuelle.

Prérequis

1. Choisissez ou créez un compartiment Amazon S3 à utiliser pour stocker des artefacts issus des tests Canary locaux, tels que des fichiers HAR et des captures d'écran. Cela nécessite que vous soyez approvisionné en IAM. Si vous ne configurez pas les buckets Amazon S3, vous pouvez toujours tester votre Canary localement, mais vous verrez un message d'erreur concernant le bucket manquant et vous n'aurez pas accès aux artefacts Canary.

   Si vous utilisez un compartiment Amazon S3, nous vous recommandons de définir le cycle de vie du compartiment de manière à supprimer les objets au bout de quelques jours, afin de réduire les coûts. Pour plus d'informations, voir Gestion du cycle de vie de votre stockage.
2. Configurez un AWS profil par défaut pour votre AWS compte. Pour plus d'informations, consultez Configuration et paramètres des fichiers d'identification.
3. Définissez la AWS région par défaut de l'environnement de débogage sur votre région préférée, telle que `us-west-2`.

4. Installez la AWS SAM CLI. Pour plus d'informations, consultez la section Installation de la AWS SAM CLI.
5. Installez Visual Studio Code Editor ouJetBrains IDE. Pour plus d'informations, consultez Visual Studio Codeou JetBrains IDE
6. Installez-le Docker pour fonctionner avec la AWS SAM CLI. Assurez-vous de démarrer le daemon docker. Pour plus d'informations, consultez la section Installation Docker à utiliser avec la AWS SAM CLI.

   Vous pouvez également installer un autre logiciel de gestion de conteneurs, par exempleRancher, à condition qu'il utilise le Docker moteur d'exécution.
7. Installez une extension de AWS boîte à outils pour votre éditeur préféré. Pour plus d'informations, voir Installation du AWS Toolkit for Visual Studio Code ou Installation du AWS Toolkit for JetBrains.

Rubriques

Configuration de l'environnement de test et de débogage

Tout d'abord, clonez le dépôt Github qui le AWS fournit en saisissant la commande suivante. Le référentiel contient des exemples de code pour les canaries Node.js et les canaries Python.

```
git clone https://github.com/aws-samples/synthetics-canary-local-debugging-sample.git
```

Effectuez ensuite l'une des opérations suivantes, en fonction de la langue de vos canaris.

Pour les canaris Node.js

1. Accédez au répertoire source Canary de Node.js en saisissant la commande suivante.

```
cd synthetics-canary-local-debugging-sample/nodejs-canary/src
```

2. Entrez la commande suivante pour installer Canary Dependencies.

```
npm install
```

Pour les canaris en Python

1. Accédez au répertoire source de Python Canary en saisissant la commande suivante.

```
cd synthetics-canary-local-debugging-sample/python-canary/src
```

2. Entrez la commande suivante pour installer Canary Dependencies.

```
pip3 install -r requirements.txt -t .
```

Utiliser Visual Studio Code IDE

Le fichier de configuration de Visual Studio lancement se trouve à l'adresse `.vscode/launch.json`. Il contient des configurations permettant au fichier modèle d'être découvert par isual Studio le code V. Il définit une charge utile Lambda avec les paramètres requis pour invoquer le canari avec succès. Voici la configuration de lancement d'un canari Node.js :

```
{
            ...
            ...
            "lambda": {
                "payload": {
                    "json": {
                        // Canary name. Provide any name you like.
                        "canaryName": "LocalSyntheticsCanary",
                        // Canary artifact location
                        "artifactS3Location": {
                            "s3Bucket": "cw-syn-results-123456789012-us-west-2",
                            "s3Key": "local-run-artifacts",
                        },
                        // Your canary handler name
                        "customerCanaryHandlerName": "heartbeat-canary.handler"
                    }
```

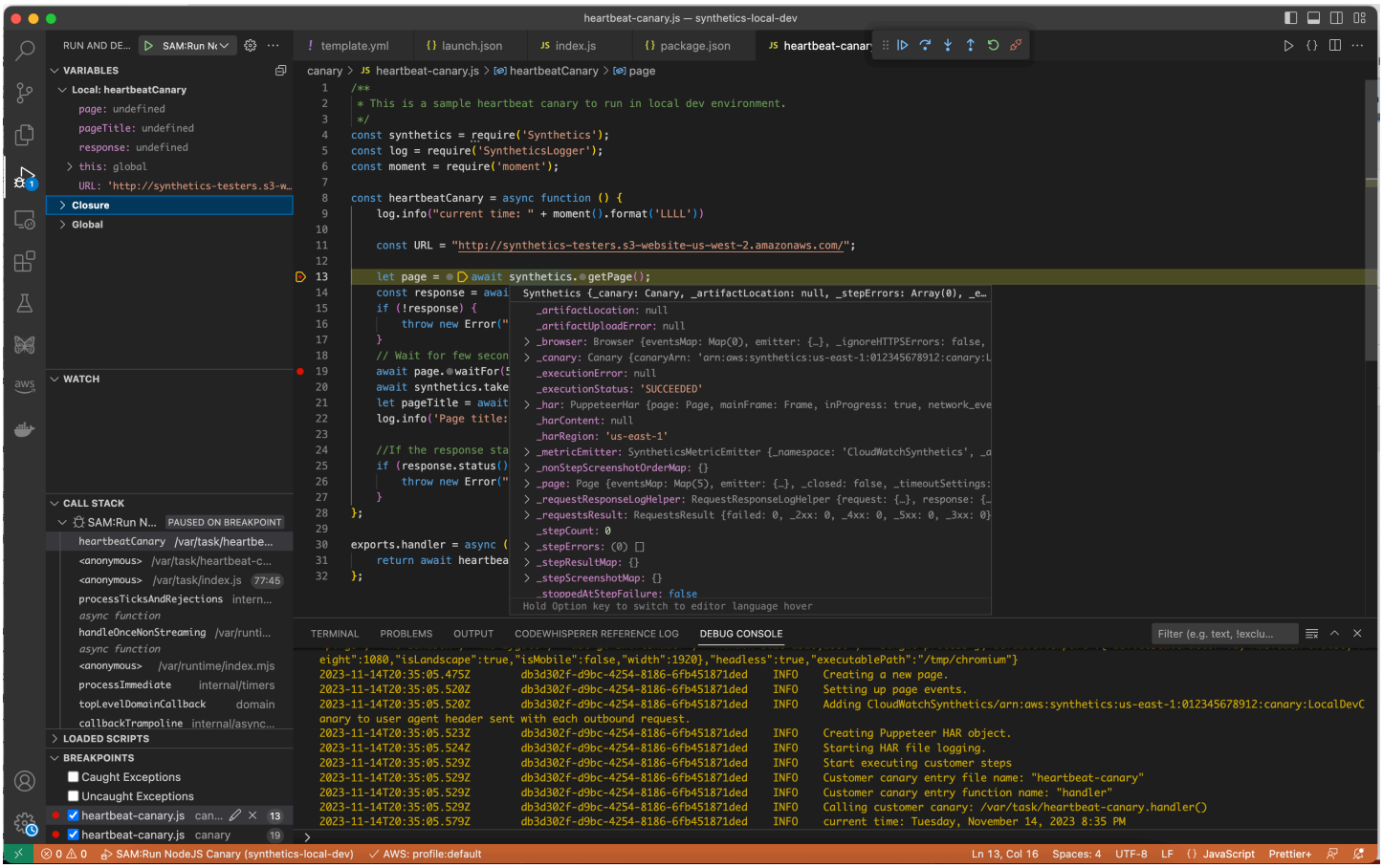
```
                },
                // Environment variables to pass to the canary code
                "environmentVariables": {}
            }
        }
    ]
}
```

Vous pouvez également éventuellement fournir les champs suivants dans le fichier JSON de charge utile :

- `s3EncryptionMode`Valeurs valides : `SSE_S3` | `SSE_KMS`
- `s3KmsKeyArn`Valeur valide : *`KMS Key ARN`*
- `activeTracing`Valeurs valides : `true` | `false`
- `canaryRunId`Valeur valide : *`UUID`* Ce paramètre est obligatoire si le suivi actif est activé.

Pour déboguer le Canary dansVisual Studio, ajoutez des points d'arrêt dans le code Canary où vous souhaitez suspendre l'exécution. Pour ajouter un point d'arrêt, choisissez la marge de l'éditeur et passez en mode Exécution et débogage dans l'éditeur. Lancez le canari en cliquant sur le bouton Play. Lorsque le canary s'exécute, les journaux sont enregistrés dans la console de débogage, vous fournissant des informations en temps réel sur le comportement du canari. Si vous avez ajouté des points d'arrêt, l'exécution de Canary s'interrompt à chaque point d'arrêt, ce qui vous permet de parcourir le code et d'inspecter les valeurs des variables, les méthodes d'instance, les attributs des objets et la pile d'appels de fonctions.

L'exécution et le débogage des canaris en local sont gratuits, à l'exception des artefacts stockés dans le compartiment Amazon S3 et des CloudWatch métriques générées par chaque exécution locale.



Utiliser JetBrains IDE

Une fois l' AWS Toolkit for JetBrains extension installée, assurez-vous que le plug-in et le JavaScript débogueur Node.js sont activés pour s'exécuter, si vous déboguez un canari Node.js. Procédez comme suit.

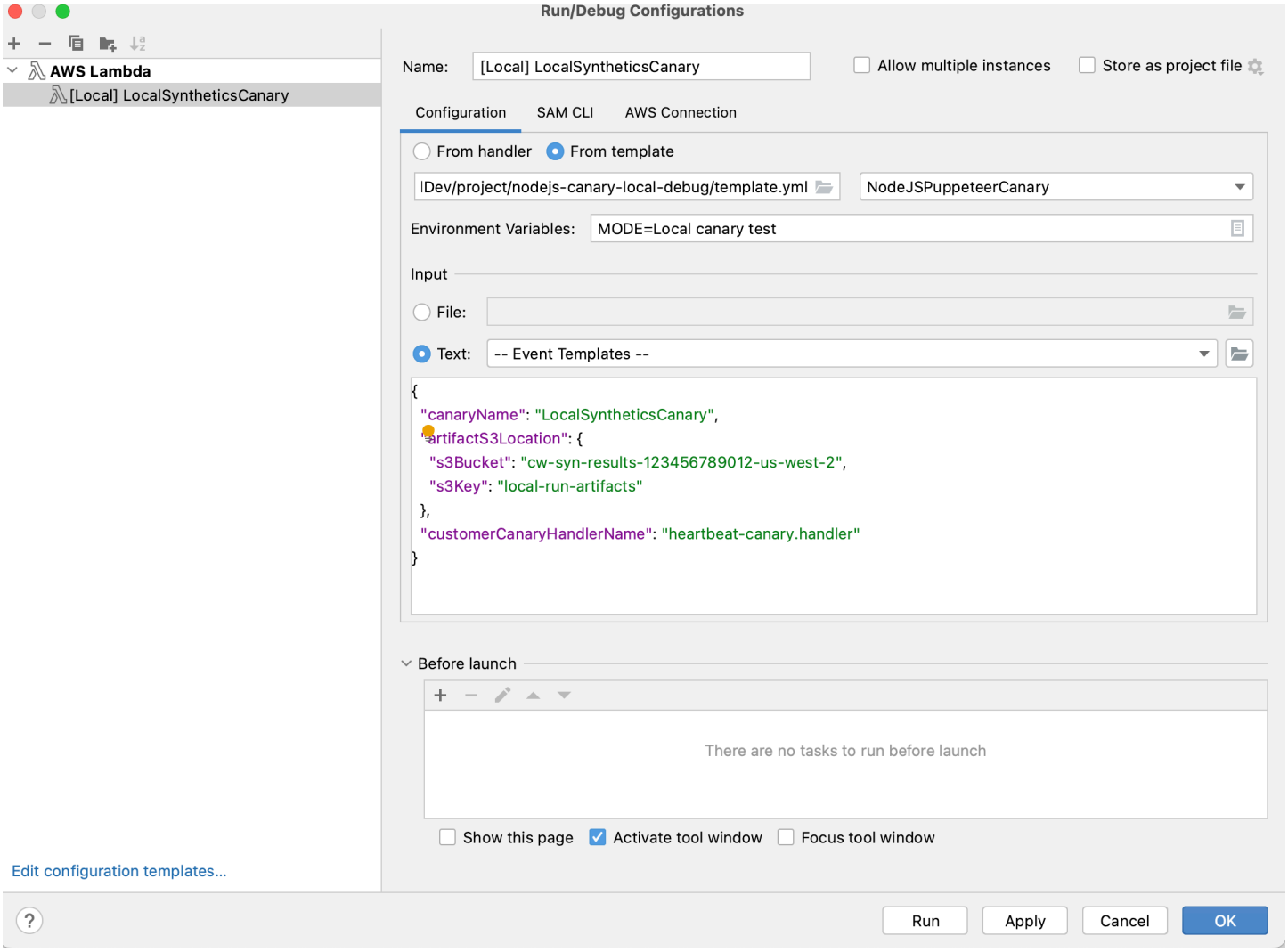
Déboguer un canari en utilisant JetBrains IDE

1. Dans le volet de navigation de gauche deJetBrains IDE, choisissez Lambda, puis choisissez le modèle de configuration local.
2. Entrez un nom pour la configuration d'exécution, tel que **`LocalSyntheticsCanary`**
3. Choisissez À partir du modèle, choisissez le navigateur de fichiers dans le champ du modèle, puis sélectionnez le fichier template.yml dans le projet, soit dans le répertoire nodejs, soit dans le répertoire python.
4. Dans la section Entrée, entrez la charge utile du canari comme indiqué dans l'écran suivant.

```
{
```

```
 "canaryName": "LocalSyntheticsCanary",
 "artifactS3Location": {
     "s3Bucket": "cw-syn-results-123456789012-us-west-2",
     "s3Key": "local-run-artifacts"
 },
 "customerCanaryHandlerName": "heartbeat-canary.handler"
}
```

Vous pouvez également définir d'autres variables d'environnement dans le fichier JSON de charge utile, comme indiqué dansUtiliser Visual Studio Code IDE.



Exécuter un canary en local avec la CLI SAM

Utilisez l'une des procédures suivantes pour exécuter votre Canary localement à l'aide de la CLI SAM (Serverless Application Model). Assurez-vous de spécifier votre propre nom de compartiment Amazon S3 pour `s3Bucket` dans `event.json`

Pour utiliser la CLI SAM pour exécuter un canari Node.js

1. Accédez au répertoire source en saisissant la commande suivante.

```
cd synthetics-canary-local-debugging-sample/nodejs-canary
```

2. Entrez les commandes suivantes :

```
sam build
sam local invoke -e ../event.json
```

Pour utiliser la CLI SAM pour exécuter un canari Python

1. Accédez au répertoire source en saisissant la commande suivante.

```
cd synthetics-canary-local-debugging-sample/python-canary
```

2. Entrez les commandes suivantes :

```
sam build
sam local invoke -e ../event.json
```

Intégrez votre environnement de test local dans un package Canary existant

Vous pouvez intégrer le débogage Canary local dans votre package Canary existant en copiant trois fichiers :

- Copiez le `template.yml` fichier dans la racine de votre package Canary. Assurez-vous de modifier le chemin pour qu'`CodeUri`il pointe vers le répertoire où se trouve votre code Canary.
- Si vous travaillez avec un canari Node.js, copiez le `cw-synthetics.js` fichier dans votre répertoire source Canary. Si vous travaillez avec un canari Python, copiez-le dans `cw-synthetics.py` votre répertoire source Canary.

- Copiez le fichier de configuration de lancement. `vscode/launch.json`dans la racine du package. Assurez-vous de le mettre dans le `.vscode` répertoire ; créez-le s'il n'existe pas déjà.

Modifier le runtime de CloudWatch Synthetics

Dans le cadre de votre débogage, vous pouvez essayer d'exécuter un canary avec un environnement d'exécution CloudWatch Synthetics différent, au lieu du dernier environnement d'exécution. Pour ce faire, recherchez le moteur d'exécution que vous souhaitez utiliser dans l'un des tableaux suivants. Assurez-vous de sélectionner le moteur d'exécution pour la bonne région. Collez ensuite l'ARN de cet environnement d'exécution à l'endroit approprié de votre `template.yml` fichier, puis exécutez le canari.

Environnements d’exécution Node.js

ARN pour -7.0 syn-nodejs-puppeteer

Le tableau suivant répertorie les ARN à utiliser pour la version `syn-nodejs-puppeteer-7.0` du runtime CloudWatch Synthetics dans AWS chaque région où il est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:44`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:46`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:44`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:47`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:44`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:45`

Région	ARN
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:20`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:26`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:18`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:44`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:30`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:46`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:49`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:44`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:44`
Canada (Centre)	`arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:44`
Canada Ouest (Calgary)	`arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:76`
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:45`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:46`

Région	ARN
Europe (Francfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:44
Europe (Irlande)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:46
Europe (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:44
Europe (Milan)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:45
Europe (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:44
Europe (Espagne)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:20
Europe (Stockholm)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:44
Europe (Zurich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:19
Israël (Tel Aviv)	arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:17
Moyen-Orient (Bahreïn)	arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:44
Moyen-Orient (EAU)	arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:19
Amérique du Sud (São Paulo)	arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:45
AWS GovCloud (USA Est)	arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:41

Région	ARN
AWS GovCloud (US-Ouest)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:42`

ARN pour -6.2 syn-nodejs-puppeteer

Le tableau suivant répertorie les ARN à utiliser pour la version `syn-nodejs-puppeteer-6.2` du runtime CloudWatch Synthetics dans AWS chaque région où il est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:41`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:43`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:41`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:44`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:41`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:42`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15`

Région	ARN
Asie-Pacifique (Mumbai)	arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:41
Asie-Pacifique (Osaka)	arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:42
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:46
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:41
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:41
Canada (Centre)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:41
Canada Ouest (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73
Chine (Beijing)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:42
Chine (Ningxia) ;	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:43
Europe (Francfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:41
Europe (Irlande)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:43
Europe (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:41

Région	ARN
Europe (Milan)	`arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:42`
Europe (Paris)	`arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:41`
Europe (Espagne)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:41`
Europe (Zurich)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16`
Israël (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:41`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:42`
AWS GovCloud (USA Est)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:39`
AWS GovCloud (US-Ouest)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:39`

ARN pour -5.2 syn-nodejs-puppeteer

Le tableau suivant répertorie les ARN à utiliser pour la version `syn-nodejs-puppeteer-5.2` du runtime CloudWatch Synthetics dans AWS chaque région où il est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:42`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:44`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:42`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:45`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:42`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:43`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:18`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:24`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:16`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:42`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:28`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:44`

Région	ARN
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:47`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:42`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:42`
Canada (Centre)	`arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:42`
Canada Ouest (Calgary)	`arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:74`
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:43`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:44`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:42`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:44`
Europe (Londres)	`arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:42`
Europe (Milan)	`arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:43`
Europe (Paris)	`arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:42`
Europe (Espagne)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:18`

Région	ARN
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:42`
Europe (Zurich)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:17`
Israël (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:15`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:42`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:17`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:43`
AWS GovCloud (USA Est)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:40`
AWS GovCloud (US-Ouest)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:40`

Environnements d’exécution Python

ARN pour -3.0 syn-python-selenium

Le tableau suivant répertorie les ARN à utiliser pour la version `syn-python-selenium-3.0` du runtime CloudWatch Synthetics dans AWS chaque région où il est disponible.

Région	ARN
USA Est (Virginie du Nord)	`aarn:aws:lambda:us-east-1:378653112637:layer:Synthetics_Selenium:32`

Région	ARN
USA Est (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics_Selenium:34`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics_Selenium:32`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics_Selenium:34`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics_Selenium:32`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics_Selenium:32`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics_Selenium:20`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics_Selenium:26`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics_Selenium:18`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics_Selenium:32`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics_Selenium:30`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics_Selenium:34`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics_Selenium:37`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics_Selenium:32`

Région	ARN
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics_Selenium:32
Canada (Centre)	arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics_Selenium:32
Canada Ouest (Calgary)	arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics_Selenium:76
Chine (Beijing)	arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics_Selenium:32
Chine (Ningxia) ;	arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics_Selenium:32
Europe (Francfort)	arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics_Selenium:32
Europe (Irlande)	arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics_Selenium:34
Europe (Londres)	arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics_Selenium:32
Europe (Milan)	arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics_Selenium:33
Europe (Paris)	arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics_Selenium:32
Europe (Espagne)	arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics_Selenium:20
Europe (Stockholm)	arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics_Selenium:32
Europe (Zurich)	arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics_Selenium:19

Région	ARN
Israël (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics_Selenium:17`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics_Selenium:32`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics_Selenium:19`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics_Selenium:33`
AWS GovCloud (USA Est)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics_Selenium:30`
AWS GovCloud (US-Ouest)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics_Selenium:31`

ARN pour -2.1 syn-python-selenium

Le tableau suivant répertorie les ARN à utiliser pour la version `syn-python-selenium-2.1` du runtime CloudWatch Synthetics dans AWS chaque région où il est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:378653112637:layer:Synthetics:29`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:772927465453:layer:Synthetics:31`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:332033056316:layer:Synthetics:29`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:760325925879:layer:Synthetics:31`

Région	ARN
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:461844272066:layer:Synthetics:29`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:129828061636:layer:Synthetics:29`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:280298676434:layer:Synthetics:17`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:246953257743:layer:Synthetics:23`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:200724813040:layer:Synthetics:15`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:724929286329:layer:Synthetics:29`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:608016332111:layer:Synthetics:27`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:989515803484:layer:Synthetics:30`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:068035103298:layer:Synthetics:34`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:584677157514:layer:Synthetics:29`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:172291836251:layer:Synthetics:29`
Canada (Centre)	`arn:aws:lambda:ca-central-1:236629016841:layer:Synthetics:29`
Canada Ouest (Calgary)	`arn:aws:lambda:ca-west-1:944448206667:layer:Synthetics:73`

Région	ARN
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:422629156088:layer:Synthetics:29`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:474974519687:layer:Synthetics:29`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:122305336817:layer:Synthetics:29`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:563204233543:layer:Synthetics:31`
Europe (Londres)	`arn:aws:lambda:eu-west-2:565831452869:layer:Synthetics:29`
Europe (Milan)	`arn:aws:lambda:eu-south-1:525618516618:layer:Synthetics:30`
Europe (Paris)	`arn:aws:lambda:eu-west-3:469466506258:layer:Synthetics:29`
Europe (Espagne)	`arn:aws:lambda:eu-south-2:029793053121:layer:Synthetics:17`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:162938142733:layer:Synthetics:29`
Europe (Zurich)	`arn:aws:lambda:eu-central-2:224218992030:layer:Synthetics:16`
Israël (Tel Aviv)	`arn:aws:lambda:il-central-1:313249807427:layer:Synthetics:14`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:823195537320:layer:Synthetics:29`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:239544149032:layer:Synthetics:16`

Région	ARN
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:783765544751:layer:Synthetics:30`
AWS GovCloud (USA Est)	`arn:aws-us-gov:lambda:us-gov-east-1:946759330430:layer:Synthetics:29`
AWS GovCloud (US-Ouest)	`arn:aws-us-gov:lambda:us-gov-west-1:946807836238:layer:Synthetics:29`

Erreurs courantes

Erreur : Docker est requis pour exécuter des projets AWS SAM en local. L'avez-vous installé et en cours d'exécution ?

Assurez-vous de démarrer Docker sur votre ordinateur.

Échec de l'appel local SAM : une erreur s'est produite (ExpiredTokenException) lors de l'appel de l' GetLayerVersion opération : le jeton de sécurité inclus dans la demande a expiré

Assurez-vous que le profil AWS par défaut est configuré.

Erreurs les plus courantes

Pour plus d'informations sur les erreurs courantes liées au SAM, consultez la section Résolution des problèmes liés à la CLI AWS SAM.

Dépannage d'un script Canary ayant échoué

Si votre script Canary échoue, vérifiez les points suivants pour le dépannage.

Résolution de problème généraux

- Utilisez la page des détails des scripts Canary pour trouver plus d'informations. Dans la CloudWatch console, choisissez Canaries dans le volet de navigation, puis choisissez le nom du canari pour ouvrir la page de détails du canari. Dans l'onglet Disponibilité, vérifiez la SuccessPercentmétrique pour voir si le problème est constant ou intermittent.

Toujours dans l'onglet Availability (Disponibilité), choisissez un point de données ayant échoué pour afficher les captures d'écran, les journaux et les rapports d'étape (le cas échéant) de cette exécution ayant échoué.

Si un rapport d'étape est disponible car les étapes font partie de votre script, vérifiez l'étape qui a échoué et consultez les captures d'écran associées pour voir le problème rencontré par vos clients.

Vous pouvez également vérifier les fichiers HAR pour voir si une ou plusieurs demandes échouent. Vous pouvez aller plus loin en utilisant les journaux pour explorer les demandes ayant échoué et les erreurs. Enfin, vous pouvez comparer ces artefacts avec les artefacts d'une exécution de script canary réussie pour identifier le problème.

Par défaut, CloudWatch Synthetics capture des captures d'écran pour chaque étape d'un canari d'interface utilisateur. Toutefois, votre script peut être configuré de manière à désactiver les captures d'écran. Pendant le débogage, vous devrez peut-être activer à nouveau les captures d'écran. De même, pour les scripts Canary d'API, vous voudrez peut-être voir les en-têtes et le corps de requête et de réponse HTTP pendant le débogage. Pour plus d'informations sur la manière d'inclure ces données dans le rapport, consultez executeHttpStep(StepName, RequestOptions, [rappel], [StepConfig]).

- S'il y a eu un déploiement récent dans votre application, annulez-le, puis déboguez plus tard.
- Connectez-vous manuellement à votre point de terminaison pour voir si vous pouvez reproduire le même problème.

Rubriques

- Problèmes de mise à niveau et de rétrogradation des versions d'exécution des scripts Canary
- Problème CORS (partage des demandes cross-origin)
- Problèmes de condition de la race Canary
- Dépannage d'un script Canary sur un VPC

Canary échoue après la mise à jour de l'environnement Lambda

CloudWatch Les canaris Synthetics sont implémentés sous forme de fonctions Lambda dans votre compte. Ces fonctions Lambda font l'objet de mises à jour régulières du runtime Lambda contenant des mises à jour de sécurité, des corrections de bogues et d'autres améliorations. Lambda s'efforce de fournir des mises à jour d'exécution rétrocompatibles avec les fonctions existantes. Cependant, comme pour les correctifs logiciels, il existe de rares cas dans lesquels une mise à jour de l'environnement d'exécution peut avoir un impact négatif sur une fonction existante. Si vous pensez que votre Canary a été affecté par une mise à jour du runtime Lambda, vous pouvez utiliser le mode manuel de gestion du runtime Lambda (dans les régions prises en charge) pour annuler temporairement la version d'exécution Lambda. Cela permet à votre fonction Canary de fonctionner et de minimiser les perturbations, ce qui vous laisse le temps de remédier à l'incompatibilité avant de revenir à la dernière version d'exécution.

Si votre Canary échoue après une mise à jour de l'environnement d'exécution Lambda, la meilleure solution consiste à passer à l'un des derniers environnements d'exécution Synthetics. Pour plus d'informations sur les derniers environnements d'exécution, consultezVersions d'exécution Synthetics.

Comme solution alternative, dans les régions où les contrôles de gestion d'exécution Lambda sont disponibles, vous pouvez rétablir un environnement d'exécution géré par Lambda sur un canary, en utilisant le mode manuel pour les contrôles de gestion d'exécution. Vous pouvez définir le mode manuel à l'aide de la console Lambda AWS CLI ou à l'aide de celle-ci, en suivant les étapes ci-dessous dans les sections suivantes.

> **Warning**
>
> Lorsque vous modifiez les paramètres d'exécution en mode manuel, votre fonction Lambda ne reçoit pas de mises à jour de sécurité automatiques tant qu'elle n'est pas revenue en mode automatique. Au cours de cette période, votre fonction Lambda peut être vulnérable à des failles de sécurité.

Prérequis

- Installer jq
- Installez la dernière version du AWS CLI. Pour plus d'informations, consultez les instructions AWS CLI d'installation et de mise à jour.

Étape 1 : Obtenir l'ARN de la fonction Lambda

Exécutez la commande suivante pour récupérer le `EngineArn` champ de la réponse. Il `EngineArn` s'agit de l'ARN de la fonction Lambda associée au canari. Vous allez utiliser cet ARN dans les étapes suivantes.

```
aws synthetics get-canary --name my-canary | jq '.Canary.EngineArn'
```

Exemple de sortie de `EngingArn` :

```
"arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-dc5015c2-db17-4cb5-
afb1-EXAMPLE991:8"
```

Étape 2 : Obtenir le dernier ARN valide de la version d'exécution Lambda

Pour savoir si votre Canary a été impacté par une mise à jour de l'environnement d'exécution Lambda, vérifiez si la date et l'heure auxquelles l'ARN de la version d'exécution Lambda a changé dans vos journaux correspondent à la date et à l'heure auxquelles vous avez constaté un impact sur votre Canary. S'ils ne correspondent pas, ce n'est probablement pas une mise à jour du moteur d'exécution Lambda qui est à l'origine de vos problèmes.

Si votre Canary est concerné par une mise à jour du moteur d'exécution Lambda, vous devez identifier l'ARN de la version fonctionnelle du moteur d'exécution Lambda que vous utilisiez auparavant. Suivez les instructions de la section Identification des modifications de version d'exécution pour trouver l'ARN de la version d'exécution précédente. Enregistrez l'ARN de la version d'exécution et passez à l'étape 3 pour définir la configuration de gestion de l'exécution.

Si votre Canary n'a pas encore été affecté par une mise à jour de l'environnement Lambda, vous pouvez trouver l'ARN de la version d'exécution Lambda que vous utilisez actuellement. Exécutez la commande suivante pour récupérer `RuntimeVersionArn` la fonction Lambda à partir de la réponse.

```
aws lambda get-function-configuration \
```

```
--function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991:8" | jq '.RuntimeVersionConfig.RuntimeVersionArn'
```

Exemple de sortie de `RuntimeVersionArn` :

```
"arn:aws:lambda:us-
west-2::runtime:EXAMPLE647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Étape 3 : mise à jour de la configuration de gestion du runtime Lambda

Vous pouvez utiliser la console Lambda AWS CLI ou la console Lambda pour mettre à jour la configuration de gestion de l'exécution.

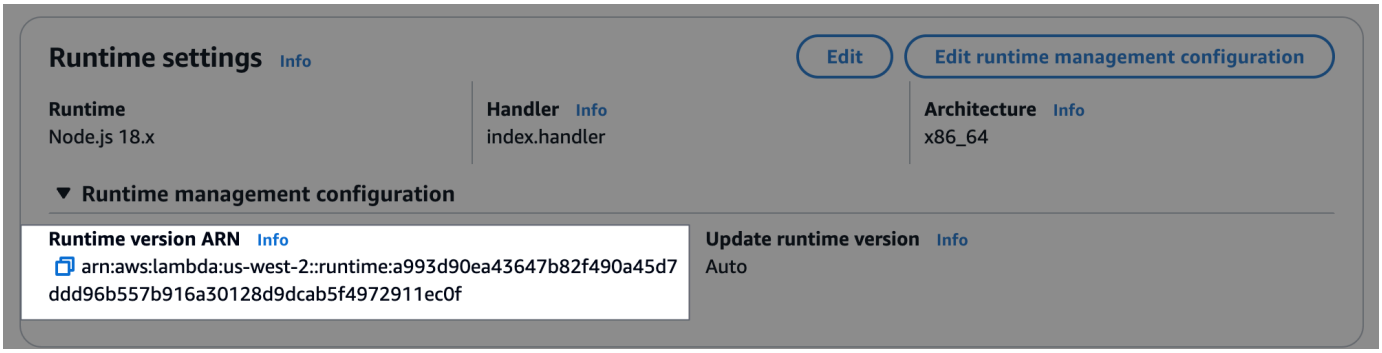
Pour définir le mode manuel de configuration de la gestion des environnements d'exécution Lambda à l'aide du AWS CLI

Entrez la commande suivante pour passer en mode manuel de la gestion de l'exécution de la fonction Lambda. Assurez-vous de remplacer le *nom et le qualificatif de la fonction* Lambda par l'ARN de la fonction Lambda et le numéro de version de la fonction Lambda respectivement, en utilisant les valeurs que vous avez trouvées à l'étape 1. Remplacez également le *runtime-version-arn*par l'ARN de version que vous avez trouvé à l'étape 2.

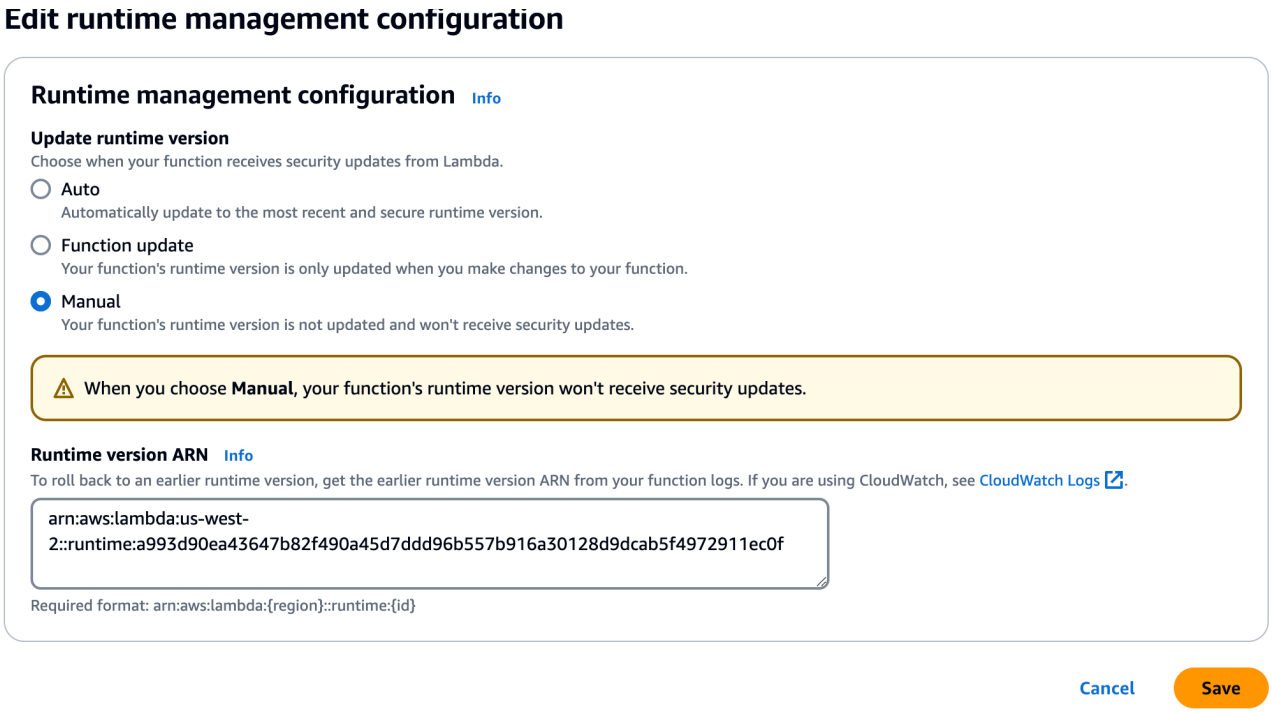
```
aws lambda put-runtime-management-config \
    --function-name "arn:aws:lambda:us-west-2:123456789012:function:cwsyn-my-canary-
dc5015c2-db17-4cb5-afb1-EXAMPLE991" \
    --qualifier 8 \
    --update-runtime-on "Manual" \
    --runtime-version-arn "arn:aws:lambda:us-
west-2::runtime:a993d90ea43647b82f490a45d7ddd96b557b916a30128d9dcab5f4972911ec0f"
```

Pour passer un Canary en mode manuel à l'aide de la console Lambda

1. Ouvrez la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/.
2. Choisissez l'onglet Versions, choisissez le lien du numéro de version correspondant à votre ARN, puis cliquez sur l'onglet Code.
3. Faites défiler l'écran jusqu'à Paramètres d'exécution, développez la configuration de gestion du temps d'exécution et copiez l'ARN de la version d'exécution.



4. Choisissez Modifier la configuration de gestion d'exécution, choisissez Manuel, collez l'ARN de la version d'exécution que vous avez copié précédemment dans le champ ARN de la version d'exécution. Ensuite, choisissez Save (Enregistrer).



Mon canari est bloqué par AWS WAF

Pour éviter AWS WAF de bloquer votre canari, configurez une condition de correspondance des AWS WAF chaînes qui autorise la chaîne`CloudWatchSynthetics`. Pour plus d'informations, consultez la section Utilisation des conditions de correspondance de chaînes dans la AWS WAF documentation.

Attente de l'apparition d'un élément

Après avoir analysé vos journaux et captures d'écran, si vous constatez que votre script attend qu'un élément apparaisse à l'écran et qu'il expire, vérifiez la capture d'écran appropriée pour voir si l'élément apparaît sur la page. Vérifiez votre `xpath` pour vous assurer qu'il est correct.

Pour les problèmes liés à Puppeteer, consultez la page de Puppeteer ou les forums Internet. GitHub

Le nœud n'est pas visible ou n'est pas un HTMLElement pour page.click()

Si un nœud n'est pas visible ou n'est pas un `HTMLElement` pour `page.click()`, vérifiez d'abord le `xpath` que vous utilisez pour cliquer sur l'élément. De plus, si votre élément se trouve en bas de l'écran, ajustez votre fenêtre d'affichage. CloudWatch Synthetics utilise par défaut une fenêtre d'affichage de 1920 x 1080. Vous pouvez définir une fenêtre d'affichage différente lorsque vous lancez le navigateur ou à l'aide de la fonction Puppeteer `page.setViewport`.

Impossible de télécharger des artefacts dans S3 (Exception : Unable to fetch S3 bucket location: Access Denied)

Si votre Canary échoue à cause d'une erreur Amazon S3, CloudWatch Synthetics n'a pas pu télécharger les captures d'écran, les journaux ou les rapports créés pour le Canary en raison de problèmes d'autorisation. Vérifiez les éléments suivants :

- Vérifiez que le rôle IAM du script Canary possède l'autorisation `s3:ListAllMyBuckets`, l'autorisation `s3:GetBucketLocation` pour le compartiment Amazon S3 approprié, et l'autorisation `s3:PutObject` pour le compartiment dans lequel le script Canary stocke ses artefacts. Si le script Canary effectue une surveillance visuelle, le rôle a également besoin de l'autorisation `s3:GetObject` pour le compartiment. Ces mêmes autorisations sont également requises dans la politique de point de terminaison de passerelle Amazon VPC S3, si le script canary est déployé dans un VPC doté d'un point de terminaison d'un VPC.
- Si le Canary utilise une clé gérée par le AWS KMS client pour le chiffrement au lieu de la clé AWS gérée standard (par défaut), le rôle IAM du Canary n'est peut-être pas autorisé à chiffrer ou à déchiffrer à l'aide de cette clé. Pour plus d'informations, consultez Chiffrement des artefacts de script Canary.
- Votre politique de compartiment peut ne pas autoriser le mécanisme de chiffrement utilisé par le script Canary. Par exemple, si votre politique de compartiment exige d'utiliser un mécanisme de chiffrement spécifique ou une clé KMS, vous devez sélectionner le même mode de chiffrement pour votre script Canary.

Si le script Canary effectue une surveillance visuelle, consultez Mise à jour de l'emplacement et du chiffrement des artefacts en utilisant la surveillance visuelle pour en savoir plus.

Erreur : erreur de protocole (exécution). callFunctionOn) : Cible fermée.

Cette erreur apparaît s'il y a des demandes réseau après la fermeture de la page ou du navigateur. Vous avez peut-être oublié d'attendre une opération asynchrone. Après avoir exécuté votre script, CloudWatch Synthetics ferme le navigateur. L'exécution de toute opération asynchrone après la fermeture du navigateur peut provoquer `target closed error`.

Canary Failed. Error: No datapoint – Le script Canary affiche une erreur de dépassement de délai d'attente

Cela signifie que l'exécution de votre script Canary a dépassé le délai d'attente. L'exécution de Canary s'est arrêtée avant que CloudWatch Synthetics puisse publier des indicateurs de réussite ou mettre à jour CloudWatch des artefacts tels que des fichiers HAR, des journaux et des captures d'écran. Si votre délai d'attente est trop court, vous pouvez l'augmenter.

Par défaut, la valeur de délai d'attente d'un script Canary est égale à sa fréquence. Vous pouvez ajuster manuellement la valeur du délai d'attente pour qu'elle soit inférieure ou égale à la fréquence du script Canary. Si la fréquence de votre script Canary est faible, vous devez l'augmenter pour augmenter le délai d'attente. Vous pouvez ajuster à la fois la fréquence et le délai d'expiration dans Schedule lorsque vous créez ou mettez à jour un Canary à l'aide de la console CloudWatch Synthetics.

Vérifiez que la valeur du délai d'attente de votre script canary n'est pas inférieure à 15 secondes pour permettre les démarrages à froid Lambda et le temps nécessaire pour démarrer l'instrumentation canary.

Les artefacts Canary ne peuvent pas être consultés dans la console CloudWatch Synthetics lorsque cette erreur se produit. Vous pouvez utiliser CloudWatch Logs pour voir les logs du canari.

Pour utiliser CloudWatch les journaux pour voir les journaux d'un canari

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, choisissez Log groups (Groupes de journaux).
3. Recherchez le groupe de journaux en saisissant le nom du script Canary dans la zone de filtre. Les groupes de journaux pour les scripts Canary ont le nom /aws/lambda/cwsyn-***canaryName***-randomId.

Tentative d'accès à un point de terminaison interne

Si vous souhaitez que votre Canary accède à un point de terminaison de votre réseau interne, nous vous recommandons de configurer CloudWatch Synthetics pour utiliser le VPC. Pour plus d'informations, consultez Exécution d'un script Canary sur un VPC.

Problèmes de mise à niveau et de rétrogradation des versions d'exécution des scripts Canary

Si vous avez récemment mis à niveau le script Canary de la version d'exécution `syn-1.0` vers une version ultérieure, il peut s'agir d'un problème CORS (partage des demandes cross-origin). Pour plus d'informations, consultez Problème CORS (partage des demandes cross-origin).

Si vous avez récemment rétrogradé Canary vers une ancienne version d'exécution, assurez-vous que les fonctions Synthetics que vous utilisez sont disponibles dans CloudWatch l'ancienne version d'exécution vers laquelle vous avez rétrogradé. Par exemple, la fonction `executeHttpStep` est disponible pour l'exécution `syn-nodejs-2.2` et version ultérieure. Pour vérifier la disponibilité des fonctions, consultez Écriture d'un script Canary.

> **Note**
>
> Lorsque vous envisagez de mettre à niveau ou de rétrograder la version d'exécution d'un Canary, nous vous recommandons de cloner d'abord le Canary et de mettre à jour la version d'exécution dans le Canary cloné. Une fois que vous avez vérifié que le clone fonctionne avec la nouvelle version d'exécution, vous pouvez mettre à jour la version d'exécution de votre script Canary d'origine et supprimer le clone.

Problème CORS (partage des demandes cross-origin)

Dans un script Canary d'interface utilisateur, si certaines demandes réseau échouent avec une erreur `403` ou `net::ERR_FAILED`, vérifiez si le suivi actif est activé pour le script Canary et si ce dernier utilise également la fonction Puppeteer `page.setExtraHTTPHeaders` pour ajouter des en-têtes. Si c'est le cas, les demandes réseau peuvent échouer en raison de restrictions CORS (partage des demandes cross-origin). Vous pouvez confirmer si c'est le cas en désactivant le suivi actif ou en supprimant les en-têtes HTTP supplémentaires.

Pourquoi cela se produit-il ?

Lorsque le suivi actif est utilisé, un en-tête supplémentaire est ajouté à toutes les demandes sortantes pour suivre l'appel. La modification des en-têtes de requête en ajoutant un en-tête de trace ou en ajoutant des en-têtes supplémentaires à l'aide de Puppeteer `page.setExtraHTTPHeaders` entraîne une vérification CORS des requêtes XML HttpRequest (XHR).

Si vous ne souhaitez pas désactiver le suivi actif ou supprimer les en-têtes supplémentaires, vous pouvez mettre à jour votre application web pour autoriser l'accès cross-origin ou désactiver la sécurité web à l'aide de l'indicateur `disable-web-security` lorsque vous lancez le navigateur Chrome dans votre script.

Vous pouvez remplacer les paramètres de lancement utilisés par CloudWatch Synthetics et transmettre des paramètres d'indicateur `disable-web-security` supplémentaires à l'aide de la fonction de lancement de Synthetics. CloudWatch Pour plus d'informations, consultez Fonctions de bibliothèque disponibles pour les scripts Canary Node.js.

> **Note**
>
> Vous pouvez remplacer les paramètres de lancement utilisés par CloudWatch Synthetics lorsque vous utilisez la version d'exécution ou une version ultérieure. `syn-nodejs-2.1`

Problèmes de condition de la race Canary

Pour une expérience optimale lors de l'utilisation de CloudWatch Synthetics, assurez-vous que le code écrit pour les canaris est idempotent. Sinon, dans de rares cas, les courses à canaris peuvent rencontrer des conditions de course lorsque le canari interagit avec la même ressource lors de différentes courses.

Dépannage d'un script Canary sur un VPC

Si vous rencontrez des problèmes après la création ou la mise à jour d'un script Canary sur un VPC, l'une des sections suivantes pourrait vous aider à résoudre le problème.

Nouveau script Canary en état d'erreur ou échec de mise à jour du script Canary

Si vous créez un script Canary à exécuter sur un VPC et qu'il obtient immédiatement un état d'erreur, ou si vous ne pouvez pas mettre à jour un script Canary à exécuter sur un VPC, le rôle du script Canary peut ne pas avoir les autorisations appropriées. Pour s'exécuter sur un VPC, un script Canary doit disposer des autorisations `ec2:CreateNetworkInterface`,

`ec2:DescribeNetworkInterfaces` et `ec2:DeleteNetworkInterface`. Ces autorisations sont toutes contenues dans la politique `AWSLambdaVPCAccessExecutionRole` gérée. Pour de plus amples informations, veuillez consulter Rôle d'exécution et autorisations utilisateur.

Si ce problème s'est produit lorsque vous avez créé un script Canary, vous devez supprimer le script Canary et en créer un nouveau. Si vous utilisez la CloudWatch console pour créer le nouveau canary, sous Autorisations d'accès, sélectionnez Créer un nouveau rôle. Un nouveau rôle est créé qui inclut toutes les autorisations requises pour exécuter le script Canary.

Si ce problème se produit lorsque vous mettez à jour un script Canary, vous pouvez à nouveau mettre à jour le script Canary et fournir un nouveau rôle qui dispose des autorisations requises.

Erreur de type « No test result returned » (aucun résultat de test retourné)

Si un script Canary affiche une erreur de type « aucun résultat de test retourné », l'un des problèmes suivants peut en être la cause :

- Si votre VPC n'a pas accès à Internet, vous devez utiliser des points de terminaison VPC pour autoriser Canary à accéder à Amazon S3 et à Amazon S3. CloudWatch Vous devez activer les options Résolution DNS et Nom d'hôte DNS dans le VPC pour que ces adresses de point de terminaison soient résolues correctement. Pour plus d'informations, consultez les sections Utilisation du DNS avec votre VPC et Utilisation et CloudWatch synthèse des points de terminaison VPC d' CloudWatch interface.
- Les scripts Canary doivent s'exécuter dans des sous-réseaux privés au sein d'un VPC. Pour vérifier cela, ouvrez la page Sous-réseaux dans la console VPC. Vérifiez les sous-réseaux que vous avez sélectionnés lors de la configuration du script Canary. S'ils ont un chemin d'accès à une passerelle Internet (igw-), ce ne sont pas des sous-réseaux privés.

Pour mieux résoudre ces problèmes, veuillez consulter les journaux pour le script Canary.

Pour consulter les événements de journalisation à partir d'un script Canary

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 script Canary. Le nom du groupe de journaux commence par `/aws/lambda/cwsyn-`*`canary-name`*.

Exemple de code pour les scripts Canary

Cette section contient des exemples de code illustrant certaines fonctions possibles pour les scripts CloudWatch Canary de Synthetics.

Exemples pour Node.js et Puppeteer

Définition des cookies

Les sites web s'appuient sur des cookies pour fournir des fonctionnalités personnalisées ou suivre les utilisateurs. En configurant des cookies dans CloudWatch les scripts Synthetics, vous pouvez imiter ce comportement personnalisé et le valider.

Par exemple, un site web peut afficher un lien Connexion pour un utilisateur qui a déjà consulté le site au lieu d'un lien S'enregistrer.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const pageLoadBlueprint = async function () {

    let url = "http://smile.amazon.com/";

    let page = await synthetics.getPage();

    // Set cookies.  I found that name, value, and either url or domain are required
 fields.
    const cookies = [{
      'name': 'cookie1',
      'value': 'val1',
      'url': url
    },{
      'name': 'cookie2',
      'value': 'val2',
      'url': url
    },{
      'name': 'cookie3',
      'value': 'val3',
      'url': url
    }];

    await page.setCookie(...cookies);
```

```
    // Navigate to the url
    await synthetics.executeStep('pageLoaded_home', async function (timeoutInMillis =
 30000) {

        var response = await page.goto(url, {waitUntil: ['load', 'networkidle0'],
 timeout: timeoutInMillis});

        // Log cookies for this page and this url
        const cookiesSet = await page.cookies(url);
        log.info("Cookies for url: " + url + " are set to: " +
 JSON.stringify(cookiesSet));
    });

};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Émulation d'appareils

Vous pouvez écrire des scripts qui émulent divers appareils pour obtenir un aperçu de l'apparence et du comportement d'une page sur ces appareils.

L'exemple suivant émule un iPhone 6. Pour de plus amples informations sur l'émulation, veuillez consulter [page.emulate(options)](#) dans la documentation Puppeteer.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');
const puppeteer = require('puppeteer-core');

const pageLoadBlueprint = async function () {

    const iPhone = puppeteer.devices['iPhone 6'];

    // INSERT URL here
    const URL = "https://amazon.com";

    let page = await synthetics.getPage();
    await page.emulate(iPhone);

    //You can customize the wait condition here. For instance,
    //using 'networkidle2' may be less restrictive.
```

```
    const response = await page.goto(URL, {waitUntil: 'domcontentloaded', timeout:
 30000});
    if (!response) {
        throw "Failed to load page!";
    }

    await page.waitFor(15000);

    await synthetics.takeScreenshot('loaded', 'loaded');

    //If the response status code is not a 2xx success code
    if (response.status() < 200 || response.status() > 299) {
        throw "Failed to load page!";
    }
};

exports.handler = async () => {
    return await pageLoadBlueprint();
};
```

Script Canary d'API à plusieurs étapes

Cet exemple de code illustre un script Canary d'API avec deux étapes HTTP afin de tester la même API pour les cas de tests positifs et négatifs. La configuration des étapes est transmise pour permettre la création de rapports sur les en-têtes de demande/réponse. En outre, il masque l'en-tête Authorization et X-Amz-Security-Token, car ils contiennent des informations d'identification de l'utilisateur.

Lorsque ce script est utilisé en tant que script Canary, vous pouvez afficher des détails sur chaque étape et les demandes HTTP associées, tels que la réussite/l'échec de l'étape, la durée et les métriques de performances telles que le temps de recherche DNS et le temps du premier octet. Vous pouvez voir le nombre de 2xx, 4xx et 5xx pour l'exécution de votre script Canary.

```
var synthetics = require('Synthetics');
const log = require('SyntheticsLogger');

const apiCanaryBlueprint = async function () {

    // Handle validation for positive scenario
    const validatePositiveCase = async function(res) {
        return new Promise((resolve, reject) => {
```

```
            if (res.statusCode < 200 || res.statusCode > 299) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            let responseBody = '';
            res.on('data', (d) => {
                responseBody += d;
            });

            res.on('end', () => {
                // Add validation on 'responseBody' here if required. For ex, your
 status code is 200 but data might be empty
                resolve();
            });
        });
    };

    // Handle validation for negative scenario
    const validateNegativeCase = async function(res) {
        return new Promise((resolve, reject) => {
            if (res.statusCode < 400) {
                throw res.statusCode + ' ' + res.statusMessage;
            }

            resolve();
        });
    };

    let requestOptionsStep1 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/product/validProductName',
        'port': 443,
        'protocol': 'https:'
    };

    let headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep1['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
```

```
    // Change the configuration at global level or add as step configuration for
 individual steps
    let stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };


    await synthetics.executeHttpStep('Verify GET products API with valid name',
 requestOptionsStep1, validatePositiveCase, stepConfig);

    let requestOptionsStep2 = {
        'hostname': 'myproductsEndpoint.com',
        'method': 'GET',
        'path': '/test/canary/InvalidName(',
        'port': 443,
        'protocol': 'https:'
    };

    headers = {};
    headers['User-Agent'] = [synthetics.getCanaryUserAgentString(), headers['User-
Agent']].join(' ');

    requestOptionsStep2['headers'] = headers;

    // By default headers, post data and response body are not included in the report
 for security reasons.
    // Change the configuration at global level or add as step configuration for
 individual steps
    stepConfig = {
        includeRequestHeaders: true,
        includeResponseHeaders: true,
        restrictedHeaders: ['X-Amz-Security-Token', 'Authorization'], // Restricted
 header values do not appear in report generated.
        includeRequestBody: true,
        includeResponseBody: true
    };

    await synthetics.executeHttpStep('Verify GET products API with invalid name',
 requestOptionsStep2, validateNegativeCase, stepConfig);
```

```
};

exports.handler = async () => {
    return await apiCanaryBlueprint();
};
```

Exemples pour Python et Selenium

L'exemple de code Selenium suivant est un script Canary qui échoue avec un message d'erreur personnalisé lorsqu'un élément cible n'est pas chargé.

```
from aws_synthetics.selenium import synthetics_webdriver as webdriver
from aws_synthetics.common import synthetics_logger as logger
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By

def custom_selenium_script():
    # create a browser instance
    browser = webdriver.Chrome()
    browser.get('https://www.example.com/')
    logger.info('navigated to home page')
    # set cookie
    browser.add_cookie({'name': 'foo', 'value': 'bar'})
    browser.get('https://www.example.com/')
    # save screenshot
    browser.save_screenshot('signed.png')
    # expected status of an element
    button_condition = EC.element_to_be_clickable((By.CSS_SELECTOR, '.submit-button'))
    # add custom error message on failure
    WebDriverWait(browser, 5).until(button_condition, message='Submit button failed to
 load').click()
    logger.info('Submit button loaded successfully')
    # browser will be quit automatically at the end of canary run,
    # quit action is not necessary in the canary script
    browser.quit()

# entry point for the canary
def handler(event, context):
    return custom_selenium_script()
```

Scripts Canary et suivi X-Ray

Vous pouvez choisir d'activer le AWS X-Ray suivi actif sur les canaris qui utilisent le runtime `syn-nodejs-2.0` ou une version ultérieure. Lorsque le suivi est activé, des traces sont envoyées pour tous les appels effectués par le Canary qui utilisent le navigateur, le AWS SDK ou les modules HTTP ou HTTPS. Les scripts canary dont le suivi est activé apparaissent sur la Carte de suivi X-Ray et dans Application Signals une fois que vous l'avez activé pour votre application.

> **Note**
>
> L'activation du suivi aux X-Ray sur les canarys n'est pas encore prise en charge en Asie-Pacifique (Jakarta).

Lorsqu'un script Canary apparaît sur la carte de suivi X-Ray, il apparaît comme un nouveau type de nœud client. Vous pouvez passer la souris sur un nœud de script Canary pour afficher les données relatives à la latence, aux demandes et aux défaillances. Vous pouvez également choisir le nœud de script Canary pour afficher plus de données en bas de la page. Dans cette zone de la page, vous pouvez choisir Afficher dans Synthetics pour accéder à la console Synthetics pour obtenir plus de détails sur le canari, ou choisir Afficher les traces pour obtenir plus de détails sur les traces des courses de ce canari. CloudWatch

Un script Canary dont le suivi est activé a également un onglet Tracing (Suivi) dans sa page de détails, avec des détails sur les suivis et les segments des exécutions du script Canary.

L'activation du suivi augmente le temps d'exécution des scripts Canary de 2,5 % à 7 %.

Un script Canary dont le suivi est activé doit utiliser un rôle avec les autorisations suivantes. Si vous utilisez la console pour créer le rôle lorsque vous créez le script Canary, il dispose de ces autorisations.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Sid230934",
            "Effect": "Allow",
            "Action": [
                "xray:PutTraceSegments"
```

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

Les suivis générés par les scripts Canary entraînent des frais. Pour plus d'informations sur la tarification de X-Ray, consultez Tarification d'AWS X-Ray.

Exécution d'un script Canary sur un VPC

Vous pouvez exécuter des scripts Canary sur des points de terminaison sur un VPC, ainsi que des points de terminaison internes publics. Pour exécuter un script Canary sur un VPC, les options Résolution DNS et Noms d'hôte DNS doivent être activées sur le VPC. Pour plus d'informations, consultez Utilisation de DNS avec votre VPC.

Lorsque vous exécutez un Canary sur un point de terminaison VPC, vous devez lui fournir un moyen d'envoyer ses métriques CloudWatch et ses artefacts à Amazon S3. Si le VPC est déjà activé pour l'accès à Internet, il n'y a rien de plus à faire. Le script Canary s'exécute dans votre VPC, mais peut accéder à Internet pour charger ses métriques et ses artefacts.

Si le VPC n'est pas encore activé pour l'accès Internet, vous avez deux options :

- Activez-le pour l'accès Internet. Pour plus d'informations, consultez la section Donner accès à Internet à votre script Canary sur un VPC suivante.
- Si vous souhaitez que votre VPC reste privé, vous pouvez configurer le Canary pour qu'il envoie ses données à Amazon S3 via des CloudWatch points de terminaison VPC privés. Si ce n'est pas déjà fait, vous devez créer un point de terminaison VPC pour CloudWatch (com.amazonaws). *region* .monitoring) et un point de terminaison de passerelle pour Amazon S3. Pour de plus amples informations, veuillez consulter Utilisation CloudWatch et CloudWatch synthèse des points de terminaison VPC d'interface et Points de terminaison Amazon VPC pour Amazon S3.

Donner accès à Internet à votre script Canary sur un VPC

Suivez ces étapes pour donner accès à Internet à votre VPC Canary ou pour attribuer à votre Canary une adresse IP statique

Pour donner accès à Internet à un script Canary sur un VPC

1. Créez une passerelle NAT dans un sous-réseau public sur le VPC. Pour obtenir des instructions, consultez Create a NAT gateway (Créer une passerelle NAT).
2. Ajoutez un nouvel acheminement à la table de routage dans le sous-réseau privé où le scirpt Canary est lancé. Spécifiez les paramètres suivants :

   - Pour Destination, saisissez **`0.0.0.0/0`**
   - Pour Cible, choisissez Passerelle NAT, puis sélectionnez l'ID de la passerelle NAT que vous avez créée.
   - Choisissez Save routes (Enregistrer les acheminements).

   Pour plus d'informations sur l'ajout d'acheminements à la table de routage, consultez Add and remove routes from a route table (Ajout et retrait d'acheminements d'une table de routage).

> **Note**
>
> Assurez-vous que les acheminements vers votre passerelle NAT sont à l'état active (actif). Si la passerelle NAT est supprimée et que vous n'avez pas mis à jour les acheminements, ils sont à l'état Blackhole. Pour plus d'informations, consultez la section Work with NAT gateways (Travailler avec des passerelles NAT).

Chiffrement des artefacts de script Canary

CloudWatch Synthetics stocke les artefacts Canary tels que les captures d'écran, les fichiers HAR et les rapports dans votre compartiment Amazon S3. Par défaut, ces artefacts sont chiffrés au repos à l'aide d'une clé AWS gérée. Pour plus d'informations, consultez la section Clés et AWS clés du client.

Vous pouvez choisir d'utiliser une autre option de chiffrement. CloudWatch Synthetics prend en charge les éléments suivants :

- SSE-S3 : chiffrement côté serveur (SSE) avec une clé gérée par Amazon S3.
- SSE-KMS : chiffrement côté serveur (SSE) avec une clé gérée par le client AWS KMS .

Si vous souhaitez utiliser l'option de chiffrement par défaut avec une clé AWS gérée, vous n'avez pas besoin d'autorisations supplémentaires.

Pour utiliser le chiffrement SSE-S3, vous devez spécifier SSE_S3 comme mode de chiffrement lorsque vous créez ou mettez à jour votre script Canary. Vous n'avez pas besoin d'autorisations supplémentaires pour utiliser ce mode de chiffrement. Pour plus d'informations, consultez Protection des données à l'aide du chiffrement côté serveur avec les clés de chiffrement gérées par Amazon S3 (SSE-S3).

Pour utiliser une clé gérée par le AWS KMS client, vous spécifiez SSE-KMS comme mode de chiffrement lorsque vous créez ou mettez à jour votre Canary, et vous fournissez également le nom de ressource Amazon (ARN) de votre clé. Vous pouvez également utiliser une clé KMS entre comptes.

Pour utiliser une clé gérée par le client, vous devez disposer des paramètres suivants :

- Le rôle IAM de votre script Canary doit être autorisé à chiffrer vos artefacts à l'aide de votre clé. Si vous utilisez la surveillance visuelle, vous devez également lui accorder l'autorisation de déchiffrer des artefacts.

```
{
    "Version": "2012-10-17",
    "Statement": {"Effect": "Allow",
        "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt"
        ],
        "Resource": "Your KMS key ARN"
    }
}
```

- Au lieu d'ajouter des autorisations à votre rôle IAM, vous pouvez ajouter votre rôle IAM à votre politique de clé. Si vous utilisez le même rôle pour plusieurs scripts canarys, vous devez envisager cette approche.

```
{
    "Sid": "Enable IAM User Permissions",
    "Effect": "Allow",
    "Principal": {
        "AWS": "Your synthetics IAM role ARN"
    },
```

```
    "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
    ],
    "Resource": "*"
}
```

- Si vous utilisez une clé KMS entre comptes, consultez Allowing users in other accounts to use a KMS key.

Affichage des artefacts de script Canary chiffrés en utilisant une clé gérée par le client

Pour afficher les artefacts Canary, mettez à jour votre clé gérée par le client pour autoriser AWS KMS le déchiffrement à l'utilisateur qui consulte les artefacts. Vous pouvez également ajouter des autorisations de déchiffrement à l'utilisateur ou au rôle IAM qui affiche les artefacts.

La AWS KMS politique par défaut active les politiques IAM du compte pour autoriser l'accès aux clés KMS. Si vous utilisez une clé KMS entre comptes, consultez Pourquoi les utilisateurs intercomptes reçoivent-ils des erreurs d'accès refusé lorsqu'ils tentent d'accéder à des objets Amazon S3 chiffrés par une clé personnalisée AWS KMS ? .

Pour plus d'informations sur le dépannage des problèmes d'accès rejeté en raison d'une clé KMS, consultez Troubleshooting key access (Résolution des problèmes de clé d'accès).

Mise à jour de l'emplacement et du chiffrement des artefacts en utilisant la surveillance visuelle

Pour effectuer une surveillance visuelle, CloudWatch Synthetics compare vos captures d'écran avec les captures d'écran de référence acquises lors de l'exécution sélectionnée comme référence. Si vous mettez à jour l'emplacement de votre artefact ou l'option de chiffrement, vous devez effectuer l'une des actions suivantes :

- Assurez-vous que votre rôle IAM dispose d'autorisations suffisantes pour l'ancien emplacement Amazon S3 et le nouvel emplacement Amazon S3 pour les artefacts. Assurez-vous également qu'il dispose d'une autorisation pour les méthodes de chiffrement précédentes, les nouvelles méthodes de chiffrement et les clés KMS.
- Créez une nouvelle référence en sélectionnant la prochaine exécution de script Canary comme nouvelle référence. Si vous utilisez cette option, il vous suffit de vous assurer que votre rôle IAM dispose des autorisations suffisantes pour la nouvelle option d'emplacement et de chiffrement des artefacts.

Nous recommandons la deuxième option de sélection de la prochaine exécution comme nouvelle référence. Cela évite de dépendre d'un emplacement d'artefact ou d'une option de chiffrement que vous n'utilisez plus pour le script Canary.

Par exemple, supposons que votre script Canary utilise l'emplacement de l'artefact A et la clé KMS K pour charger des artefacts. Si vous mettez à jour votre script Canary vers l'emplacement de l'artefact B et la clé KMS L, vous pouvez vous assurer que votre rôle IAM dispose d'autorisations sur les deux emplacements d'artefact (A et B) et les deux clés KMS (K et L). Vous pouvez également sélectionner la prochaine exécution comme nouvelle référence et vous assurer que votre rôle IAM de script Canary dispose des autorisations sur l'emplacement d'artefact B et la clé KMS L.

Affichage des politiques et détails sur les scripts Canary

Vous pouvez afficher des détails sur vos scripts Canary et voir des statistiques sur leurs exécutions.

Pour être en mesure de voir tous les détails sur les résultats de l'exécution de votre script Canary, vous devez être connecté à un compte disposant d'autorisations suffisantes. Pour plus d'informations, consultez Rôles et autorisations requis pour les CloudWatch canaris.

Pour afficher les statistiques et les détails des scripts Canary

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Synthetics Canaries.

   Détails sur les scripts Canary que vous avez créés :

   - L'Status (État) montre visuellement combien de vos scripts Canary ont passé leurs exécutions les plus récentes.
   - Groups (Groupes) affiche les groupes que vous avez créés et indique combien d'entre eux ont des versions Canary défaillantes ou en alarme.
   - Slowest performers (Les plus lents) affiche le groupe et la région avec les versions Canary les plus lentes. Elles sont calculées en additionnant la durée moyenne de toutes les versions Canary (sur la période sélectionnée) au sein d'un groupe ou d'une région et en la divisant par le nombre de versions Canary dans le groupe ou la région. Si vous choisissez la métrique pour le groupe le plus lent, le tableau est filtré pour afficher uniquement les groupes les plus lents et leurs versions Canary. Le tableau est trié par durée moyenne.
   - Un tableau affichant tous les scripts Canary se trouve près du bas de la page. Une colonne affiche les alarmes créées pour chaque script Canary. Seules les alertes conformes à la norme d'affectation de noms pour les alertes de scripts Canary sont affichées. Cette norme est

`Synthetics-Alarm-`*`canaryName`*`-`*`index`* . Les alarmes Canary que vous créez dans la section Synthetics de CloudWatch la console utilisent automatiquement cette convention de dénomination. Si vous créez des alarmes Canary dans la section Alarmes de la CloudWatch console ou en utilisant AWS CloudFormation, sans utiliser cette convention de dénomination, les alarmes fonctionnent mais elles n'apparaissent pas dans cette liste.

3. Pour de plus amples informations sur un seul script Canary, choisissez le nom du script Canary dans la table Versions Canary.

   Détails sur ce script Canary :

   - L'onglet Availability (Disponibilité) affiche des informations sur les exécutions récentes de ce script Canary.

     Sous Canary runs (Exécutions de scripts Canary), vous pouvez choisir l'une des lignes pour voir les détails de ladite exécution.

     Sous le graphique, vous pouvez choisir Steps (Étapes), Screenshots (Capture d'écran), Logs (Journaux) ou HAR file (Fichier HAR) pour afficher ces types de détails. Si le suivi actif est activé sur le script Canary, vous pouvez également choisirTraces (Suivis) pour voir les informations de suivi des exécutions du script Canary.

     Les journaux des courses Canary sont stockés dans des compartiments S3 et dans des CloudWatch journaux.

     Les captures d'écran montrent comment vos clients voient vos pages web. Vous pouvez utiliser les fichiers HAR (fichiers d'archive HTTP) pour afficher des données détaillées sur les performances des pages web. Vous pouvez analyser la liste des demandes web et détecter des problèmes liés aux performances, notamment le temps de chargement d'un élément. Les fichiers journaux montrent l'enregistrement des interactions entre l'exécution d'un script Canary et la page web, et ils peuvent être utilisés pour identifier les détails des erreurs.

     Si le script Canary utilise l'exécution `syn-nodejs-2.0-beta` ou version ultérieure, vous pouvez trier les fichiers HAR par code de statut, taille de demande ou durée.

     L'onglet Steps (Étapes) affiche une liste des étapes du script Canary, le statut de chaque étape, le motif de l'échec, l'URL après l'exécution de l'étape, les captures d'écran et la durée de l'exécution de l'étape. Pour les scripts Canary d'API avec des étapes HTTP, vous pouvez afficher les étapes et les requêtes HTTP correspondantes si vous utilisez l'exécution `syn-nodejs-2.2` ou version ultérieure.

Cliquez sur l'onglet HTTP Requests (Requêtes HTTP) pour afficher le journal de chaque demande HTTP effectuée par le script Canary. Vous pouvez afficher les en-têtes de requête/réponse, le corps de réponse, le code de statut, les minutages d'erreur et de performance (durée totale, temps de connexion TCP, temps de liaison TLS, temps du premier octet et temps de transfert de contenu). Toutes les requêtes HTTP qui utilisent le module HTTP/HTTPS sous le capot sont capturées ici.

Par défaut, dans les scripts Canary d'API, l'en-tête de requête, l'en-tête de réponse, le corps de requête et le corps de réponse ne sont pas inclus dans le rapport pour des raisons de sécurité. Si vous choisissez de les inclure, les données sont stockées uniquement dans votre compartiment S3. Pour plus d'informations sur la manière d'inclure ces données dans le rapport, consultez executeHttpStep(StepName, RequestOptions, [rappel], [StepConfig]).

Les types de contenu du corps de réponse texte, HTML et JSON sont pris en charge. Les types de contenu tels que text/HTML, text/plain, application/JSON et application/ -1.0 sont pris en charge. x-amz-json Les réponses compressées ne sont pas prises en charge.

- L'onglet Surveillance affiche des graphiques des CloudWatch statistiques publiées par ce canari. Pour plus d'informations sur ces métriques, consultez CloudWatch statistiques publiées par canaries.

  Sous les CloudWatch graphiques publiés par le canari se trouvent des graphiques des métriques Lambda liées au code Lambda du canari.
- L'onglet Configuration affiche des informations de configuration et de planification concernant le script Canary.
- L'onglet Groups (Groupes) affiche les groupes auxquels ce script Canary est associé, le cas échéant.
- L'onglet Tags (Identifications) affiche les identifications associées au script Canary.

CloudWatch statistiques publiées par canaries

Les Canaries publient les métriques suivantes CloudWatch dans l'espace de `CloudWatchSynthetics` noms. Pour plus d'informations sur l'affichage CloudWatch des métriques, consultezAffichage des métriques disponibles.

Métrique	Description
`SuccessPercent`	Le pourcentage des exécutions de ce script Canary qui réussissent et ne trouvent aucun échec. Dimensions valides : CanaryName Statistique valide : moyenne Unités : pourcentage
`Duration`	La durée de l'exécution du script Canary, en millisecondes. Dimensions valides : CanaryName Statistique valide : moyenne Unités : millisecondes
`Errors`	Le nombre de fois où le canari n'a pas réussi à exécuter son script complet. Dimensions valides : CanaryName Statistique valide : somme
`2xx`	Le nombre de demandes réseau effectuées par le script Canary qui ont renvoyé des réponses OK, avec des codes de réponse compris entre 200 et 299. Cette métrique est signalée pour les scripts Canary d'interface utilisateur qui utilisent la version d'exécution `syn-nodejs-2.0` ou ultérieure, ainsi que pour les scripts Canary d'API qui utilisent la version d'exécution `syn-nodejs-2.2` ou ultérieure. Dimensions valides : CanaryName Statistique valide : somme Unités : nombre

Métrique	Description
`4xx`	Le nombre de demandes réseau effectuées par le script Canary qui ont renvoyé des réponses Error (Erreur), avec des codes de réponse compris entre 400 et 499. Cette métrique est signalée pour les scripts Canary d'interface utilisateur qui utilisent la version d'exécution `syn-nodejs-2.0` ou ultérieure, ainsi que pour les scripts Canary d'API qui utilisent la version d'exécution `syn-nodejs-2.2` ou ultérieure. Dimensions valides : CanaryName Statistique valide : somme Unités : nombre
`5xx`	Le nombre de demandes réseau effectuées par le script Canary qui ont renvoyé des réponses Fault (Panne), avec des codes de réponse compris entre 500 et 599. Cette métrique est signalée pour les scripts Canary d'interface utilisateur qui utilisent la version d'exécution `syn-nodejs-2.0` ou ultérieure, ainsi que pour les scripts Canary d'API qui utilisent la version d'exécution `syn-nodejs-2.2` ou ultérieure. Dimensions valides : CanaryName Statistique valide : somme Unités : nombre
`Failed`	Le nombre d'exécutions de scripts Canary qui ont échoué. Ces échecs sont liés au script Canary lui-même. Dimensions valides : CanaryName Statistique valide : somme Unités : nombre

Métrique	Description
`Failed requests`	Le nombre de demandes HTTP exécutées par le script Canary sur le site web cible qui ont échoué sans réponse. Dimensions valides : CanaryName Statistique valide : somme Unités : nombre
`VisualMonitoringSuccessPercent`	Le pourcentage de comparaisons visuelles qui correspondent aux captures d'écran de référence lors de l'exécution d'un script Canary. Dimensions valides : CanaryName Statistique valide : moyenne Unités : pourcentage
`VisualMonitoringTotalComparisons`	Le nombre total de comparaisons visuelles qui se sont produites lors de l'exécution d'un script Canary. Dimensions valides : CanaryName Unités : nombre

> **Note**
>
> Les scripts Canary qui utilisent les méthodes `executeStep()` ou `executeHttpStep()` de la bibliothèque Synthetics publient également les métriques `SuccessPercent` et `Duration` avec les dimensions `CanaryName` et `StepName` pour chaque étape.

Modification ou suppression d'un canary

Vous pouvez modifier ou supprimer un script Canary existant.

Modifier un canary

Lorsque vous modifiez un script Canary, même si vous ne modifiez pas sa planification, celle-ci est réinitialisée en fonction du moment où vous modifiez le script Canary. Par exemple, si vous avez un script Canary qui s'exécute toutes les heures et que vous le modifiez, le script Canary s'exécutera immédiatement après la fin des modifications, puis toutes les heures après cela.

Pour modifier ou mettre à jour un script Canary

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Synthetics Canaries.
3. Sélectionnez le bouton en regard du nom du script Canary, puis choisissez Actions, Edit (Modifier).
4. (Facultatif) Si ce script Canary effectue une surveillance visuelle des captures d'écran et que vous souhaitez définir la prochaine exécution du script Canary comme référence, sélectionnez Set next run as new baseline (Définir la prochaine exécution en tant que nouvelle référence).
5. (Facultatif) Si ce script Canary effectue une surveillance visuelle des captures d'écran et que vous souhaitez retirer une capture d'écran de la surveillance visuelle ou que vous souhaitez ignorer des parties de la capture d'écran lors des comparaisons visuelles, sous Visual Monitoring (Surveillance visuelle), choisissez Edit Baseline (Modifier la référence).

   La capture d'écran s'affiche et vous pouvez procéder de l'une des manières suivantes :

   - Pour empêcher l'utilisation de la capture d'écran lors de la surveillance visuelle, sélectionnez Remove screenshot from visual test baseline (Retirer la capture d'écran de la référence du test visuel).
   - Pour désigner des parties de la capture d'écran à ignorer lors des comparaisons visuelles, cliquez et faites glisser pour dessiner des zones de l'écran à ignorer. Une fois que vous avez dessiné toutes les zones que vous souhaitez ignorer pendant les comparaisons, choisissez Save (Enregistrer).
6. Si vous le souhaitez, apportez d'autres modifications au script Canary, puis choisissez Save (Enregistrer).

Supprimer un canary

Lorsque vous supprimez un canary, vous pouvez choisir de supprimer également d'autres ressources utilisées et créées par le canary. Lorsque vous supprimez un canary, vous devez également supprimer les éléments suivants :

- Fonctions Lambda et les couches utilisées par ce script Canary. Leur préfixe est `cwsyn-`*`MyCanaryName`*.
- CloudWatch alarmes créées pour ce canari. Ces alertes portent un nom qui commence par `Synthetics-Alarm-`*`MyCanaryName`*. Pour plus d'informations sur la suppression d'alertes, consultez Modifier ou supprimer une CloudWatch alarme.
- Objets et compartiments Amazon S3, tels que l'emplacement des résultats du script Canary et l'emplacement de l'artefact.
- Rôles IAM créés pour le script Canary. Ceux-ci ont le nom `role/service-role/CloudWatchSyntheticsRole-`*`MyCanaryName`*.
- Groupes de CloudWatch journaux dans les journaux créés pour le canari. Ces groupes de journaux portent les noms suivants : `/aws/lambda/cwsyn-`*`MyCanaryName-randomId`*.

Avant de supprimer un script Canary, vous pouvez souhaiter afficher les détails de ce dernier et prendre note de ces informations. Ainsi, vous pouvez supprimer les ressources appropriées après avoir supprimé le script Canary.

Pour supprimer un script Canary

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Synthetics Canaries.
3. Si le canary se trouve actuellement dans l'état `RUNNING`, vous devez l'arrêter. Seuls les canarys dans les états `STOPPED`, `READY(NOT_STARTED)` ou `ERROR` peuvent être supprimés.

   Pour arrêter un canary, sélectionnez le bouton en regard du nom du canary, puis choisissez Actions, Edit (Modifier).
4. Sélectionnez le bouton en regard du nom du script Canary, puis choisissez Actions, Delete (Supprimer).
5. Choisissez si vous souhaitez également supprimer les autres ressources créées et utilisées par le canary. Cela inclut la fonction Lambda et les couches, ainsi que le rôle IAM et la politique IAM du canary.

   Pour supprimer le rôle IAM et la politique IAM de canary, vous devez disposer d'autorisations suffisantes. Pour plus d’informations, consultez AWS politiques gérées (prédéfinies) pour CloudWatch Synthetics.
6. Saisissez **`Delete`** dans la zone et choisissez Delete (Supprimer).

7. Supprimez les autres ressources utilisées par le script Canary et crées pour celui-ci, comme indiqué précédemment dans cette section.

Démarrage, arrêt, suppression ou mise à jour de l'exécution de plusieurs canaris

Vous pouvez arrêter, démarrer, supprimer ou mettre à jour la durée d'exécution d'un maximum de cinq canaris en une seule action. Si vous mettez à jour le temps d'exécution d'un canary, il est mis à jour avec le dernier temps d'exécution disponible pour le langage et le cadre que le canary utilise.

Si vous sélectionnez plusieurs canaris et que seuls certains d'entre eux sont dans un état valable pour l'action que vous sélectionnez, l'action est exécutée uniquement sur les canaris où cette action est valide. Par exemple, si vous sélectionnez des canaris qui sont en cours d'exécution et d'autres qui ne le sont pas, et que vous choisissez de démarrer les canaris, ceux qui n'étaient pas encore en cours d'exécution démarreront, et ceux qui étaient déjà en cours d'exécution ne seront pas affectés.

Si aucun des canaris que vous sélectionnez n'est valide pour une action, cette action ne sera pas disponible dans le menu.

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Synthetics Canaries.
3. Cochez les cases en regard des canaris que vous voulez arrêter, démarrer ou supprimer.
4. Sélectionnez Actions, puis Start (Démarrer), Stop (Arrêter), Delete (Supprimer) ou Update Runtime (Mettre à jour l'exécution).

Surveiller les événements liés aux canaris avec Amazon EventBridge

Les règles relatives aux EventBridge événements Amazon peuvent vous avertir lorsque les canaris changent de statut ou terminent des courses. EventBridgefournit un near-real-time flux d'événements système décrivant les modifications apportées aux AWS ressources. CloudWatch Synthetics envoie ces événements EventBridge à dans la mesure du possible. Dans le meilleur des cas, CloudWatch Synthetics essaie d'envoyer tous les événements EventBridge à, mais dans de rares cas, il se peut qu'un événement ne soit pas organisé. EventBridge traite tous les événements reçus au moins une fois. En outre, les écouteurs d'événements peuvent ne pas recevoir les événements dans l'ordre dans lequel ces derniers se produisent.

> **Note**
>
> Amazon EventBridge est un service de bus d'événements que vous pouvez utiliser pour connecter vos applications à des données provenant de diverses sources. Pour plus d'informations, consultez Qu'est-ce qu'Amazon EventBridge ? dans le guide de EventBridge l'utilisateur Amazon.

CloudWatch Synthetics émet un événement lorsqu'un canari change d'état ou termine une course. Vous pouvez créer une EventBridge règle qui inclut un modèle d'événement correspondant à tous les types d'événements envoyés par CloudWatch Synthetics, ou qui ne correspond qu'à des types d'événements spécifiques. Lorsqu'un canari déclenche une règle, il EventBridge invoque les actions cibles définies dans la règle. Cela vous permet d'envoyer des notifications, de capturer des informations sur les événements et d'effectuer des actions correctives en réponse à un changement d'état ou à la fin de l'exécution d'un script Canary. Par exemple, vous pouvez créer des règles pour les cas d'utilisation suivants :

- Réaliser un examen lors de l'échec de l'exécution d'un script Canary
- Réaliser un examen lorsqu'un script Canary passe à l'état `ERROR`
- Suivre le cycle de vie d'un script Canary
- Contrôler la réussite ou l'échec de l'exécution d'un script Canary dans le cadre d'un flux de travail

Exemples d'événements de CloudWatch Synthetics

Cette section répertorie des exemples d'événements de CloudWatch Synthetics. Pour plus d'informations sur le format des événements, consultez la section Événements et modèles d'événements dans EventBridge.

Changement de statut d'un canary

Dans ce type d'événement, les valeurs de `current-state` et `previous-state` peuvent être les suivantes :

`CREATING` | `READY` | `STARTING` | `RUNNING` | `UPDATING` | `STOPPING` | `STOPPED` | `ERROR`

```
{
                "version": "0",
                "id": "8a99ca10-1e97-2302-2d64-316c5dedfd61",
```

```
                "detail-type": "Synthetics Canary Status Change",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:19:43Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "current-state": "STOPPED",
                                "previous-state": "UPDATING",
                                "source-location": "NULL",
                                "updated-on": 1612909161.767,
                                "changed-config": {
                                                "executionArn": {
                                                                "previous-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:function:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                },
                                                "vpcId": {
                                                                "current-value": "NULL"
                                                },
                                                "testCodeLayerVersionArn": {
                                                                "previous-
value": "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:1",
                                                                "current-value":
 "arn:aws:lambda:us-east-1:123456789012:layer:cwsyn-events-bb-1-af3e3a05-
dc5a-4f5f-96d1-989EXAMPLE:2"
                                                }
                                },
                                "message": "Canary status has changed"
                }
}
```

Réussite de l'exécution d'un canary

```
{
                "version": "0",
```

```
                "id": "989EXAMPLE-f4a5-57a7-1a8f-d9cc768a1375",
                "detail-type": "Synthetics Canary TestRun Successful",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:01Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "989EXAMPLE-dc5a-4f5f-96d1-989b75a94226",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "c6c39152-8f4a-471c-9810-989EXAMPLE",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-28ddbe266797/2021/02/09/22/23-41-200",
                                "test-run-status": "PASSED",
                                "state-reason": "null",
                                "canary-run-timeline": {
                                                "started": 1612909421,
                                                "completed": 1612909441
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Échec de l'exécution d'un canary

```
{
                "version": "0",
                "id": "2644b18f-3e67-5ebf-cdfd-bf9f91392f41",
                "detail-type": "Synthetics Canary TestRun Failure",
                "source": "aws.synthetics",
                "account": "123456789012",
                "time": "2021-02-09T22:24:27Z",
                "region": "us-east-1",
                "resources": [],
                "detail": {
                                "account-id": "123456789012",
                                "canary-id": "af3e3a05-dc5a-4f5f-96d1-9989EXAMPLE",
                                "canary-name": "events-bb-1",
                                "canary-run-id": "0df3823e-7e33-4da1-8194-
b04e4d4a2bf6",
                                "artifact-location": "cw-syn-results-123456789012-us-
east-1/canary/us-east-1/events-bb-1-ec3-989EXAMPLE/2021/02/09/22/24-21-275",
```

```
                                "test-run-status": "FAILED",
                                "state-reason": "\"Error: net::ERR_NAME_NOT_RESOLVED
 \""
                                "canary-run-timeline": {
                                        "started": 1612909461,
                                        "completed": 1612909467
                                },
                                "message": "Test run result is generated successfully"
                }
}
```

Il est possible que les événements soient dupliqués ou hors service. Pour déterminer l'ordre des événements, utilisez la propriété `time`.

Conditions préalables à la création de règles EventBridge

Avant de créer une EventBridge règle pour CloudWatch Synthetics, vous devez effectuer les opérations suivantes :

- Familiarisez-vous avec les événements, les règles et les cibles dans EventBridge.
- Créez et configurez les cibles invoquées par vos EventBridge règles. Les règles peuvent appeler de nombreux types de cibles, notamment :
  - Rubriques Amazon SNS
  - AWS Lambda fonctions
  - Flux Kinesis
  - Files d'attente Amazon SQS

Pour plus d'informations, consultez Qu'est-ce qu'Amazon EventBridge ? et Getting started with Amazon EventBridge dans le guide de EventBridge l'utilisateur Amazon.

Création d'une EventBridge règle (CLI)

Les étapes décrites dans l'exemple suivant créent une EventBridge règle qui publie une rubrique Amazon SNS lorsque le canari nommé `my-canary-name` `us-east-1` termine une exécution ou change d'état.

1. Créez la règle .

```
aws events put-rule \
```

```
  --name TestRule \
  --region us-east-1 \
  --event-pattern "{\"source\": [\"aws.synthetics\"], \"detail\": {\"canary-name\":
[\"my-canary-name\"]}}"
```

Les propriétés qui sont omises dans le modèle sont ignorées.

2. Ajoutez la rubrique en tant que cible de règle.

   - Remplacez *topic-arn* par l'Amazon Resource Name (ARN) de votre rubrique Amazon SNS.

```
aws events put-targets \
  --rule TestRule \
  --targets "Id"="1","Arn"="topic-arn"
```

> **Note**
>
> Pour autoriser Amazon EventBridge à appeler votre sujet cible, vous devez ajouter une politique basée sur les ressources à votre sujet. Pour plus d'informations, consultez les autorisations Amazon SNS dans le guide de EventBridge l'utilisateur Amazon.

Pour plus d'informations, consultez la section Événements et modèles d'événements EventBridge dans le guide de EventBridge l'utilisateur Amazon.

Réalisez des lancements et des expériences A/B avec Evidently CloudWatch

Vous pouvez utiliser Amazon CloudWatch Evidently pour valider de nouvelles fonctionnalités en toute sécurité en les proposant à un pourcentage spécifique de vos utilisateurs pendant que vous déployez la fonctionnalité. Vous pouvez surveiller les performances de la nouvelle fonction afin de décider du moment où vous souhaitez augmenter le trafic vers vos utilisateurs. Ainsi, vous pouvez réduire les risques et identifier les conséquences involontaires avant de lancer pleinement la fonction.

Vous êtes également en mesure de réaliser des expériences A/B pour prendre des décisions relatives à la conception des fonctions en vous fondant sur des preuves et des données. Une expérience peut permettre de tester jusqu'à cinq variations en même temps. Evidently recueille des

données d'expérimentation et les analyse au moyen de méthodes statistiques. Il donne également des recommandations claires concernant les variations les plus performantes. Vous pouvez tester à la fois les fonctions orientées vers l'utilisateur et les fonctions backend.

Tarification d'Evidently

Evidently débite votre compte en fonction des événements Evidently et des unités d'analyse Evidently. Les événements Evidently comprennent à la fois les événements liés aux données, tels que les clics et les consultations de pages, et les événements d'affectation qui déterminent la variation de la fonction à proposer à un utilisateur.

Les unités d'analyse Evidently sont produites à partir des événements Evidently, sur la base des règles que vous avez créées dans Evidently. Les unités d'analyse désignent le nombre de correspondances entre les règles et les événements. Par exemple, un événement de clic d'utilisateur peut produire une seule unité d'analyse Evidently, un nombre de clics. Un autre exemple est celui d'un événement de paiement par un utilisateur qui pourrait produire deux unités d'analyse Evidently, la valeur du paiement et le nombre d'articles dans le panier. Pour plus d'informations sur les tarifs, consultez Amazon CloudWatch Pricing.

CloudWatch Evidently est actuellement disponible dans les régions suivantes :

- USA Est (Ohio)
- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Stockholm)

Rubriques

- Stockage des données du projet
- Comment Evidently calcule les résultats
- Affichage des résultats du lancement dans le tableau de bord
- Affichage des résultats des expériences dans le tableau de bord
- Comment CloudWatch Evidently collecte et stocke les données
- Utilisation de rôles liés à un service pour Evidently
- CloudWatch De toute évidence, des quotas
- Didacticiel : tests A/B avec l'exemple d'application Evidently

Politiques IAM permettant d'utiliser Evidently

Pour gérer CloudWatch Evidently dans son intégralité, vous devez être connecté en tant qu'utilisateur ou en tant que rôle IAM disposant des autorisations suivantes :

- La stratégie AmazonCloudWatchEvidentlyFullAccess
- La stratégie ResourceGroupsandTagEditorReadOnlyAccess

En outre, pour créer un projet qui stocke les événements d'évaluation dans Amazon S3 ou CloudWatch Logs, vous devez disposer des autorisations suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketPolicy",
                "s3:PutBucketPolicy",
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
```

```
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:PutResourcePolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Autorisations supplémentaires pour l'intégration de CloudWatch RUM

En outre, si vous avez l'intention de gérer les lancements ou les expériences d'Evidently qui s'intègrent à Amazon CloudWatch RUM et utilisent les métriques CloudWatch RUM à des fins de surveillance, vous avez besoin de la FullAccess politique AmazonCloudWatchRUM. Pour créer un rôle IAM afin d'autoriser le client Web CloudWatch RUM à envoyer des données à CloudWatch RUM, vous devez disposer des autorisations suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:CreatePolicy",
                "iam:AttachRolePolicy"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*",
                "arn:aws:iam::*:policy/service-role/CloudWatchRUMEvidentlyPolicy-*"
            ]
        }
    ]
}
```

Autorisations pour l'accès en lecture seule à Evidently

Pour les autres utilisateurs qui ont besoin de consulter les données d'Evidently mais qui n'ont pas besoin de créer de ressources Evidently, vous pouvez accorder la AmazonCloudWatchEvidentlyReadOnlyAccesspolitique.

Créez des projets, des fonctions, des lancements et des expériences

Pour commencer à utiliser CloudWatch Evidently, que ce soit pour le lancement d'une fonctionnalité ou pour une expérience A/B, vous devez d'abord créer un projet. Un projet désigne un regroupement logique de ressources. Dans le projet, vous créez des fonctions qui possèdent des variations que vous souhaitez tester ou lancer. Vous pouvez créer une fonction avant de créer un lancement ou une expérience, ou créer les deux simultanément.

Rubriques

Création d'un nouveau projet.

Suivez ces étapes pour configurer un nouveau projet CloudWatch Evidently.

Pour créer un nouveau projet CloudWatch Evidently

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez Create a project (Créer un projet).
4. Dans le champ Nom du projet, entrez un nom à utiliser pour identifier ce projet dans la console CloudWatch Evidently.

   En option, vous pouvez ajouter une description du projet.
5. Pour le Stockage d'événements d'évaluation, sélectionnez si vous souhaitez stocker les événements d'évaluation que vous recueillez à l'aide d'Evidently. Même si vous ne stockez pas lesdits événements, Evidently les agrège pour créer des métriques et d'autres données d'expérimentation que vous pouvez consulter dans le tableau de bord Evidently. Pour plus d’informations, consultez Stockage des données du projet.
6. Pour Use client-side evaluation (Utiliser l'évaluation côté client), indiquez si vous voulez activer l'évaluation côté client pour ce projet. Grâce à l'évaluation côté client, votre application

peut attribuer des variations aux sessions utilisateur localement plutôt qu'en appelant l'EvaluateFeatureopération. Cela atténue les risques de latence et de disponibilité qui accompagnent un appel d'API. Pour plus d'informations, consultez Utilisation de l'évaluation côté client – optimisée par AWS AppConfig.

Pour créer un projet avec une évaluation côté client, vous devez disposer de l'autorisation `evidently:ExportProjectAsConfiguration`.

Si vous activez l'évaluation côté client, suivez également les étapes suivantes :

a. Choisissez d'utiliser une AWS AppConfig application existante ou d'en créer une nouvelle.

b. Choisissez d'utiliser un AWS AppConfig environnement existant ou d'en créer un nouveau.

Pour plus d'informations sur les applications et les environnements dans AWS AppConfig, consultez la section AWS AppConfig Fonctionnement.

7. (En option) Pour ajouter des balises à ce projet, sélectionnez Tags (Balises), Add new tag (Ajouter une nouvelle balise).

Ensuite, pour Key (Clé), saisissez un nom de balise Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).

Pour ajouter une autre étiquette, sélectionnez à nouveau Add new tag (Ajouter une nouvelle étiquette).

Pour plus d'informations, consultez la section AWS Ressources de balisage.

8. Sélectionnez Create a project (Créer un projet).

Utilisation de l'évaluation côté client – optimisée par AWS AppConfig

Vous pouvez utiliser l'évaluation côté client, alimentée par AWS AppConfig (évaluation côté client) dans un projet, qui permet à votre application d'attribuer des variations aux sessions utilisateur localement au lieu d'attribuer des variations en appelant l'opération. EvaluateFeature Cela atténue les risques de latence et de disponibilité qui accompagnent un appel d'API.

Pour utiliser l'évaluation côté client, associez l'extension AWS AppConfig Lambda en tant que couche à vos fonctions Lambda et configurez les variables d'environnement. L'évaluation côté client s'exécute comme un processus secondaire sur l'hôte local. Ensuite, vous pouvez appeler les PutProjectEventopérations EvaluationFeatureet contre`localhost`. Le processus d'évaluation côté

client gère l'affectation des variations, la mise en cache et la synchronisation des données. Pour plus d'informations AWS AppConfig, consultez la section AWS AppConfig Fonctionnement.

Lors de l'intégration à AWS AppConfig, vous spécifiez un ID d' AWS AppConfig application et un ID d' AWS AppConfig environnement à Evidently. Vous pouvez utiliser les mêmes ID d'application et d'environnement pour tous les projets Evidently.

Lorsque vous créez un projet avec l'évaluation côté client activée, Evidently crée un profil de AWS AppConfig configuration pour ce projet. Le profil de configuration de chaque projet sera différent.

Contrôle d'accès à l'évaluation côté client

L'évaluation côté client d'Evidently utilise un mécanisme de contrôle d'accès différent de celui du reste d'Evidently. Nous vous recommandons fortement de comprendre cela afin de pouvoir implémenter les mesures de sécurité appropriées.

Avec Evidently, vous pouvez créer des politiques IAM qui limitent les actions qu'un utilisateur peut effectuer sur des ressources individuelles. Par exemple, vous pouvez créer un rôle d'utilisateur qui interdit à un utilisateur d'effectuer cette EvaluateFeatureaction. Pour plus d'informations sur les actions Evidently qui peuvent être contrôlées à l'aide des politiques IAM, consultez Actions définies par Amazon CloudWatch Evidently.

Le modèle d'évaluation côté client permet des évaluations locales des fonctions Evidently qui utilisent des métadonnées du projet. Un utilisateur d'un projet pour lequel l'évaluation côté client est activée peut appeler l'EvaluateFeatureAPI par rapport à un point de terminaison hôte local, mais cet appel d'API n'atteint pas Eviently et n'est pas authentifié par les politiques IAM du service Eviently. Cet appel est réussi même si l'utilisateur n'a pas l'autorisation IAM pour utiliser l'EvaluateFeatureaction. Cependant, un utilisateur doit toujours être autorisé à PutProjectEventsautoriser l'agent à mettre en mémoire tampon les événements d'évaluation ou les événements personnalisés et à transférer les données vers Evidently de manière asynchrone.

En outre, un utilisateur doit disposer de l'autorisation `evidently:ExportProjectAsConfiguration` pour pouvoir créer un projet qui utilise l'évaluation côté client. Cela vous permet de contrôler l'accès aux EvaluateFeatureactions appelées lors de l'évaluation côté client.

Si vous ne faites pas attention, le modèle de sécurité de l'évaluation côté client peut contrecarrer les politiques que vous avez définies sur le reste d'Evidently. Un utilisateur `evidently:ExportProjectAsConfiguration` autorisé peut créer un projet avec l'évaluation

côté client activée, puis utiliser l'EvaluateFeatureaction pour l'évaluation côté client avec ce projet, même si l'action lui est expressément refusée dans une politique EvaluateFeatureIAM.

Mise en route avec Lambda

Evidently prend actuellement en charge l'évaluation côté client grâce à un environnement AWS Lambda . Pour commencer, déterminez d'abord l' AWS AppConfig application et l'environnement à utiliser. Choisissez une application et un environnement existants, ou créez-en de nouveaux.

Les exemples de AWS AppConfig AWS CLI commandes suivants permettent de créer une application et un environnement.

```
aws appconfig create-application --name YOUR_APP_NAME
```

```
aws appconfig create-environment --application-id YOUR_APP_ID --
name YOUR_ENVIRONMENT_NAME
```

Créez ensuite un projet Evidently en utilisant ces AWS AppConfig ressources. Pour plus d'informations, consultez Création d'un nouveau projet..

L'évaluation côté client est prise en charge dans Lambda grâce à une couche Lambda. Il s'agit d'une couche publique qui fait partie `AWS-AppConfig-Extension` d'une AWS AppConfig extension publique créée par le AWS AppConfig service. Pour plus d'informations sur les couches Lambda, consultez Couche.

Pour utiliser l'évaluation côté client, vous devez ajouter cette couche à votre fonction Lambda et configurer les autorisations et les variables d'environnement.

Pour ajouter la couche Lambda d'évaluation côté client Evidently à votre fonction Lambda et la configurer

1. Créez une fonction Lambda si ce n'est pas déjà fait.
2. Ajoutez la couche d'évaluation côté client à votre fonction. Vous pouvez soit spécifier son ARN, soit le sélectionner dans la liste des AWS couches si ce n'est pas déjà fait. Pour plus d'informations, consultez Configuration des fonctions pour utiliser des couches et Versions disponibles de l'extension AWS AppConfig Lambda.
3. Créez une politique IAM nommée EvidentlyAppConfigCachingAgentPolicyavec le contenu suivant et associez-la au rôle d'exécution de la fonction. Pour plus d'informations, consultez Rôle d'exécution Lambda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "appconfig:GetLatestConfiguration",
                "appconfig:StartConfigurationSession",
                "evidently:PutProjectEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Ajoutez la variable d'environnement requise `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` à votre fonction Lambda. Cette variable d'environnement spécifie le mappage entre le projet Evidently et les AWS AppConfig ressources.

   Si vous utilisez cette fonction pour un seul projet Evidently, définissez la valeur de la variable d'environnement sur : `applications/`*`APP_ID`*`/environments/`*`ENVIRONMENT_ID`*`/configurations/`*`PROJECT_NAME`*

   Si vous utilisez cette fonction pour plusieurs projets Evidently, utilisez une virgule pour séparer les valeurs, comme dans l'exemple suivant : `applications/`*`APP_ID_1`*`/environments/`*`ENVIRONMENT_ID_1`*`/configurations/`*`PROJECT_NAME_1`*`,applications/`*`APP_ID_2`*`/environments/`*`ENVIRONMENT_ID_2`*`/configurations/`*`PROJECT_NAME_2`*

5. (Facultatif) Définissez d'autres variables d'environnement. Pour plus d'informations, consultez Configuration de l'extension AWS AppConfig Lambda.

6. Dans votre application, obtenez localement les évaluations d'Evidently en envoyant `EvaluateFeature` à `localhost`.

   Exemple Python :

```
import boto3
from botocore.config import Config
```

```
def lambda_handler(event, context):
    local_client = boto3.client(
        'evidently',
        endpoint_url="http://localhost:2772",
        config=Config(inject_host_prefix=False)
    )
    response = local_client.evaluate_feature(
        project=event['project'],
        feature=event['feature'],
        entityId=event['entityId']
    )
    print(response)
```

Exemple de module Node.js :

```
const AWS = require('aws-sdk');
const evidently = new AWS.Evidently({
    region: "us-west-2",
    endpoint: "http://localhost:2772",
    hostPrefixEnabled: false
});

exports.handler = async (event) => {

    const evaluation = await evidently.evaluateFeature({
        project: 'John_ETCProject_Aug2022',
        feature: 'Feature_IceCreamFlavors',
        entityId: 'John'
    }).promise()

    console.log(evaluation)
    const response = {
        statusCode: 200,
        body: evaluation,
    };
    return response;
};
```

Exemple en Kotlin :

```
String localhostEndpoint = "http://localhost:2772/"
```

```
public AmazonCloudWatchEvidentlyClient getEvidentlyLocalClient() {
    return AmazonCloudWatchEvidentlyClientBuilder.standard()

 .withEndpointConfiguration(AwsClientBuilder.EndpointConfiguration(localhostEndpoint,
 region))

 .withClientConfiguration(ClientConfiguration().withDisableHostPrefixInjection(true))
        .withCredentials(credentialsProvider)
        .build();
}

AmazonCloudWatchEvidentlyClient evidently = getEvidentlyLocalClient();

// EvaluateFeature via local client.
EvaluateFeatureRequest evaluateFeatureRequest = new
 EvaluateFeatureRequest().builder()
   .withProject(${YOUR_PROJECT}) //Required.
   .withFeature(${YOUR_FEATURE}) //Required.
   .withEntityId(${YOUR_ENTITY_ID}) //Required.
   .withEvaluationContext(${YOUR_EVAL_CONTEXT}) //Optional: a JSON object of
 attributes that you can optionally pass in as part of the evaluation event sent to
 Evidently.
   .build();

EvaluateFeatureResponse evaluateFeatureResponse =
 evidently.evaluateFeature(evaluateFeatureRequest);

// PutProjectEvents via local client.
PutProjectEventsRequest putProjectEventsRequest = new
 PutProjectEventsRequest().builder()
    .withData(${YOUR_DATA})
    .withTimeStamp(${YOUR_TIMESTAMP})
    .withType(${YOUR_TYPE})
    .build();

PutProjectEvents putProjectEventsResponse =
 evidently.putProjectEvents(putProjectEventsRequest);
```

Configuration de la fréquence à laquelle le client envoie des données à Evidently

Pour spécifier la fréquence à laquelle l'évaluation côté client envoie des données à Evidently, vous pouvez configurer deux variables d'environnement.

- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_EVENT_BATCH_SIZE` spécifie le nombre d'événements par projet à regrouper avant de les envoyer à Evidently. Les valeurs valides sont des nombres entiers compris entre 1 et 50, la valeur par défaut étant 40.
- `AWS_APPCONFIG_EXTENSION_EVIDENTLY_BATCH_COLLECTION_DURATION` spécifie la durée en secondes pour attendre les événements avant de les envoyer à Evidently. La valeur par défaut est 30.

Résolution des problèmes

Utilisez les informations suivantes pour résoudre les problèmes liés à l'utilisation d' CloudWatchEvidently avec une évaluation côté client - optimisée par. AWS AppConfig

Une erreur s'est produite (BadRequestException) lors de l'appel de l' EvaluateFeature opération : la méthode HTTP n'est pas prise en charge pour le chemin fourni

Vos variables d'environnement sont peut-être mal configurées. Par exemple, vous avez peut-être utilisé `EVIDENTLY_CONFIGURATIONS` comme nom de variable d'environnement au lieu de `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS`.

ResourceNotFoundException: déploiement introuvable

Votre mise à jour des métadonnées du projet n'a pas été déployée sur AWS AppConfig. Vérifiez s'il s'agit d'un déploiement actif dans l' AWS AppConfig environnement que vous avez utilisé pour l'évaluation côté client.

ValidationException: Aucune configuration évidente pour le projet

Votre variable d'environnement `AWS_APPCONFIG_EXTENSION_EVIDENTLY_CONFIGURATIONS` est peut-être configurée avec un nom de projet incorrect.

Ajouter une fonction à un projet

Une fonctionnalité dans CloudWatch Evidently représente une fonctionnalité que vous souhaitez lancer ou dont vous souhaitez tester des variantes.

Avant de pouvoir ajouter une fonction, vous devez créer un projet. Pour plus d'informations, consultez Création d'un nouveau projet..

Pour ajouter une fonction à un projet

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet.
4. Sélectionnez Ajout de fonction.
5. Pour Feature name (Nom de fonction), saisissez un nom permettant d'identifier cette fonction dans ce projet.

   Ensuite, pour Key (Clé), saisissez un nom de balise
6. Pour Variations des fonctions, pour Type de variation, sélectionnez Booléen, Long,Double, ou Chaîne. Pour plus d'informations, consultez Types de variations.
7. Ajoutez jusqu'à cinq variations pour votre fonction. La Valeur de chaque variation doit être valide pour le Type de variation que vous avez sélectionné.

   Définissqez l'une des variations comme valeur par défaut. C'est la base de référence à laquelle les autres variations seront comparées, et celle-ci devrait être la variation qui est servie à vos utilisateurs actuellement. C'est également la variation qui est servie aux utilisateurs qui ne sont pas ajoutés à un lancement ou à une expérience pour cette fonction.
8. Sélectionnez Exemple de code. L'exemple de code indique ce que vous devez ajouter à votre application pour configurer les variations et leur attribuer des séances d'utilisateurs. Vous pouvez choisir entre JavaScript Java et Python pour le code.

   Il n'est pas nécessaire d'ajouter le code à votre application tout de suite, mais vous devez le faire avant de démarrer un lancement ou une expérience.

   Pour plus d'informations, consultez Ajout de code à votre application.
9. (En option) Pour préciser que certains utilisateurs voient toujours une certaine variation, sélectionnez Overrides (Remplacer), Add override (Ajouter une valeur de remplacement). Puis définissez un utilisateur en saisissant son ID d'utilisateur, son ID de compte ou un autre identifiant dans Identifier (Identifiant), et précisez la variation qu'il doit voir.

   Cette configuration peut être utile pour les membres de votre propre équipe de test ou d'autres utilisateurs internes lorsque vous voulez vous assurer qu'ils voient une variation précise. Les sessions des utilisateurs auxquels des remplacements sont attribués ne contribuent pas aux mesures de lancement ou d'expérimentation.

   Vous pouvez répéter cette opération pour un maximum de 20 utilisateurs en sélectionnant à nouveau Ajouter une dérogation.

10. (En option) Pour ajouter des balises à cette fonction, sélectionnez Tags (Balises), Add new Tag (Ajouter une nouvelle balise).

    Ensuite, pour Clé, saisissez un nom de balise. Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).

    Pour ajouter une autre étiquette, sélectionnez à nouveau Add new tag (Ajouter une nouvelle étiquette).

    Pour plus d'informations, consultez la section AWS Ressources de balisage.
11. Sélectionnez Add feature (Ajouter une fonction).

Types de variations

Lorsque vous créez une fonction et définissez les variations, vous devez sélectionner l'option Variation type (Type de variation). Les types possibles sont les suivants :

- Booléen
- nombre entier long
- Nombre à virgule flottante de double précision
- Chaîne

Le type de variation définit le mode de différenciation des différentes variations dans votre code. Vous pouvez utiliser le type de variation pour simplifier la mise en œuvre d' CloudWatchEvidently et également pour simplifier le processus de modification des fonctionnalités lors de vos lancements et expériences.

Par exemple, si vous définissez une fonction avec le type de variation nombre entier long, les nombres entiers que vous spécifiez pour différencier les variations peuvent être des nombres transmis directement dans votre code. Par exemple, vous pouvez tester la taille en pixels d'un bouton. Les valeurs des types de variation peuvent être le nombre de pixels utilisés dans chaque variation. Le code de chaque variation peut lire la valeur du type de variation et l'utiliser comme taille du bouton. Pour tester une nouvelle taille de bouton, il est possible de modifier le nombre utilisé pour la valeur de la variation, sans apporter d'autres modifications au code.

Lorsque vous définissez les valeurs de vos types de variations au sein d'une fonction, vous devez éviter d'attribuer les mêmes valeurs à plusieurs variantes, sauf si vous souhaitez effectuer un test A/

A pour essayer CloudWatch Evidently dans un premier temps, ou si vous avez d'autres raisons de le faire.

Evidently n'a pas de prise en charge native de JSON en tant que type, en revanche, vous pouvez passer JSON dans le type de variation String (Chaîne), et analyser ce JSON dans votre code.

Utilisez des segments pour cibler votre audience

Vous pouvez définir des segments d'audience et les utiliser dans vos lancements et dans vos expériences. Un segment est une partie de votre audience qui partage une ou plusieurs caractéristiques. Par exemple, les utilisateurs du navigateur Chrome, les utilisateurs en Europe ou les utilisateurs du navigateur Firefox en Europe qui répondent également à d'autres critères collectés par votre application, tels que l'âge.

L'utilisation d'un segment dans une expérience limite cette expérience à l'évaluation des seuls utilisateurs qui répondent aux critères du segment. Lorsque vous utilisez un ou plusieurs segments dans un lancement, vous pouvez définir différentes parts de trafic pour les différents segments d'audience.

Syntaxe de modèle de règle de segment

Pour créer un segment, définissez un modèle de règle de segment. Spécifiez les attributs que vous voulez utiliser pour évaluer si une session utilisateur sera dans le segment. Le modèle que vous créez est comparé à la valeur de `evaluationContext` qu'Evidently trouve dans une séance d'utilisateur. Pour plus d'informations, consultez En utilisant EvaluateFeature.

Pour créer un modèle de règle de segment, spécifiez les champs auxquels vous voulez que le modèle corresponde. Vous pouvez également utiliser la logique dans votre modèle, comme `And`, `Or`, `Not` et `Exists`.

Pour qu'un `evaluationContext` corresponde à un modèle, le `evaluationContext` doit correspondre à toutes les parties du modèle de règles. Evidently ignore les champs du `evaluationContext` qui ne sont pas inclus dans le modèle de règles.

Les valeurs auxquelles les modèles de règles correspondent suivent les règles JSON. Vous pouvez inclure des chaînes entre guillemets ("), des nombres et les mots-clés `true`, `false`, et `null`.

Pour les chaînes, Evidently utilise une character-by-character correspondance exacte sans rabattement ni aucune autre normalisation des chaînes. Par conséquent, les correspondances de

règles sont sensibles à la casse. Par exemple, si votre `evaluationContext` comprend un attribut `browser` mais que votre modèle de règle vérifie `Browser`, il ne correspondra pas.

Pour les nombres, Evidently utilise également une représentation par chaîne. Par exemple, 300, 300,0 et 3.0e2 ne sont pas considérés égaux.

Lorsque vous écrivez des modèles de règles pour correspondre à `evaluationContext`, vous pouvez utiliser l'API `TestSegmentPattern` ou la commande CLI `test-segment-pattern` pour vérifier que votre modèle correspond au bon JSON. Pour plus d'informations, consultez TestSegmentPattern.

Le résumé suivant présente tous les opérateurs de comparaison disponibles dans les modèles de segment Evidently.

Comparison (Comparaison)	Exemple	Syntaxe des règles
Null	UserID est null	`{ "UserID": [ null ] }`
Vide	LastName est vide	`{ "LastName": [""] }`
Égal à	Le navigateur est « Chrome »	`{ "Browser": [ "Chrome" ] }`
And	Le pays est « France » et l'appareil est « Mobile »	`{ "Country": [ "France" ], "Device": ["Mobile"] }`
Or (plusieurs valeurs d'un même attribut)	Le navigateur est « Chrome » ou « Firefox »	`{ "Browser": ["Chrome", "Firefox"]`

Comparison (Comparaison)	Exemple	Syntaxe des règles
		}

Or (attributs différents)	Le navigateur est « Safari » ou l'appareil est « Tablette »	{
		 "$or": [
		 {"Browser":
		 ["Safari"]},
		 {"Device": ["Tablet"
		}]
		]
		}

Pas	Le navigateur est tout sauf « Safari »	{
		 "Browser":
		 [{ "anything-but":
		 ["Safari"] }]
		}

Numérique (égal à)	Le prix est de 100	{
		 "Price":
		 [{ "numeric": ["=",
		 100] }]
		}

Numérique (plage)	Le prix est supérieur à 10 et inférieur ou égal à 20	{
		 "Price":
		 [{ "numeric": [">",
		 10, "<=", 20] }]
		}

Existe	Le champ d'âge existe	{
		 "Age": [{ "exists":
		 true }]
		}

Comparison (Comparaison)	Exemple	Syntaxe des règles
N’existe pas	Le champ d'âge n'existe pas	`{ "Age": [ { "exists": false } ] }`
Commence par un préfixe	La région est aux États-Unis	`{ "Region": [ {"prefix": "us-" } ] }`
Se termine par un suffixe	L'emplacement a un suffixe « West » (Ouest)	`{ "Region": [ {"suffix": "West" } ] }`

Exemples de règles de segments

Tous les exemples suivants s'appuient sur le fait que vous passez des valeurs pour `evaluationContext` avec les mêmes étiquettes et valeurs de champ que celles que vous utilisez dans vos modèles de règles.

L'exemple suivant correspond si `Browser` est Chrome ou Firefox et `Location` est US-West.

```
{
  "Browser": ["Chrome", "Firefox"],
  "Location": ["US-West"]
}
```

L'exemple suivant correspond à `Browser` est un navigateur à l'exception de Chrome, le`Location` commence par US, et un Age existe.

```
{
  "Browser": [ {"anything-but": ["Chrome"]}],
  "Location": [{"prefix": "US"}],
  "Age": [{"exists": true}]
```

```
}
```

L'exemple suivant correspond à `Location` est le Japon et soit `Browser` est Safari soit `Device` est une tablette.

```
{
 "Location": ["Japan"],
 "$or": [
   {"Browser": ["Safari"]},
   {"Device": ["Tablet"]}
 ]
}
```

Créer un segment

Après avoir créé un segment, vous pouvez l'utiliser dans n'importe quel lancement ou expérience dans n'importe quel projet.

Pour créer un segment

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Cliquez sur l'onglet Segments.
4. Choisissez Create segment (Créer un segment).
5. Pour Segment name (Nom de segment), saisissez un nom à utiliser pour identifier ce segment.

   En option, ajoutez une description.
6. Pour Segment pattern (Modèle de segment), entrez un bloc JSON qui définit le modèle de règle. Pour plus d'informations sur la syntaxe des modèles de règle, consultez Syntaxe de modèle de règle de segment.

Création d'un lancement

Pour exposer une nouvelle fonction ou une modification à un pourcentage déterminé de vos utilisateurs, créez un lancement. Ensuite, vous pouvez surveiller les métriques clés telles que les temps de chargement des pages et les conversions avant de déployer la fonction à l'intention de tous vos utilisateurs.

Pour pouvoir ajouter un lancement, vous devez avoir créé un projet. Pour plus d'informations, consultez Création d'un nouveau projet..

Lorsque vous ajoutez un lancement, il est possible d'utiliser une fonction que vous avez déjà créée ou de créer une nouvelle fonction pendant que vous créez le lancement.

Pour ajouter un lancement à un projet

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le bouton à côté du nom du projet et sélectionnez Project Actions (Actions du projet), Create launch (Créer un lancement).
4. Pour Nom du lancement, saisissez un nom à utiliser pour identifier cette fonction dans ce projet.

   Vous pouvez également ajouter une description en option.
5. Sélectionnez soit Select from existing features (Sélectionner parmi les fonctions existantes) soit Add new feature (Ajouter une nouvelle fonction).

   Si vous utilisez une fonction existante, sélectionnez-la sous Feature name (Nom de la fonction).

   Si vous choisissez Add new feature (Ajouter une nouvelle fonction), procédez comme suit :

   a. Pour Feature name (Nom de la fonction), saisissez un nom à utiliser pour identifier cette fonction dans ce projet. Vous pouvez également ajouter une description .en option.

   b. Pour Feature variations (Variations des fonctions), pourType de variation, sélectionnez Booléen, Long,Double, ou Chaîne. Pour plus d'informations, consultez Types de variations.

   c. Ajoutez jusqu'à cinq variations pour votre fonction. La Valeur de chaque variation doit être valide pour le Type de variation que vous avez sélectionné.

      Définissqez l'une des variations comme valeur par défaut. C'est la base de référence à laquelle les autres variations seront comparées, et celle-ci doit être la variation qui est servie à vos utilisateurs actuellement. Si vous arrêtez une expérience, cette variation par défaut sera ensuite diffusée à tous les utilisateurs.

   d. Sélectionnez Sample code (Exemple de code). L'exemple de code indique ce que vous devez ajouter à votre application pour configurer les variations et leur attribuer des séances d'utilisateurs. Vous pouvez choisir entre JavaScript Java et Python pour le code.

Vous n'avez pas besoin d'ajouter le code à votre application actuellement, mais vous devez le faire avant de démarrer le lancement.

Pour plus d'informations, consultez Ajout de code à votre application.

6. Pour Launch configuration (Configuration de lancement), déterminez si vous souhaitez démarrer le lancement immédiatement ou le programmer pour qu'il démarre ultérieurement.
7. (Facultatif) Pour spécifier différentes parts de trafic pour les segments d'audience que vous avez définis, au lieu de la répartition du trafic que vous utiliserez pour votre public général, choisissez Add Segment Overrides (Ajouter des remplacements de segment).

   Dans Segment Overrides (Remplacements de segment), sélectionnez un segment et définissez la répartition du trafic à utiliser pour ce segment.

   Vous pouvez éventuellement définir d'autres segments pour définir des parts de trafic en choisissant Add Segment Override (Ajouter un remplacement de segment). Un lancement peut avoir jusqu'à six remplacements de segments.

   Pour plus d’informations, consultez Utilisez des segments pour cibler votre audience.
8. Pour Traffic configuration (Configuration du trafic), sélectionnez le pourcentage de trafic à attribuer à chaque variation pour le public général qui ne correspond pas aux remplacements de segments. Vous pouvez également choisir de ne pas proposer de variations aux utilisateurs.

   Traffic summary (Le résumé du trafic) indique la part de votre trafic global disponible pour ce lancement.
9. Si vous optez pour la planification du lancement afin qu'il démarre ultérieurement, vous pouvez ajouter plusieurs étapes audit lancement. Chaque étape peut utiliser des pourcentages différents pour servir les variations. À cet effet, sélectionnez Add another step (Ajouter une autre étape), puis définissez le calendrier et les pourcentages de trafic pour l'étape suivante. Vous pouvez inclure jusqu'à cinq étapes dans un lancement.
10. Si vous souhaitez suivre les performances de vos fonctions à l'aide de métriques pendant le lancement, sélectionnez Metrics (Métriques), Add metric (Ajouter une métrique). Vous pouvez utiliser des métriques CloudWatch RUM ou des métriques personnalisées.

    Pour utiliser une métrique personnalisée, vous pouvez créer la métrique ici à l'aide d'une EventBridge règle Amazon. Pour créer une métrique personnalisée, procédez comme suit :

    - Sélectionnez Custom metrics (Métriques personnalisées) et saisissez le nom de la métrique.

- Sous Metric rule (Règle de métrique), pour l'ID de l'entité, saisissez le mode d'identification de l'entité. Il peut s'agir d'un utilisateur ou d'une séance qui effectue une action entraînant l'enregistrement d'une valeur de métrique. Par exemple : `userDetails.userID`.
- PourValue key (Valeur de la clé), saisissez la valeur à suivre pour produire la métrique.
- En option, vous pouvez saisir un nom pour les unités de la métrique. Ce nom d'unité est uniquement destiné à être affiché, pour être utilisé sur les graphiques de la console Evidently.

Lorsque vous entrez ces champs, la zone affiche des exemples de code de la EventBridge règle pour créer la métrique. Pour plus d'informations EventBridge, consultez Qu'est-ce qu'Amazon EventBridge ?

Pour utiliser les métriques RUM, un moniteur d'application RUM doit déjà être configuré pour votre application. Pour plus d'informations, consultez Configuration d'une application pour utiliser CloudWatch RUM.

> **Note**
>
> Si vous utilisez des métriques RUM et que le moniteur d'application n'est pas configuré pour échantillonner 100 % des séances d'utilisateurs, toutes les séances d'utilisateurs participant au lancement n'enverront pas de métriques à Evidently. Pour garantir l'exactitude des métriques de lancement, nous recommandons que le moniteur de l'application utilise 100 % des séances d'utilisateurs à des fins d'échantillonnage.

11. (Facultatif) Si vous créez au moins une métrique pour le lancement, vous pouvez associer une CloudWatch alarme existante à ce lancement. Pour ce faire, sélectionnez Associer des CloudWatch alarmes.

    Lorsque vous associez une alarme à un lancement, CloudWatch Evidently doit ajouter des balises à l'alarme avec le nom du projet et le nom du lancement. Cela permet à CloudWatch Evidently d'afficher les alarmes correctes dans les informations de lancement de la console.

    Pour confirmer qu' CloudWatch Evidently ajoutera ces balises, choisissez Autoriser Evidently pour étiqueter la ressource d'alarme identifiée ci-dessous avec cette ressource de lancement. Puis sélectionnez Associate alarm (alerte associée) et saisissez le nom de l'alerte.

    Pour plus d'informations sur la création d' CloudWatch alarmes, consultez Utilisation des CloudWatch alarmes Amazon.

12. (En option) Pour ajouter des balises à ce lancement, sélectionnez Tags (Balises), Add new tag (Ajouter une nouvelle balise).

    Puis, dans Key (Clé), saisissez un nom de balise. Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).

    Pour ajouter une autre étiquette, sélectionnez à nouveau Add new tag (Ajouter une nouvelle étiquette).

    Pour plus d'informations, consultez la section AWS Ressources de balisage.
13. Sélectionnez Create launch (Créer un lancement).

Création d'une expérience

Utilisez des expériences pour tester différentes versions d'une fonction ou d'un site web et recueillir des données depuis des séances d'utilisateurs réelles. Ainsi, vous pouvez faire des choix pour votre application en fonction de preuves et de données.

Avant de pouvoir ajouter une expérience, vous devez avoir créé un projet. Pour plus d'informations, consultez Création d'un nouveau projet..

Lorsque vous ajoutez une expérience, vous pouvez utiliser une fonction que vous avez déjà créée ou créer une nouvelle fonction pendant la création de l'expérience.

Pour ajouter une expérience à un projet

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le bouton à côté du nom du projet et choisissez Project actions (Actions du projet),Create experiment (Créer une expérience).
4. Pour Experiment name (Nom de l'expérience), saisissez un nom à utiliser pour identifier cette fonction dans ce projet.

   En option, vous pouvez ajouter une description.
5. Sélectionnez soit Select from existing features (Sélectionner parmi les fonctions existantes) soit Add new feature (Ajouter une nouvelle fonction).

   Si vous utilisez une fonction existante, sélectionnez-la sous Feature name (Nom de la fonction).

Si vous choisissez Add new feature (Ajouter une nouvelle fonction), procédez comme suit :

a. Pour Feature name (Nom de fonction), saisissez un nom à utiliser pour identifier cette fonction dans ce projet. En option, vous pouvez également saisir une description.

b. Pour Feature variations (Variations des fonctions), pour Type de variation, sélectionnez Booléen,Long, Double, ou Chaîne. Le type définit le type de valeur utilisé pour chaque variation. Pour plus d'informations, consultez Types de variations.

c. Ajoutez jusqu'à cinq variations pour votre fonction. La Valeur de chaque variation doit être valide pour le Type de variation que vous avez sélectionné.

   Définissqez l'une des variations comme valeur par défaut. Il s'agit de la base de référence à laquelle les autres variations seront comparées, et celle-ci devrait être la variation qui est servie à vos utilisateurs actuellement. Si vous arrêtez une expérience qui utilise cette fonction, la variation par défaut est ensuite servie au pourcentage d'utilisateurs qui faisaient partie de l'expérience précédemment.

d. Sélectionnez Sample code (Exemple de code). L'exemple de code indique ce que vous devez ajouter à votre application pour configurer les variations et leur attribuer des séances d'utilisateurs. Vous pouvez choisir entre JavaScript Java et Python pour le code.

   Vous n'avez pas besoin d'ajouter le code à votre application actuellement, mais vous devez le faire avant de démarrer l'expérience. Pour plus d’informations, consultez Ajout de code à votre application.

6. Pour Audience (Public), sélectionnez éventuellement un segment que vous avez créé si vous souhaitez que cette expérience s'applique uniquement aux utilisateurs qui correspondent à ce segment. Pour plus d'informations sur les segments, consultez Utilisez des segments pour cibler votre audience.

7. Pour Traffic split for the experiment (Répartition du trafic pour l'expérience), indiquez le pourcentage du public sélectionné dont les séances seront utilisées dans l'expérience. Puis attribuez le trafic aux différentes variations utilisées par l'expérience.

   Si un lancement et une expérience sont tous deux exécutés en même temps pour la même fonction, le public est d'abord dirigé vers le lancement. Ensuite, le pourcentage de trafic défini pour le lancement provient du public global. Après cela, le pourcentage que vous définissez ici correspond au pourcentage du public restant utilisé pour l'expérice. Ensuite, tout trafic restant reçoit la variation par défaut.

8. Pour Metrics (Métriques), sélectionnez les métriques à utiliser pour évaluer les variations au cours de l'expérience. Vous devez utiliser au moins une métrique pour l'évaluation.

   a. Pour Metric Source, choisissez d'utiliser des métriques CloudWatch RUM ou des métriques personnalisées.

   b. Saisissez le nom de la métrique. Pour Goal (Objectif), sélectionnez Increase (Augmenter) si vous souhaitez qu'une valeur plus élevée pour la métrique indique une meilleure variation. Sélectionnez Decrease (Diminuer) si vous souhaitez qu'une valeur inférieure pour la métrique indique une meilleure variation.

   c. Si vous utilisez une métrique personnalisée, vous pouvez la créer ici à l'aide d'une EventBridge règle Amazon. Pour créer une métrique personnalisée, procédez comme suit :

      - Sous Metric rule (Règle de métriques), pour Entity (ID ID de l'entité), saisissez un moyen d'identifier l'entité. Il peut s'agir d'un utilisateur ou d'une séance qui effectue une action qui entraîne l'enregistrement d'une valeur de métrique. Par exemple : `userDetails.userID`.
      - PourValue key (Valeur de la clé), saisissez la valeur à suivre pour produire la métrique.
      - En option, vous pouvez saisir un nom pour les unités de la métrique. Ce nom d'unité est uniquement destiné à être affiché, pour être utilisé sur les graphiques de la console Evidently.

      Vous ne pouvez utiliser les métriques RUM que si vous avez configuré RUM pour surveiller cette application. Pour plus d'informations, consultez Utiliser du CloudWatch rhum.

      > **Note**
      >
      > Si vous utilisez des métriques RUM et que le moniteur d'applications n'est pas configuré pour échantillonner 100 % des séances d'utilisateurs, toutes les séances d'utilisateurs de l'expérience n'enverront pas de métriques à Evidently. Pour garantir l'exactitude des métriques de l'expérience, nous recommandons que le moniteur d'applications utilise 100 % des séances d'utilisateurs pour l'échantillonnage.

   d. (En option) Pour ajouter d'autres métriques à évaluer, sélectionnez Add metric (Ajouter une métrique). Vous pouvez évaluer jusqu'à trois métriques au cours de l'expérience.

9. (Facultatif) Pour créer des CloudWatch alarmes à utiliser dans le cadre de cette expérience, sélectionnez des CloudWatchalarmes. Les alertes peuvent contrôler si la différence de résultats

entre chaque variation et la variation par défaut est supérieure à un seuil que vous définissez. Si les performances d'une variation sont inférieures à celles de la variation par défaut, et que la différence est supérieure à votre seuil, le système passe en état d'alerte et vous en informe.

La création d'une alerte à ce niveau génère une alerte pour chaque variation qui n'est pas la variation par défaut.

Si vous en créez un alerte, indiquez ce qui suit :

- Pour Nom de métrique, sélectionnez la métrique d'expérience à utiliser pour l'alerte.
- Pour Condition d'alerte sélectionnez la condition qui induit la condition d'alerte lorsque les valeurs de la métrique de variation sont comparées aux valeurs de métrique de variation par défaut. Par exemple, sélectionnez Greater (Plus grand) ouGreater/Equal (Plus grand/Égal si des nombres plus élevés indiquent que la performance est médiocre. Cela serait approprié si la métrique mesure le temps de chargement des pages, par exemple.
- Saisissez un nombre pour le seuil, qui correspond au pourcentage de différence de performance qui entraînera le passage de l'alerte à `ALARM`l'état.
- PourMoyenne sur la période, sélectionnez la quantité de données de métriques pour chaque variation qui est agrégée ensemble avant la comparaison.

Vous pouvez sélectionner à nouveau Add new alarm (Ajouter une nouvelle alerte) pour ajouter d'autres alertes à l'expérience.

Ensuite, sélectionnez Set notifications for the alarm (Définir les notifications de l'alerte) et sélectionnez ou créez une rubrique Amazon Simple Notification Service pour envoyer des notifications d'alerte à. Pour plus d'informations, consultez Configuration des notifications Amazon SNS,

10. (En option) Pour ajouter des balises à cette expérience, sélectionnez Tags (Balises), Add new Tag (Ajouter une nouvelle balise).

    Ensuite, pour Clé de balise, saisissez un nom de balise. Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).

    Pour ajouter une autre étiquette, sélectionnez à nouveau Add new tag (Ajouter une nouvelle étiquette).

    Pour plus d'informations, consultez la section AWS Ressources de balisage.

11. Sélectionnez Create Experiment (Créer une expérience).
12. Si vous ne l'avez pas encore fait, créez les variantes de fonctions dans votre application.
13. Sélectionnez Exécuté. L'expérience ne démarre pas tant que vous ne l'avez pas démarrée.

Après avoir réalisé les étapes de la procédure suivante, l'expérience démarre immédiatement.

Pour démarrer une expérience que vous avez créée

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet
4. Cliquez sur l'onglet Experiments (Expériences).
5. Sélectionnez le bouton à côté du nom de l'expérience et sélectionnez Actions (Actions), Start experiment (Démarrer une expérience).
6. (En option) Pour afficher ou modifier les paramètres d'expérience que vous avez définis lorsque vous avez créé cette expérience, sélectionnez Experiment setup (Configuration de l'expérience).
7. Sélectionnez une heure pour la fin de l'expérience.
8. Sélectionnez Start experiment (Démarrer une expérience).

   L'expérience démarre immédiatement.

Gérez les fonctions, les lancements et les expériences

Suivez les procédures de ces sections pour gérer les fonctions, les lancements et les expériences que vous avez créés.

Rubriques

- Consultez les règles d'évaluation actuelles et le trafic de public ciblé pour une fonction
- Modification trafic de lancement
- Modifier les étapes ultérieures d'un lancement
- Modification du trafic d'expérience
- Arrêt d'un lancement
- Arrêt d'une expérience

Consultez les règles d'évaluation actuelles et le trafic de public ciblé pour une fonction

Vous pouvez utiliser la console CloudWatch Evidently pour voir comment les règles d'évaluation de la fonctionnalité répartissent le trafic d'audience entre les lancements, les tests et les variantes actuels de la fonctionnalité.

Pour afficher le trafic de public ciblé d'une entité

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet qui contient la fonction.
4. Cliquez sur l'onglet Features (Fonctions).
5. Sélectionnez le nom de la fonction.

   Dans l'onglet Evaluation rules (Règles d'évaluation), vous pouvez voir le flux de trafic de public ciblé pour votre fonction, comme suit :

   - D'abord, les valeurs de remplacement sont évaluées. Celles-ci précisent que certains utilisateurs bénéficient toujours d'une variation spécifique. Les sessions des utilisateurs auxquels des remplacements sont attribués ne contribuent pas aux mesures de lancement ou d'expérimentation.
   - Ensuite, le trafic restant est disponible pour le lancement en cours, s'il y en a un. Si un lancement est en cours, le tableau de la section Lancements affiche le nom du lancement et le trafic de lancement reparti entre les variations de fonctions. Sur le côté droit de la section Lancement, un indicateur de trafic affiche la quantité de public ciblé disponible (après les remplacements) allouée à ce lancement. Le reste du trafic non alloué au lancement passe à l'expérience (le cas échéant), puis à la variation par défaut.
   - Ensuite, le trafic restant est disponible pour l'expérience en cours, s'il y en a une. Si une expérience est en cours, le tableau de la section Expériences affiche le nom et la progression de l'expérience. Sur le côté droit de la section Expériences, un indicaqteur de trafic affiche la quantité de public ciblé disponible (après les remplacements et les lancements) allouée à cette expérience. Le reste du trafic non alloué au lancement ou à l'expérience reçoit la variation par défaut de la fonction.

Modification trafic de lancement

Vous pouvez modifier l'allocation du trafic pour un lancement à tout moment, y compris pendant que le lancement est en cours.

Si un lancement et une expérimentation sont en cours pour la même fonction, toute modification du trafic de la fonction entraînera une modification du trafic de l'expérimentation. En effet, le public ciblé disponible pour l'expérience est la partie de l'ensemble de votre public ciblé qui n'est pas déjà allouée au lancement. L'augmentation du trafic de lancement réduit le public ciblé disponible pour l'expérience, et la diminution du trafic de lancement ou la fin du lancement augmente le public visé disponible pour l'expérience.

Pour modifier l'allocation du trafic pour un lancement

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet qui contient le lancement.
4. Cliquez sur l'onglet Launches (Lancements).
5. Sélectionnez le nom du lancement.

   Sélectionnez Modifier le trafic de lancement.
6. Pour Serve (Servir), sélectionnez le nouveau pourcentage de trafic à attribuer à chaque variation. Vous pouvez également décider de ne pas proposer de variantes aux utilisateurs. Lorsque vous modifiez ces valeurs, vous pouvez voir les effets mis à jour sur le trafic global de vos fonctions sous Traffic summary (Résumé du trafic).

   Traffic summary (Le résumé du trafic) indique la part de votre trafic global disponible pour ce lancement et la part de ce trafic disponible allouée à ce lancement.
7. Sélectionnez Modifier.

Modifier les étapes ultérieures d'un lancement

Vous pouvez modifier la configuration des étapes de lancement qui ne se sont pas encore produites et ajouter d'autres étapes à un lancement.

Pour modifier les étapes d'un lancement

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

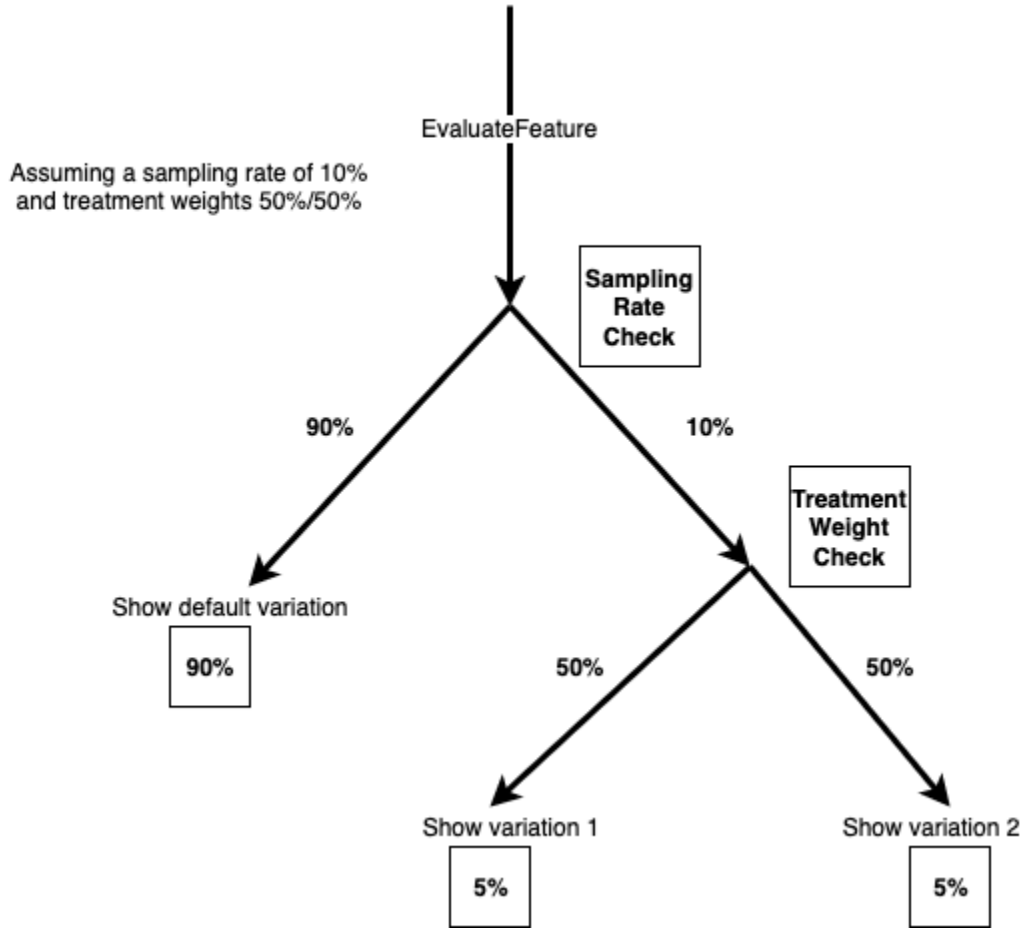
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet qui contient le lancement.
4. Cliquez sur l'onglet Launches (Lancements).
5. Sélectionnez le nom du lancement.

   Sélectionnez Modify launch traffic (Modifier le trafic de lancement).
6. Sélectionnez Schedule launch (Planification du lancement).
7. Pour toutes les étapes qui n'ont pas encore démarré, vous pouvez modifier le pourcentage de public ciblé disponible à utiliser lors de l'expérience. Vous pouvez également modifier le mode d'allocation de leur trafic parmi les variations.

   Vous pouvez ajouter d'autres étapes au lancement en sélectionnant Add another step (Ajouter une autre étape). Un lancement peut comporter cinq étapes maximum.
8. Sélectionnez Modifier.

Modification du trafic d'expérience

Vous pouvez modifier le taux d'échantillonnage d'une expérience à tout moment, y compris pendant que l'expérience est en cours. Toutefois, vous ne pouvez pas mettre à jour les poids du traitement après l'exécution d'une expérience. Par conséquent, vous pouvez modifier le trafic total exposé à l'expérience après son exécution, mais pas l'allocation relative à chaque traitement. Si vous modifiez le trafic d'une expérience en cours, nous vous conseillons de n'augmenter que l'allocation de trafic, afin de ne pas introduire de biais.



Pour modifier l'allocation du trafic d'une expérience

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Application monitoring (Surveillance des applications), Evidently.
3. Sélectionnez le nom du projet qui contient le lancement.
4. Cliquez sur l'onglet Experiments (Expériences).
5. Sélectionnez le nom du lancement.
6. Sélectionnez Modify experiment traffic (Modifier le trafic d'expérience).
7. Saisissez un pourcentage ou utilisez le curseur pour indiquer la quantité de trafic disponible à allouer à cette expérience. Le trafic disponible est l'ensemble du public visé moins le trafic alloué à un lancement en cours, s'il y en a un. Le trafic qui n'est pas alloué au lancement ou à l'expérience reçoit la variation par défaut.
8. Sélectionnez Modifier.

Arrêt d'un lancement

Si vous arrêtez un lancement en cours, vous ne pourrez pas le reprendre ou le redémarrer. En outre, il ne sera pas évalué en règle générale pour l'allocation du trafic, et le trafic alloué au lancement sera à la place disponible pour l'expérience de la fonction, s'il y en a une. Dans le cas contraire, tout le trafic recevra la variation par défaut une fois le lancement arrêté.

Pour arrêter définitivement un lancement

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet qui contient le lancement.
4. Cliquez sur l'onglet Launch (Lancement).
5. Sélectionnez le bouton se trouvant à gauche du nom du lancement.
6. Sélectionnez Actions (Actions), Cancel Launcwh (Annuler le lancement) ou Actions Actions), Application monitoring (Marquer comme terminé).

Arrêt d'une expérience

Si vous arrêtez une expérience en cours, vous ne pourrez pas la reprendre ou la redémarrer. La partie du trafic précédemment utilisée dans l'expérience sera servie avec la variation par défaut.

Lorsqu'une expérience n'est pas arrêtée manuellement et excède sa date de fin, le trafic ne change pas. La partie du trafic allouée à l'expérience est toujours destinée à l'expérience. Pour arrêter cela, et faire en sorte que le trafic de l'expérience soit servi avec la variation par défaut, marquez l'expérience comme terminée.

Lorsque vous arrêtez une expérience, vous pouvez choisir de l'annuler ou de la marquer comme terminée. Si vous l'annulez, elle s'affichera comme Cancelled (Annulée) dans la liste des expériences. Si vous préférez de la marquer comme terminée, elle s'affiche comme Terminée.

Pour arrêter définitivement une expérience

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet qui contient l'expérience.
4. Cliquez sur l'onglet Experiments (Expériences).

5. Sélectionnez le bouton se trouvant à gauche du nom de l'expérience.
6. Sélectionnez Actions (Actions), Cancel experiment (Annuler une expérience) ou Actions (Actions), Mark as complete (Marquer comme terminé).

Ajout de code à votre application

Pour utiliser CloudWatch Evidently, vous devez ajouter du code à votre application pour attribuer une variation à chaque session utilisateur et pour envoyer des métriques à Evidently. Vous utilisez l'`EvaluateFeature`opération CloudWatch Evidently pour attribuer des variations aux sessions utilisateur, et vous utilisez l'`PutProjectEvents`opération pour envoyer des événements à Evidently afin qu'ils soient utilisés pour calculer des métriques pour vos lancements ou vos expériences.

Lorsque vous créez des variations ou des métriques personnalisées, la console CloudWatch Evidently fournit des exemples du code que vous devez ajouter.

Pour un end-to-end exemple, voirDidacticiel : tests A/B avec l'exemple d'application Evidently.

En utilisant EvaluateFeature

Lorsque des variantes de fonctionnalités sont utilisées lors d'un lancement ou d'une expérience, l'application utilise l' EvaluateFeatureopération pour attribuer une variation à chaque session utilisateur. L'attribution d'une variation à un utilisateur est un événement d'évaluation. Lorsque vous appelez cette opération, vous transmettez ce qui suit :

- Nom de fonction– Obligatoire. Evidently traite l'évaluation selon les règles d'évaluation des fonctions du lancement ou de l'expérience, et sélectionne une variation pour l'entité.
- entityId– Obligatoire. Représente un utilisateur unique.
- evaluationContex– En option. Objet JSON représentant des informations supplémentaires sur un utilisateur. Evidently utilisera cette valeur pour faire correspondre l'utilisateur à un segment de votre audience lors des évaluations de fonctionnalités, si vous avez créé des segments. Pour plus d'informations, consultez Utilisez des segments pour cibler votre audience.

  Voici un exemple de valeur `evaluationContext` que vous pouvez envoyer à Evidently.

```
{
    "Browser": "Chrome",
    "Location": {
        "Country": "United States",
```

```
        "Zipcode": 98007
    }
}
```

Évaluations persistantes

CloudWatch Utilise évidemment des évaluations « persistantes ». Une configuration unique d'`entityId`, de la fonctionnalité, de la configuration de fonctionnalité et d'`evaluationContext` reçoit toujours la même attribution de variation. Le seul moment où l'affectation de variation change est lorsqu'une entité est ajoutée à une substitution ou que le trafic d'expérience est composé.

Une configuration de fonctionnalité comprend les éléments suivants :

- Les variations de fonctionnalités
- La configuration de variation (pourcentages attribués à chaque variation) pour une expérience en cours pour cette fonctionnalité, le cas échéant.
- La configuration de variation pour un lancement en cours d'exécution de cette fonctionnalité, le cas échéant. La configuration de variation inclut les remplacements de segment définis, le cas échéant.

Si l'allocation de trafic d'une expérience est augmentée, tous les `entityId` qui étaient précédemment affectés à un groupe de traitement d'expérience continueront à bénéficier du même traitement. Tout `entityId` ayant été précédemment attribué au groupe de contrôle peut être affecté à un groupe de traitement d'expérience, selon la configuration de variation spécifiée pour l'expérience.

Si l'allocation de trafic d'une expérience est réduite, un `entityId` peut passer d'un groupe de traitement à un groupe de contrôle, mais il ne sera pas transféré vers un autre groupe de traitement.

En utilisant PutProjectEvents

Pour coder une métrique personnalisée pour Evidently, vous utilisez l' PutProjectEventsopération. Voici un exemple de charge utile simple.

```
{
    "events": [
        {
            "timestamp": {{$timestamp}},
            "type": "aws.evidently.custom",
```

```
            "data": "{\"details\": {\"pageLoadTime\": 800.0}, \"userDetails\":
 {\"userId\": \"test-user\"}}"
        }
    ]
}
```

Le `entityIdKey` peut simplement être un `entityId` ou vous pouvez le renommer par autre chose quelconque, par exemple `userId`. Dans l'éventualité même, `entityId` peut être un nom d'utilisateur, un ID de séance, etc.

```
"metricDefinition":{
    "name": "noFilter",
    "entityIdKey": "userDetails.userId", //should be consistent with jsonValue in
 events "data" fields
    "valueKey": "details.pageLoadTime"
},
```

Pour vous assurer que les événements sont associés au lancement ou à l'expérience correcte, vous devez passer la même chose.`entityId`lorsque vous appelez les deux`EvaluateFeature`et`PutProjectEvents`. N'oubliez pas d'appeler `PutProjectEvents` après l'`EvaluateFeature`appel, sinon les données seront supprimées et ne seront pas utilisées par CloudWatch Evidently.

Le`PutProjectEvents`ne nécessite pas le nom de l'entité en tant que paramètre d'entrée. De cette façon, vous pouvez utiliser un seul événement dans différentes expériences. Par exemple, supposons que vous appelez`EvaluateFeature`avec le`entityId`Réglé sur`userDetails.userId`. Si deux expériences ou plus sont en cours d'exécution, un seul événement de la session de cet utilisateur peut émettre des mesures pour chacune de ces expériences. Pour ce faire, vous appelez`PutProjectEvents`une fois pour chaque expérience, en utilisant cette même`entityId`.

Timing

Après les appels de votre application`EvaluateFeature`, il y a une période d'une heure où les événements métriques de`PutProjectEvents`sont attribués sur la base de cette évaluation. Si d'autres événements surviennent après la période d'une heure, ils ne sont pas attribués.

Toutefois, si la même chose`entityId`est utilisé pour un nouveau`EvaluateFeature`appel pendant la fenêtre d'une heure de cet appel initial, le plus tard`EvaluateFeature`est maintenant utilisé à

la place, et le minuteur d'une heure est redémarré. Cela ne peut se produire que dans certaines circonstances, par exemple lorsque le trafic d'expérience est composé entre les deux affectations, comme expliqué dans la précédenteÉvaluations persistantesSection.

Pour un end-to-end exemple, voirDidacticiel : tests A/B avec l'exemple d'application Evidently.

Stockage des données du projet

Evidently recueille deux types d'événements :

- Les événements d'évaluation sont liés à la variation de fonction attribuée à une séance utilisateur. Evidently utilise ces événements pour produire des métriques et d'autres données d'expérimentation et de lancement, que vous pouvez consulter dans la console Evidently.

  Vous pouvez également choisir de stocker ces événements d'évaluation dans Amazon CloudWatch Logs ou Amazon S3.
- Les événements personnalisés permettent de produire des métriques à partir d'actions d'utilisateur telles que les clics et les paiements. Evidently ne vous propose pas de méthode pour stocker des événements personnalisés. Si vous souhaitez les enregistrer, vous devez modifier le code de votre application pour les envoyer vers une option de stockage en dehors d'Evidently.

Format des journaux d'événements d'évaluation

Si vous choisissez de stocker les événements d'évaluation dans CloudWatch Logs ou Amazon S3, chaque événement d'évaluation est stocké sous forme d'événement de journal au format suivant :

```
{
    "event_timestamp": 1642624900215,
    "event_type": "evaluation",
    "version": "1.0.0",
    "project_arn": "arn:aws:evidently:us-east-1:123456789012:project/petfood",
    "feature": "petfood-upsell-text",
    "variation": "Variation1",
    "entity_id": "7",
    "entity_attributes": {},
    "evaluation_type": "EXPERIMENT_RULE_MATCH",
    "treatment": "Variation1",
    "experiment": "petfood-experiment-2"
}
```

Voici plus de détails sur le format d'événement d'évaluation précédent :

- L'horodatage est en heure UNIX avec des millisecondes
- La variation est le nom de la variation de la fonction affectée à cette session utilisateur.
- L'ID de l'entité est une chaîne.
- Les attributs d'entité sont un hachage de valeurs arbitraires envoyées par le client. Par exemple, si le `entityId` est mappé en bleu ou en vert, vous pouvez éventuellement envoyer des ID utilisateur, des données de session ou tout autre élément souhaité du point de vue de la corrélation et de l'entrepôt de données.

Politique et chiffrement IAM pour le stockage d'événements d'évaluation dans Amazon S3

Si vous souhaitez utiliser Amazon S3, vous devez ajouter une politique IAM comme la suivante pour autoriser Evidently à publier les journaux dans le compartiment Amazon S3. Cela est dû au fait que les compartiments Amazon S3 et les objets qu'ils contiennent sont privés et qu'ils n'autorisent pas l'accès à d'autres services par défaut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSLogDeliveryWrite",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::bucket_name/optional_folder/AWSLogs/account_id/
*",
            "Condition": {"StringEquals": {"s3:x-amz-acl": "bucket-owner-full-
control"}}
        },
        {
            "Sid": "AWSLogDeliveryCheck",
            "Effect": "Allow",
            "Principal": {"Service": "delivery.logs.amazonaws.com"},
            "Action": ["s3:GetBucketAcl", "s3:ListBucket"],
            "Resource": "arn:aws:s3:::bucket_name"
        }
    ]
```

```
}
```

Si vous stockez des données Evidently dans Amazon S3, vous pouvez également choisir de les chiffrer à l'aide du chiffrement côté serveur en utilisant des AWS Key Management Service clés (SSE-KMS). Pour plus d'informations, consultez Protection des données à l'aide du chiffrement côté serveur.

Si vous utilisez une clé gérée par le client depuis AWS KMS, vous devez ajouter ce qui suit à la politique IAM relative à votre clé. Cette action permet à Evidently d'écrire dans le compartiment.

```
{
    "Sid": "AllowEvidentlyToUseCustomerManagedKey",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "delivery.logs.amazonaws.com"
        ]
    },
   "Action": [
       "kms:Encrypt",
       "kms:Decrypt",
       "kms:ReEncrypt*",
       "kms:GenerateDataKey*",
       "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

Comment Evidently calcule les résultats

Vous pouvez utiliser les tests A/B d'Amazon CloudWatch Evidently comme outil de prise de décision basée sur les données. Dans un test A/B, les utilisateurs sont affectés de manière aléatoire soit au groupe de contrôle (également appelé la variation par défaut), soit à l'un des groupes de traitement (également appelés les variations testées). Par exemple, les utilisateurs du groupe de contrôle peuvent utiliser le site Web, le service ou l'application de la même manière qu'avant le début de l'expérience. Pendant ce temps, les utilisateurs du groupe de traitement peuvent remarquer le changement.

CloudWatch Supporte évidemment jusqu'à cinq variantes différentes dans une expérience. Evidently affecte aléatoirement le trafic à ces variations. De cette façon, vous pouvez suivre les métriques

commerciales (telles que les revenus) et les métriques de performance (telles que la latence) pour chaque groupe. Evidently effectue les opérations suivantes :

- Il compare le traitement avec le contrôle. (Par exemple, il compare si les revenus augmentent ou diminuent avec un nouveau processus de paiement.)
- Il indique si la différence observée entre le traitement et le contrôle est significative. Pour cela, Evidently propose deux approches : les niveaux de signification fréquentistes et les probabilités bayésiennes.

Pourquoi utiliser les approches fréquentiste et bayésienne ?

Considérons un cas où le traitement n'a aucun effet par rapport au contrôle, ou un cas où le traitement est identique au contrôle (un test A/A). Vous observeriez toujours une petite différence entre le traitement et le contrôle dans les données. Cela est dû au fait que les participants au test sont constitués d'un échantillon fini d'utilisateurs, représentant un petit pourcentage de tous les utilisateurs du site Web, du service ou de l'application. Les niveaux de signification fréquentistes et les probabilités bayésiennes permettent de savoir si la différence observée est significative ou due au hasard.

Evidently prend en compte les éléments suivants pour déterminer si la différence observée est significative :

- L'ampleur de la différence
- Le nombre d'échantillons dans le test
- La manière dont les données sont distribuées

Analyse fréquentiste dans Evidently

Evidently utilise des tests séquentiels, ce qui permet d'éviter les problèmes habituels de peeking (coup d'oeil), un écueil courant des statistiques fréquentistes. Le peeking est la pratique qui consiste à vérifier les résultats d'un test A/B en cours afin de l'arrêter et de prendre une décision en fonction des résultats observés. Pour plus d'informations sur les tests séquentiels, consultez Time-uniform, nonparametric, nonasymptotic confidence sequences par Howard et autres. (Ann. Statist. 49 (2) 1055 - 1080, 2021).

Comme les résultats d'Evidently sont valides à tout moment (résultats anytime-valid), vous pouvez jeter un coup d'œil aux résultats pendant l'expérience et continuer à tirer des conclusions solides.

Cela peut réduire certains coûts de l'expérimentation, car vous pouvez arrêter une expérience avant l'heure prévue si les résultats sont déjà significatifs.

Evidently génère des niveaux de signification valides à tout moment et des intervalles de confiance à 95 % valides à tout moment de la différence entre la variation testée et la variation par défaut de la métrique cible. La colonne Result (Résultat) des résultats de l'expérience indique la performance de la variation testée, qui peut être l'une des suivantes :

- Inconclusive (Non concluant) : le niveau de signification est inférieur à 95 %
- Better (Mieux) : le niveau de signification est de 95 % ou plus et l'un des éléments suivants est vrai :
  - La limite inférieure de l'intervalle de confiance à 95 % est supérieure à zéro et la métrique doit augmenter
  - La limite supérieure de l'intervalle de confiance à 95 % est inférieure à zéro et la métrique doit diminuer
- Worse (Pire) : le niveau de signification est de 95 % ou plus et l'une des situations suivantes est vraie :
  - La limite supérieure de l'intervalle de confiance à 95 % est supérieure à zéro et la métrique doit augmenter
  - La limite inférieure de l'intervalle de confiance à 95 % est inférieure à zéro et la métrique doit diminuer
- Best (Meilleur) : l'expérience comporte au moins deux variations testées en plus de la variation par défaut, et les conditions suivantes sont remplies :
  - La variation répond aux critères de la désignation Better (Mieux)
  - L'une des conditions suivantes est vraie :
    - La limite inférieure de l'intervalle de confiance à 95 % est supérieure à la limite supérieure des intervalles de confiance à 95 % de toutes les autres variations et la métrique doit augmenter
    - La limite supérieure de l'intervalle de confiance à 95 % est inférieure à la limite inférieure des intervalles de confiance à 95 % de toutes les autres variations et la métrique doit diminuer

Analyse bayésienne dans Evidently

Avec l'analyse bayésienne, vous pouvez calculer la probabilité que la moyenne dans la variation testée soit supérieure ou inférieure à la moyenne dans la variation par défaut. Evidently effectue une inférence bayésienne pour la moyenne de la métrique cible en utilisant des a priori conjugués. Avec

des a priori conjugués, Evidently peut déduire plus efficacement la distribution a posteriori nécessaire à l'analyse bayésienne.

Evidently attend la date de fin de l'expérience pour calculer les résultats de l'analyse bayésienne. La page de résultats affiche les éléments suivants :

- probability of increase (probabilité d'augmentation) : la probabilité que la moyenne de la métrique dans la variation testée soit au moins 3 % supérieure à la moyenne dans la variation par défaut
- probability of decrease (probabilité de diminution) : la probabilité que la moyenne de la métrique dans la variation testée soit inférieure d'au moins 3 % à la moyenne dans la variation par défaut
- probability of no change (probabilité de non-changement) : la probabilité que la moyenne de la métrique dans la variation testée se situe à ±3 % de la moyenne dans la variation par défaut

La colonne Result (Résultat) indique la performance de la variation, et peut être l'une des suivantes :

- Better (Mieux) : la probabilité d'augmentation est d'au moins 90 % et la métrique doit augmenter, ou la probabilité de diminution est d'au moins 90 % et la métrique doit diminuer
- Worse (Pire) : la probabilité de diminution est d'au moins 90 % et la métrique doit augmenter, ou la probabilité d'augmentation est d'au moins 90 % et la métrique doit diminuer

Affichage des résultats du lancement dans le tableau de bord

Vous pouvez voir la progression et les résultats métriques d'une expérience pendant qu'elle est en cours et après qu'elle se termine.

Pour voir la progression et les résultats d'un lancement

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet qui contient le lancement.
4. Cliquez sur l'onglet Launch (Lancement).
5. Sélectionnez le nom du lancement.
6. Pour voir les étapes de lancement et les allocations de trafic pour chaque étape, sélectionnez l'onglet Launch (Lancement).

7. Pour voir le nombre de séances d'utilisateurs affectées à chaque variation au fil du temps et pour afficher les métriques de performance de chaque variation du lancement, sélectionnez l'onglet Monitoring (Surveillance).

   Cette vue indique également si des alertes de lancement sont passées à `ALARM`state (état) pendant le lancement.
8. Pour voir les variations, les métriques, les alertes et les balises de ce lancement, sélectionnez l'onglet Configuration (Configuration).

Affichage des résultats des expériences dans le tableau de bord

Vous pouvez voir les résultats statistiques d'une expérience pendant qu'elle est en cours et une fois qu'elle est terminée. Les résultats de l'expérience sont disponibles jusqu'à 63 jours après le début de l'expérience. Ils ne sont plus disponibles par la suite en raison des politiques de conservation CloudWatch des données.

Aucun résultat statistique n'est affiché tant que chaque variation n'a pas eu au moins 100 événements.

Evidently effectue une analyse de valeur p hors ligne supplémentaire à la fin de l'expérience. L'analyse des valeurs p hors ligne peut détecter la signification statistique dans certains cas où les valeurs p utilisées à tout moment pendant l'expérience ne trouvent pas de signification statistique.

Pour plus d'informations sur la façon dont CloudWatch Eviently calcule les résultats des expériences, voir. Comment Evidently calcule les résultats

Pour consulter les résultats d'une expérience

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez le nom du projet qui contient l'expérience.
4. Cliquez sur l'onglet Experiments (Expériences).
5. Sélectionnez le nom de l'expérience, puis sélectionnez l'onglet Results (Résultats).
6. Par Performance de variation, il existe un contrôle dans lequel vous pouvez sélectionner les statistiques d'expérience à afficher. Si vous sélectionnez plus d'une statistique, Evidently affiche un graphique et un tableau pour chaque statistique.

   Chaque graphique et tableau affiche les résultats de l'expérience jusqu'à présent.

Chaque graphique peut afficher les résultats suivants. Vous pouvez utiliser la commande se trouvant à droite du graphique pour déterminer lequel des éléments suivants est affiché :

- Le nombre d'événements de séances d'utilisateurs enregistrés pour chaque variation.
- La valeur moyenne de la métrique qui est sélectionnée en haut du graphique, pour chaque variation.
- La pertinence statistique des expériences. Cela permet de comparer la différence pour la métrique sélectionnée en haut du graphique avec la variation par défaut et chacune des autres variations.
- Les limites de confiance supérieure et inférieure de 95 % sur la différence de la métrique sélectionnée, entre chacune des variations et la variation par défaut.

Le tableau affiche une ligne pour chaque variation. Pour chaque variation qui n'est pas celle par défaut, Evidently indique s'il a reçu suffisamment de données pour déclarer les résultats statistiquement pertinents. Il montre également si l'amélioration de la valeur statistique de la variation a atteint un niveau de confiance de 95 %.

Enfin, dans la colonne Résultat, Evidently fournit une recommandation sur les variations les plus performantes en fonction de cette statistique, ou du fait que les résultats ne sont pas concluants.

Comment CloudWatch Evidently collecte et stocke les données

Amazon collecte et stocke CloudWatch évidemment les données relatives aux configurations de projets afin que les clients puissent effectuer des expériences et des lancements. Les données comprennent les éléments suivants :

- Les métadonnées sur les projets, fonctions, lancements et expériences.
- Événements de métriques
- Données d'évaluation

Les métadonnées des ressources sont stockées dans Amazon DynamoDB. Les données sont cryptées au repos par défaut, à l'aide de Clés détenues par AWS. Ces clés sont un ensemble de AWS KMS clés qu'un utilisateur Service AWS possède et gère pour une utilisation multiple Comptes AWS. Les clients ne peuvent pas afficher, gérer ou auditer l'utilisation de ces clés. Les

clients ne sont pas non plus tenus de prendre des mesures ou de modifier les programmes pour protéger les clés qui chiffrent leurs données.

Pour plus d'informations, consultez Clés détenues par AWSle guide du AWS Key Management Service développeur.

Les événements de métriques et les événements d'évaluation Evently sont livrés directement aux sites appartenant aux clients.

Les données en transit sont automatiquement chiffrées à l'aide du protocole HTTPS. Ces données seront livrées à des sites appartenant aux clients.

Vous pouvez également choisir de stocker les événements d'évaluation dans Amazon Simple Storage Service ou Amazon CloudWatch Logs. Pour plus d'informations sur la manière dont vous pouvez sécuriser vos données dans ces services, consultez Activer le chiffrement des compartiments par défaut d'Amazon S3 et Chiffrer les données des CloudWatch journaux dans Logs using AWS KMS.

Récupération des données

Vous pouvez récupérer vos données à l'aide des API CloudWatch Evidently. Pour récupérer les données du projet, utilisez GetProjectou ListProjects.

Pour récupérer les données des fonctionnalités, utilisez GetFeatureou ListFeatures.

Pour récupérer les données de lancement, utilisez GetLaunchou ListLaunches.

Pour récupérer des données d'expérience GetExperiment, utilisez ListExperiments, ou GetExperimentResults.

Modification et suppression de données

Vous pouvez modifier et supprimer vos données à l'aide des API CloudWatch Evidently. Pour les données du projet, utilisez UpdateProjectou DeleteProject.

Pour les données relatives aux fonctionnalités, utilisez UpdateFeatureou DeleteFeature.

Pour les données de lancement, utilisez UpdateLaunchou DeleteLaunch.

Pour les données d'expérience, utilisez UpdateExperimentou DeleteExperiment.

Utilisation de rôles liés à un service pour Evidently

CloudWatch Utilise évidemment des rôles AWS Identity and Access Management liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM lié directement à Evidently. Les rôles liés au service sont prédéfinis par Eviently 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 simplifie la configuration d'Evidently, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. Evidently définit les autorisations de ses rôles liés à un service ; sauf définition contraire, seul Evidently peut endosser ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources Evidently sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services prenant en charge les rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM et recherchez les services présentant la mention Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Autorisations des rôles liés à un service pour Evidently

Utilise le rôle lié au service nommé AWSServiceRoleForCloudWatchEvidently— Permet à CloudWatch Evidently de gérer les AWS ressources associées pour le compte du client.

Le rôle AWSServiceRoleForCloudWatchEvidently lié à un service fait confiance aux services suivants pour assumer le rôle :

- `CloudWatch Evidently`

La politique d'autorisations de rôle nommée AmazonCloudWatchEvidentlyServiceRolePolicy permet à Evidently d'effectuer les actions suivantes sur les ressources spécifiées :

- Actions : `appconfig:StartDeployment`, `appconfig:StopDeployment`, `appconfig:ListDeployments` et `appconfig:TagResource` sur les clients lourds Evidently.

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. 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 Evidently

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous commencez à utiliser un client Evidently Thick dans l' AWS Management Console AWS API AWS CLI, Evidently 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 commencez à utiliser un client lourd Evidently, Evidently crée à nouveau le rôle lié à un service pour vous.

Modification d'un rôle lié à un service pour Evidently

Ne vous permet évidemment pas de modifier le rôle lié au AWSServiceRoleForCloudWatchEvidently service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour Evidently

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. Vous devez supprimer tous les projets Evidently qui utilisent des clients lourds.

> **Note**
>
> Si le service Evidently utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer Eviently les ressources utilisées par AWSServiceRoleForCloudWatchEvidently

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation, sélectionnez Application monitoring (Surveillance des applications) , Evidently.
3. Dans la liste des projets, cochez la case en regard des projets qui ont utilisé des clients lourds.
4. Choisissez Project actions (Actions sur le projet), Delete project (Supprimer le projet).

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, le AWS CLI, ou l' AWS API pour supprimer le rôle lié au AWSServiceRoleForCloudWatchEvidently service. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service Evidently

Evidently prend en charge l'utilisation des rôles liés à un service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez Régions et points de terminaison AWS.

CloudWatch De toute évidence, des quotas

CloudWatch Possède évidemment les quotas suivants.

Ressource	Quota par défaut
Projets	50 par région et par compte Vous pouvez demander une augmentation de quota.
Segments	500 par région et par compte Vous pouvez demander une augmentation de quota.
Quotas par projet	• Total de 100 fonctions • 500 lancements en tout • 50 lancements en cours • 500 expériences en tout • 50 expériences en cours Vous pouvez demander une augmentation pour tous ces quotas.

Ressource	Quota par défaut
Quotas d'API (tous les quotas sont par région)	• PutProjectEvents: 1 000 transactions par seconde (TPS) dans l'est des États-Unis (Virginie du Nord), dans l'ouest des États-Unis (Oregon) et en Europe (Irlande). 200 TPS dans toutes les autres régions. • EvaluateFeature: 1000 TPS dans l'est des États-Unis (Virginie du Nord), dans l'ouest des États-Unis (Oregon) et en Europe (Irlande). 200 TPS dans toutes les autres régions. • BatchEvaluateFeature: 50 POINTS PAR SECONDE • API Créer, lire, mettre à jour, supprimer (CRUD) : 10 TPS combinés pour toutes les API CRUD Vous pouvez demander une augmentation pour tous ces quotas.

Didacticiel : tests A/B avec l'exemple d'application Evidently

Cette section fournit un didacticiel sur l'utilisation d'Amazon CloudWatch Evidently pour les tests A/B. Ce tutoriel est l'exemple d'application Evidently, qui est une simple application react. L'exemple d'application sera configuré pour afficher une fonction `showDiscount` ou non. Lorsque la fonction est affichée à un utilisateur, le prix affiché sur le site Web d'achat affichait une réduction de 20 %.

En plus de montrer la réduction à certains utilisateurs et non à d'autres, dans ce tutoriel, vous configurez Evidently pour collecter des métriques de temps de chargement des pages à partir des deux variations.

> **Warning**
>
> Ce scénario nécessite que les utilisateurs IAM disposent d'un accès programmatique et d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces utilisateurs lorsqu'ils ne sont plus nécessaires. Les clés d'accès peuvent être mises à jour si nécessaire.

Pour plus d'informations, consultez la section Mise à jour des clés d'accès dans le guide de l'utilisateur IAM.

Étape 1 : télécharger l'exemple d'application

Commencez par télécharger l'exemple d'application Evidently.

Pour télécharger l'exemple d'application

1. Téléchargez l'exemple d'application à partir du compartiment Simple Storage Service (Amazon S3) suivant :

   ```
   https://evidently-sample-application.s3.us-west-2.amazonaws.com/evidently-sample-shopping-app.zip
   ```

2. Décompressez le package.

Étape 2 : ajouter le point de terminaison Evidently et configurer les informations d'identification

Ensuite, ajoutez la Région et le point de terminaison pour Evidently dans le fichier `config.js` du répertoire `src` dans l'exemple de package d'application, comme dans l'exemple suivant :

```
evidently: {
    REGION: "us-west-2",
    ENDPOINT: "https://evidently.us-west-2.amazonaws.com (https://evidently.us-west-2.amazonaws.com/)",
},
```

Vous devez également vous assurer que l'application est autorisée à appeler CloudWatch Evidently.

Pour accorder à l'exemple d'application les autorisations pour appeler Evidently

1. Fédérez sur votre AWS compte.
2. Créez un utilisateur IAM et associez la AmazonCloudWatchEvidentlyFullAccesspolitique à cet utilisateur.
3. Notez l'identifiant de la clé d'accès et la clé d'accès secrète de l'utilisateur IAM, car vous en aurez besoin lors de l'étape suivante.

4. Dans le même fichier `config.js` que vous avez modifié précédemment dans cette section, saisissez les valeurs de l'identifiant de la clé d'accès et de la clé d'accès secrète, comme dans l'exemple suivant :

```
credential: {
   accessKeyId: "Access key ID",
   secretAccessKey: "Secret key"
}
```

> **Important**
>
> Nous utilisons cette étape pour rendre l'exemple d'application aussi simple que possible pour vous. Nous vous déconseillons de placer vos informations d'identification utilisateur IAM dans votre application de production réelle. En lieu et place, nous vous recommandons d'utiliser Amazon Cognito pour l'authentification. Pour plus d'informations, consultez Intégration d'Amazon Cognito aux applications Web et mobiles.

Étape 3 : configurer le code pour l'évaluation des fonctions

Lorsque vous utilisez CloudWatch Evidently pour évaluer une fonctionnalité, vous devez utiliser l'EvaluateFeatureopération pour sélectionner de manière aléatoire une variante de fonctionnalité pour chaque session utilisateur. Cette opération attribue des séances d'utilisateurs à chaque variation de la fonction, en fonction des pourcentages que vous avez indiqués dans l'expérience.

Pour configurer le code d'évaluation des fonctions pour l'application de démonstration bookstore

1. Ajoutez le générateur de clients dans le fichier `src/App.jsx` pour que l'exemple d'application puisse appeler Evidently.

```
import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
    endpoint,
    region,
) => {
    const credentials = {
        accessKeyId: config.credential.accessKeyId,
        secretAccessKey: config.credential.secretAccessKey
```

```
    }
    return new Evidently({
        endpoint,
        region,
        credentials,
    });
};
```

2. Ajoutez ce qui suit dans la section de code `const App` pour initier le client.

```
  if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
```

3. Construisez `evaluateFeatureRequest` en ajoutant le code suivant. Ce code préremplit le nom du projet et le nom de la fonction que nous recommandons plus loin dans ce didacticiel. Vous pouvez les remplacer par vos propres noms de projets et de fonctions, à condition que vous spécifiez également ces noms de projet et de fonction dans la console Evidently.

```
const evaluateFeatureRequest = {
              entityId: id,
              // Input Your feature name
              feature: 'showDiscount',
              // Input Your project name'
              project: 'EvidentlySampleApp',
            };
```

4. Ajoutez le code à appeler Evidently pour l'évaluation des fonctions. Lorsque la demande est envoyée, Evidently attribue de manière aléatoire à la session utilisateur pour voir la fonction `showDiscount` ou non.

```
client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
    if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
    }
    getPageLoadTime()
 })
```

Étape 4 : configurer le code pour les métriques d'expérience

Pour la métrique personnalisée, utilisez l'API `PutProjectEvents` d'Evidently pour envoyer les résultats des métriques à Evidently. Les exemples suivants décrivent la configuration de la métrique personnalisée et l'envoi des données d'expérience à Evidently.

Ajoutez la fonction suivante pour calculer le temps de chargement de la page et utilisez `PutProjectEvents` pour envoyer les valeurs de métrique à Evidently. Ajoutez la fonction suivante dans `Home.tsx` et appelez cette fonction dans l'API `EvaluateFeature` :

```
const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
```

Voici à quoi devrait ressembler le fichier `App.js` après toutes les modifications que vous avez effectuées depuis son téléchargement.

```
import React, { useEffect, useState } from "react";
import { BrowserRouter as Router, Switch } from "react-router-dom";
import AuthProvider from "contexts/auth";
import CommonProvider from "contexts/common";
import ProductsProvider from "contexts/products";
import CartProvider from "contexts/cart";
import CheckoutProvider from "contexts/checkout";
import RouteWrapper from "layouts/RouteWrapper";
```

```
import AuthLayout from "layouts/AuthLayout";
import CommonLayout from "layouts/CommonLayout";
import AuthPage from "pages/auth";
import HomePage from "pages/home";
import CheckoutPage from "pages/checkout";
import "assets/scss/style.scss";
import { Spinner } from 'react-bootstrap';

import Evidently from 'aws-sdk/clients/evidently';
import config from './config';

const defaultClientBuilder = (
  endpoint,
  region,
) => {
  const credentials = {
    accessKeyId: config.credential.accessKeyId,
    secretAccessKey: config.credential.secretAccessKey
  }
  return new Evidently({
    endpoint,
    region,
    credentials,
  });
};

const App = () => {
  const [isLoading, setIsLoading] = useState(true);
  const [startTime, setStartTime] = useState(new Date());
  const [showDiscount, setShowDiscount] = useState(false);
  let client = null;
  let id = null;

  useEffect(() => {
    id = new Date().getTime().toString();
    setStartTime(new Date());
    if (client == null) {
      client = defaultClientBuilder(
        config.evidently.ENDPOINT,
        config.evidently.REGION,
      );
    }
    const evaluateFeatureRequest = {
      entityId: id,
```

```
      // Input Your feature name
      feature: 'showDiscount',
      // Input Your project name'
      project: 'EvidentlySampleApp',
    };

    // Launch
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
    });

    // Experiment
    client.evaluateFeature(evaluateFeatureRequest).promise().then(res => {
      if(res.value?.boolValue !== undefined) {
        setShowDiscount(res.value.boolValue);
      }
      getPageLoadTime()
    })

    setIsLoading(false);
  },[]);

  const getPageLoadTime = () => {
    const timeSpent = (new Date().getTime() - startTime.getTime()) * 1.000001;
    const pageLoadTimeData = `{
      "details": {
        "pageLoadTime": ${timeSpent}
      },
      "UserDetails": { "userId": "${id}", "sessionId": "${id}"}
    }`;
    const putProjectEventsRequest = {
      project: 'EvidentlySampleApp',
      events: [
        {
          timestamp: new Date(),
          type: 'aws.evidently.custom',
          data: JSON.parse(pageLoadTimeData)
        },
      ],
    };
    client.putProjectEvents(putProjectEventsRequest).promise();
  }
```

```
  return (
    !isLoading? (
    <AuthProvider>
      <CommonProvider>
        <ProductsProvider>
          <CartProvider>
            <CheckoutProvider>
              <Router>
                <Switch>
                  <RouteWrapper
                    path="/"
                    exact
                    component={() => <HomePage showDiscount={showDiscount}/>}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/checkout"
                    component={CheckoutPage}
                    layout={CommonLayout}
                  />
                  <RouteWrapper
                    path="/auth"
                    component={AuthPage}
                    layout={AuthLayout}
                  />
                </Switch>
              </Router>
            </CheckoutProvider>
          </CartProvider>
        </ProductsProvider>
      </CommonProvider>
    </AuthProvider> ) : (
      <Spinner animation="border" />
    )
  );
};

export default App;
```

Chaque fois qu'un utilisateur visite l'application exemple, une métrique personnalisée est envoyée à Evidently pour analyse. Evidently analyse chaque métrique et affiche les résultats en temps réel sur le tableau de bord d'Evidently. L'exemple suivant illustre une charge utile de métrique :

```
[ {"timestamp": 1637368646.468, "type": "aws.evidently.custom", "data": "{\"details
\":{\"pageLoadTime\":2058.002058},\"userDetails\":{\"userId\":\"1637368644430\",
\"sessionId\":\"1637368644430\"}}" } ]
```

Étape 5 : créer le projet, la fonction et l'expérience

Ensuite, vous créez le projet, la fonctionnalité et l'expérience dans la console CloudWatch Evidently.

Pour créer le projet, la fonction et l'expérience pour ce tutoriel

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Sélectionnez Create project (Créer un projet) et remplissez les champs. Vous devez utiliser **`EvidentlySampleApp`** pour que le nom du projet de l'exemple fonctionne correctement. Pour Evaluation event storage (Stockage d'événements d'évaluation), choisissez Don't store Evaluation events (Ne stockez pas les événements d'évaluation).

   Une fois les champs remplis, choisissez Create Project (Créer le projet).

   Pour en savoir plus, consultez Création d'un nouveau projet..
4. Une fois le projet créé, créez une fonction dans ce projet. Nommez la fonction**`showDiscount`**. Dans cette fonction, créez deux variations du type **`Boolean`** Nommez la première variation **`disable`** par une valeur de **`False`** et nommez la deuxième variation **`enable`** par une valeur de**`True`**.

   Pour plus d'informations sur la création d'une fonction, consultez Ajouter une fonction à un projet.
5. Une fois la création de la fonction terminée, créez une expérience dans le projet. Nommez l'expérience**`pageLoadTime`**.

   Cette expérience utilisera une métrique personnalisée appelée `pageLoadTime` qui mesure le temps de chargement de la page testée. Les métriques personnalisées pour les tests sont créées à l'aide d'Amazon EventBridge. Pour plus d'informations EventBridge, consultez Qu'est-ce qu'Amazon EventBridge ? .

   Pour créer cette métrique personnalisée, procédez comme suit lorsque vous créez l'expérience :

   - Sous Métriques, pour Source métrique, choisissez Métriques personnalisées.
   - Pour Metric name (Nom de métrique), saisissez **`pageLoadTime`**.

- Pour Goal (Objectif), sélectionnez Decrease (Diminuer). Cela indique que nous souhaitons qu'une valeur inférieure de cette métrique indique la meilleure variation de fonction.
- Pour Metric rule (Règle de métrique), saisissez ce qui suit.
  - Pour Entity ID (ID d'entité), saisissez **`UserDetails.userId`**.
  - Pour Value key (Valeur de la clé), saisissez **`details.pageLoadTime`**.
  - Pour Units (Unités), saisissez **`ms`**.
- Sélectionnez Add metric (Ajouter une métrique).

Pour Audiences (Publics ciblés), sélectionnez 100 % afin que tous les utilisateurs soient saisis dans l'expérience. Configurez la répartition du trafic entre les variations de 50 % chacune.

Ensuite, sélectionnez Create Experiment (Créer une expérience) pour créer l'expérience. Une fois que vous l'avez créée, elle ne démarre pas tant que vous n'avez pas demandé à Evidently de la démarrer.

Étape 6 : Commencez l'expérience et testez CloudWatch évidemment

Les dernières étapes consistent à démarrer l'expérience et à démarrer l'application exemple.

Pour démarrer l'expérience du tutoriel

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, Evidently.
3. Choisissez le EvidentlySampleAppprojet.
4. Sélectionnez l'onglet Experiments (Expériences).
5. Cliquez sur le bouton à côté pageLoadTimeet choisissez Actions, Démarrer l'expérience.
6. Sélectionnez une heure pour la fin de l'expérience.
7. Sélectionnez Start experiment (Démarrer une expérience).

   L'expérience démarre immédiatement.

Ensuite, démarrez l'exemple d'application Evidently à l'aide de la commande suivante :

```
npm install -f && npm start
```

Une fois l'application démarrée, vous serez affecté à l'une des deux variations de fonctions testées. L'une des variations affiche « 20 % de réduction », et l'autre ne l'affiche pas. Continuez à actualiser la page pour voir les différentes variations.

> **Note**
>
> Evidently a des évaluations collantes. Les évaluations des fonctions sont déterministes, ce qui signifie pour la même `entityId` et fonction, un utilisateur recevra toujours la même attribution de variation. Le seul moment où les affectations de variation changent est lorsqu'une entité est ajoutée à une substitution ou que le trafic d'expérience est composé. Toutefois, pour vous faciliter l'utilisation du tutoriel de l'application exemple, Evidently réaffecte l'évaluation de la fonction de l'application exemple à chaque fois que vous rafraîchissez la page, afin que vous puissiez expérimenter les deux variantes sans devoir ajouter des substitutions.

Dépannage

Nous vous recommandons d'utiliser `npm` version 6.14.14. Si vous constatez des erreurs concernant la création ou le démarrage de l'exemple d'application et que vous utilisez une version différente de npm, procédez comme suit.

Pour installer **`npm`** version 6.14.14

1. Utiliser un navigateur pour vous connecter à https://nodejs.org/download/release/v14.17.5/.
2. Téléchargez node-v14.17.5.pkg et exécutez ce pkg pour installer npm.

   Si vous voyez une erreur `webpack not found`, accédez au répertoire `evidently-sample-shopping-app` et essayez ce qui suit :

   a. Supprimez `package-lock.json`

   b. Supprimez `yarn-lock.json`

   c. Supprimez `node_modules`

   d. Supprimez la dépendance au Webpack dans `package.json`

   e. Exécutez les commandes suivantes :

   ```
   npm install -f && npm
   ```

Utiliser du CloudWatch rhum

Avec CloudWatch RUM, vous pouvez effectuer une surveillance réelle des utilisateurs afin de collecter et de visualiser les données côté client concernant les performances de votre application Web à partir de sessions utilisateur réelles en temps quasi réel. Les données que vous pouvez visualiser et analyser incluent les temps de chargement des pages, les erreurs côté client et le comportement des utilisateurs. Lorsque vous visualisez ces données, vous pouvez les voir agrégées ensemble ou réparties par navigateurs et appareils utilisés par vos clients.

Vous pouvez utiliser les données collectées pour identifier et corriger rapidement les problèmes de performance côté client. CloudWatch RUM vous aide à visualiser les anomalies des performances de votre application et à trouver les données de débogage pertinentes, telles que les messages d'erreur, les traces de pile et les sessions utilisateur. Vous pouvez également utiliser RUM pour comprendre la portée de l'impact sur les utilisateurs finaux, y compris le nombre d'utilisateurs, les géolocalisations et les navigateurs utilisés.

Les données d'utilisateur final que vous collectez pour CloudWatch RUM sont conservées pendant 30 jours, puis automatiquement supprimées. Si vous souhaitez conserver les événements RUM plus longtemps, vous pouvez demander au moniteur de l'application d'envoyer des copies des événements aux CloudWatch journaux de votre compte. Vous pouvez ensuite ajuster la période de rétention pour ce groupe de journaux.

Pour utiliser RUM, vous devez créer un moniteur d'application et fournir quelques informations. RUM génère un JavaScript extrait que vous pouvez coller dans votre application. Cet extrait importe le code du client web RUM. Le client web RUM capture les données provenant d'un pourcentage des sessions utilisateur de votre application, qui sont affichées dans un tableau de bord prédéfini. Vous pouvez spécifier le pourcentage de sessions utilisateur à partir desquelles collecter des données.

CloudWatch RUM est intégré à Application Signals, qui permet de découvrir et de surveiller les services de votre application, vos clients, les canaries Synthetics et les dépendances des services. Utilisez Application Signals pour consulter une liste ou une carte visuelle de vos services, consulter les métriques d’intégrité en fonction de vos objectifs de niveau de service (SLO) et effectuer une analyse descendante pour voir les suivis X-Ray corrélés afin de résoudre les problèmes de manière plus détaillée. Pour voir les demandes de page client RUM dans Application Signals, activez le suivi actif de X-Ray en créant un moniteur d’application ou en configurant manuellement le client Web RUM. Vos clients RUM sont affichés sur la carte des services connectée à vos services et sur la page Détails du service des services qu’ils appellent.

Le client web RUM est open source. Pour plus d'informations, consultez la section Client Web CloudWatch RUM.

Considérations sur les performances

Cette section décrit les considérations relatives aux performances liées à l'utilisation de CloudWatch RUM.

- Impact sur les performances de charge — Le client Web CloudWatch RUM peut être installé dans votre application Web sous forme de JavaScript module ou chargé dans votre application Web de manière asynchrone à partir d'un réseau de diffusion de contenu (CDN). Il ne bloque pas le processus de chargement de l'application. CloudWatch RUM est conçu pour qu'il n'y ait aucun impact perceptible sur le temps de chargement de l'application.
- Impact sur le temps d'exécution : le client Web RUM effectue un traitement pour enregistrer et envoyer les données RUM au service CloudWatch RUM. Comme les événements sont peu fréquents et que la quantité de traitement est faible, le CloudWatch RUM est conçu pour qu'il n'y ait aucun impact détectable sur les performances de l'application.
- Impact sur le réseau — Le client Web RUM envoie régulièrement des données au service CloudWatch RUM. Les données sont distribuées à intervalles réguliers pendant l'exécution de l'application et immédiatement avant le déchargement de l'application par le navigateur. Les données envoyées immédiatement avant le déchargement de l'application par le navigateur sont envoyées sous forme de balises beacon conçues pour n'avoir aucun impact détectable sur le temps de déchargement de l'application.

Tarification RUM

Avec CloudWatch RUM, vous devez payer des frais pour chaque événement RUM organisé par CloudWatch RUM. Chaque élément de données collecté à l'aide du client web RUM est considéré comme un événement RUM. Les exemples d'événements RUM incluent un affichage de page, une JavaScript erreur et une erreur HTTP. Vous pouvez déterminer quels types d'événements sont collectés par chaque moniteur d'application. Vous pouvez activer ou désactiver des options pour collecter des événements de télémétrie de performance, des JavaScript erreurs, des erreurs HTTP et des traces X-Ray. Pour de plus amples informations sur ces options, veuillez consulter Étape 2 : création d'un moniteur d'application et Informations collectées par le client Web CloudWatch RUM. Pour plus d'informations sur les tarifs, consultez Amazon CloudWatch Pricing.

Disponibilité dans les Régions

CloudWatch Le RUM est actuellement disponible dans les régions suivantes :

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Californie du Nord)
- USA Ouest (Oregon)
- Afrique (Le Cap)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Hyderabad)
- Asie-Pacifique (Melbourne)
- Asie-Pacifique (Osaka)
- Asia Pacific (Seoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Paris)
- Europe (Espagne)
- Europe (Stockholm)
- Europe (Zurich)
- Moyen-Orient (Bahreïn)
- Moyen-Orient (EAU)
- Amérique du Sud (São Paulo)

Rubriques

Politiques IAM relatives à l'utilisation du RUM CloudWatch

Pour pouvoir gérer entièrement CloudWatch RUM, vous devez être connecté en tant qu'utilisateur ou rôle IAM doté de la politique AmazonCloudWatchRUM FullAccess IAM. En outre, vous aurez peut-être besoin d'autres stratégies ou autorisations :

- Pour créer un moniteur d'applications qui crée un nouveau pool d'identités Amazon Cognito à des fins d'autorisation, vous devez disposer du rôle Admin IAM ou de la AdministratorAccesspolitique IAM.
- Pour créer un moniteur d'applications qui envoie des données à CloudWatch Logs, vous devez être connecté à un rôle ou à une politique IAM disposant des autorisations suivantes :

```
{
    "Effect": "Allow",
    "Action": [
        "logs:PutResourcePolicy"
    ],
    "Resource": [
        "*"
    ]
```

```
}
```

Les autres utilisateurs qui ont besoin de consulter les données CloudWatch RUM mais qui n'ont pas besoin de créer de ressources CloudWatch RUM peuvent bénéficier de la ReadOnlyAccess politique AmazonCloudWatchRUM.

Configuration d'une application pour utiliser CloudWatch RUM

Suivez les étapes décrites dans ces sections pour configurer votre application afin qu'elle commence à utiliser CloudWatch RUM pour collecter des données de performance à partir de sessions utilisateur réelles.

Rubriques

Étape 1 : Autorisez votre application à envoyer des données à AWS

Pour utiliser CloudWatch RUM, votre application doit disposer d'une autorisation.

Trois possibilités s'offrent à vous pour configurer l'autorisation :

- Laissez CloudWatch RUM créer un nouveau pool d'identités Amazon Cognito pour l'application. Cette méthode est la plus simple à configurer. Il s'agit de l'option par défaut.

  Le groupe d'identités contiendra une identité non authentifiée. Cela permet au client Web CloudWatch RUM d'envoyer des données à CloudWatch RUM sans authentifier l'utilisateur de l'application.

  Un rôle IAM est attaché au groupe d'identités Amazon Cognito. L'identité non authentifiée Amazon Cognito permet au client Web d'assumer le rôle IAM autorisé à envoyer des données à RUM. CloudWatch

- Utilisez un groupe d'identités Amazon Cognito existant. Dans ce cas, vous devez également modifier le rôle IAM attaché au groupe d'identités. Utilisez cette option pour les pools d'identités qui prennent en charge les utilisateurs non authentifiés. Vous ne pouvez utiliser des groupes d'identités que provenant de la même région.
- Utilisez l'authentification d'un fournisseur d'identité existant que vous avez déjà configuré. Dans ce cas, vous devez obtenir des informations d'identification du fournisseur d'identité et votre application doit transférer ces informations d'identification au client web RUM.

  Utilisez cette option pour les pools d'identités qui ne prennent en charge que les utilisateurs authentifiés.

Les sections suivantes contiennent plus de détails sur ces options.

CloudWatch RUM crée un nouveau pool d'identités Amazon Cognito

Il s'agit de l'option la plus simple à configurer. Si vous optez pour celle-ci, aucune autre étape de configuration n'est requise. Vous devez disposer d'autorisations administratives pour utiliser cette option. Pour plus d’informations, consultez Politiques IAM relatives à l'utilisation du RUM CloudWatch .

Avec cette option, CloudWatch RUM crée les ressources suivantes :

- Un nouveau groupe d'identités Amazon Cognito.
- Une identité Amazon Cognito non authentifiée. Celle-ci permet au client web RUM d'assumer un rôle IAM sans authentifier l'utilisateur de l'application.
- Le rôle IAM que le client web RUM assumera. La stratégie IAM attachée à ce rôle lui permet d'utiliser l'API `PutRumEvents` avec la ressource du moniteur d'application. En d'autres termes, elle permet au client web RUM d'envoyer des données à RUM.

Le client Web RUM utilise l'identité Amazon Cognito pour obtenir AWS des informations d'identification. Les AWS informations d'identification sont associées au rôle IAM. Le rôle IAM est autorisé à être utilisé `PutRumEvents` avec la AppMonitor ressource.

Amazon Cognito envoie le jeton de sécurité nécessaire pour permettre à votre application d'envoyer des données à CloudWatch RUM. L'extrait de JavaScript code généré par CloudWatch RUM inclut les lignes suivantes pour activer l'authentification.

```
   {
    identityPoolId: [identity pool id],  // e.g., 'us-west-2:EXAMPLE4a-66f6-4114-902a-
EXAMPLEbad7'
   }
);
```

Utilisation d'un groupe d'identités Amazon Cognito existant

Si vous choisissez d'utiliser un pool d'identités Amazon Cognito existant, vous devez le spécifier lorsque vous ajoutez l'application à CloudWatch RUM. Le groupe doit prendre en charge l'activation de l'accès à des identités non authentifiées. Vous ne pouvez utiliser des groupes d'identités que provenant de la même région.

Vous devez également ajouter les autorisations suivantes à la stratégie IAM attachée au rôle IAM associé à ce groupe d'identités.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountid]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

Amazon Cognito enverra ensuite le jeton de sécurité nécessaire pour permettre à votre application d'accéder CloudWatch à RUM.

Fournisseur tiers

Si vous choisissez l'authentification privée d'un fournisseur tiers, vous devez obtenir des informations d'identification du fournisseur d'identité et les transférer à AWS. Le meilleur moyen d'y parvenir consiste à utiliser un fournisseur de jetons de sécurité. Vous pouvez utiliser n'importe quel fournisseur de jetons de sécurité, y compris Amazon Cognito avec. AWS Security Token Service Pour plus d'informations AWS STS, consultez la section Welcome to the AWS Security Token Service API Reference.

Si vous souhaitez utiliser Amazon Cognito comme fournisseur de jetons dans ce scénario, vous pouvez configurer Amazon Cognito pour qu'il fonctionne avec un fournisseur d'authentification. Pour

plus d'informations, consultez Démarrez avec les groupes d'identités Amazon Cognito (identités fédérées).

Après avoir configuré Amazon Cognito pour qu'il fonctionne avec votre fournisseur d'identité, vous devez également effectuer les opérations suivantes :

- Créez un rôle IAM avec les autorisations suivantes. Votre application utilisera ce rôle pour accéder à AWS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "rum:PutRumEvents",
            "Resource": "arn:aws:rum:[region]:[accountID]:appmonitor/[app monitor
 name]"
        }
    ]
}
```

- Ajoutez ce qui suit à votre application pour qu'elle transmette les informations d'identification de votre fournisseur à CloudWatch RUM. Insérez cette ligne pour qu'elle s'exécute après qu'un utilisateur s'est connecté à votre application et que celle-ci a reçu les informations d'identification à utiliser pour accéder à AWS.

```
cwr('setAwsCredentials', {/* Credentials or CredentialProvider */});
```

Pour plus d'informations sur les fournisseurs d'informations d'identification dans le AWS JavaScript SDK, voir Configuration des informations d'identification dans un navigateur Web dans le guide du développeur v3 pour le SDK JavaScript, Configuration des informations d'identification dans un navigateur Web dans le guide du développeur v2 pour le SDK pour JavaScript, et @aws -sdk/credential-providers.

Vous pouvez également utiliser le SDK du client Web CloudWatch RUM pour configurer les méthodes d'authentification du client Web. Pour plus d'informations sur le SDK du client Web, consultez la section SDK du client Web CloudWatch RUM.

Étape 2 : création d'un moniteur d'application

Pour commencer à utiliser CloudWatch RUM avec votre application, vous devez créer un moniteur d'applications. Lorsque le moniteur d'applications est créé, RUM génère un JavaScript extrait que vous pouvez coller dans votre application. Cet extrait importe le code du client web RUM. Le client web RUM capture les données provenant d'un pourcentage des sessions utilisateur de votre application et les envoie à RUM.

Pour créer un moniteur d'application

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.
3. Choisissez Add app monitor (Ajouter un moniteur d'application).
4. Saisissez les informations et les paramètres de votre application :

   - Pour le nom du moniteur d'application, entrez un nom à utiliser pour identifier ce moniteur d'application dans la console CloudWatch RUM.
   - Pour Application domain (Domaine de l'application), saisissez le nom du domaine de premier niveau sur lequel votre application dispose d'une autorité administrative. Ce nom doit être au format de domaine d'URL.

     Choisissez Include sub domains (Inclure les sous-domaines) pour que le moniteur d'application collecte également des données de tous les sous-domaines situés dans le domaine de premier niveau.
5. Pour Configure RUM data collection (Configurer la collecte des données RUM), indiquez si vous souhaitez que le moniteur d'application collecte chacun des éléments suivants :

   - Performance telemetry (Télémétrie de performance) : collecte des informations sur le chargement des pages et les temps de chargement des ressources.
   - JavaScript erreurs — Recueille des informations sur les JavaScript erreurs non gérées signalées par votre application
   - HTTP errors (Erreurs HTTP) : collecte des informations sur les erreurs HTTP levées par votre application.

   La sélection de ces options fournit plus d'informations sur votre application, mais génère également davantage d'événements CloudWatch RUM et entraîne donc des frais supplémentaires.

Si vous ne sélectionnez aucun de ces éléments, le moniteur d'application continue de collecter les événements de démarrage de session et les ID de page. Ainsi, vous pouvez voir combien d'utilisateurs utilisent votre application, y compris des répartitions par type et version de système d'exploitation, type et version de navigateur, type d'appareil et emplacement.

6. Sélectionnez Cochez cette option pour autoriser le client Web CloudWatch RUM à définir des cookies si vous souhaitez pouvoir collecter des identifiants utilisateur et des identifiants de session à partir d'échantillons de sessions utilisateur. Les ID utilisateur sont générés aléatoirement par RUM. Pour plus d'informations, consultez CloudWatch Cookies du client Web RUM (ou technologies similaires).
7. Pour Session samples (Exemples de session), saisissez le pourcentage de sessions utilisateur qui seront utilisées pour collecter des données RUM. La valeur par défaut est 100 %. Si vous réduisez cette valeur, vous obtiendrez moins de données, mais vos frais seront réduits. Pour plus d'informations sur la tarification RUM, consultez Tarification RUM.
8. Les données d'utilisateur final que vous collectez pour CloudWatch RUM sont conservées pendant 30 jours, puis supprimées. Si vous souhaitez conserver des copies des événements RUM dans les CloudWatch journaux et configurer la durée de conservation de ces copies, choisissez Cocher cette option pour stocker les données de télémétrie de votre application dans votre compte CloudWatch Logs sous Stockage des données. Par défaut, le groupe de CloudWatch journaux Logs conserve les données pendant 30 jours. Vous pouvez ajuster la période de conservation dans la console CloudWatch Logs.
9. Pour Authorization (Autorisation), spécifiez s'il faut utiliser un groupe d'identités Amazon Cognito nouveau ou existant, ou utiliser un autre fournisseur d'identité. La création d'un nouveau groupe d'identités est l'option la plus simple et ne nécessite aucune autre étape de configuration. Pour plus d'informations, consultez Étape 1 : Autorisez votre application à envoyer des données à AWS.

   La création d'un nouveau groupe d'identités Amazon Cognito nécessite des autorisations administratives. Pour plus d'informations, consultez Politiques IAM relatives à l'utilisation du RUM CloudWatch .
10. (Facultatif) Par défaut, lorsque vous ajoutez l'extrait de code RUM à votre application, le client Web injecte la JavaScript balise pour surveiller l'utilisation dans le code HTML de toutes les pages de votre application. Pour modifier ce comportement, choisissez Configure pages (Configurer les pages), puis choisissez Include only these pages (Inclure uniquement ces pages) ou Exclude these pages (Exclure ces pages). Ensuite, spécifiez les pages à inclure ou à exclure.

Pour spécifier une page à inclure ou à exclure, saisissez ses URL complètes. Pour spécifier des pages supplémentaires, choisissez Add URL (Ajouter une URL).

11. Pour activer le AWS X-Ray suivi des sessions utilisateur échantillonnées par le moniteur de l'application, choisissez Suivi actif, puis sélectionnez Tracer mon service avec AWS X-Ray.

    Si vous sélectionnez cette option, les demandes `XMLHttpRequest` et `fetch` effectuées pendant les sessions utilisateur échantillonnées par le moniteur d'application sont suivies. Vous pouvez ensuite voir les suivis et les segments de ces sessions utilisateur dans le tableau de bord RUM, ainsi que sur les pages de la carte de suivi X-Ray et de détails des suivis X-Ray. Ces sessions utilisateur apparaîtront également sous forme de pages client dans Application Signals une fois que vous les aurez activées pour votre application.

    En apportant des modifications de configuration supplémentaires au client Web CloudWatch RUM, vous pouvez ajouter un en-tête de trace X-Ray aux requêtes HTTP afin de permettre le end-to-end suivi des sessions utilisateur jusqu'aux services AWS gérés en aval. Pour plus d’informations, consultez Activation du end-to-end traçage par rayons X.

12. (Facultatif) Pour ajouter des identifications au moniteur d'application, choisissezTags (Identifications), Add new tag (Ajouter une nouvelle identification).

    Ensuite, pour Key (Clé), saisissez un nom pour l'identification. Vous pouvez ajouter une valeur en option pour la balise dans Value (Valeur).

    Pour ajouter une autre étiquette, sélectionnez à nouveau Add new tag (Ajouter une nouvelle étiquette).

    Pour plus d'informations, consultez la section AWS Ressources de balisage.

13. Choisissez Add app monitor (Ajouter un moniteur d'application).

14. Dans la section Sample code (Exemple de code), vous pouvez copier l'extrait de code à utiliser pour ajouter à votre application. Nous vous recommandons de choisir JavaScriptou TypeScriptd'utiliser NPM pour installer le client Web CloudWatch RUM en tant que JavaScript module.

    Vous pouvez également choisir le format HTML pour utiliser un réseau de diffusion de contenu (CDN) afin d'installer le client Web CloudWatch RUM. L'inconvénient de l'utilisation d'un CDN est que le client web est souvent bloqué par des bloqueurs de publicités.

15. Choisissez Copy (Copier) ou Download (Télécharger), puis Done (Terminé).

(Facultatif) Étape 3 : modifiez manuellement l'extrait de code pour configurer le client Web CloudWatch RUM

Vous pouvez modifier l'extrait de code avant de l'insérer dans votre application, pour activer ou désactiver plusieurs options. Pour plus d'informations, consultez la documentation du client Web CloudWatch RUM.

Il y a trois options de configuration que vous devez absolument connaître. Celles-ci sont décrites dans ces sections.

Blocage de la collecte d'URL de ressource susceptibles de contenir des informations personnelles

Par défaut, le client Web CloudWatch RUM est configuré pour enregistrer les URL des ressources téléchargées par l'application. Ces ressources incluent des fichiers HTML, des images, des JavaScript fichiers CSS, des fichiers, etc. Pour certaines applications, les URL peuvent contenir des données d'identification personnelle (PII).

Si c'est le cas pour votre application, nous vous recommandons vivement de désactiver la collecte des URL de ressource en définissant `recordResourceUrl: false` dans la configuration de l'extrait de code avant de l'insérer dans votre application.

Enregistrement manuel des consultations de page

Par défaut, le client web enregistre les consultations de page lorsque la page se charge pour la première fois et lorsque l'API d'historique du navigateur est appelée. L'ID de page par défaut est `window.location.pathname`. Toutefois, dans certains cas, vous devrez peut-être remplacer ce comportement et utiliser l'application pour qu'elle enregistre les consultations de page par programmation. Cela vous permet de contrôler l'ID de page et le moment où il est enregistré. Prenons l'exemple d'une application Web dotée d'un URI avec un identifiant variable, tel que `/entity/123` ou `/entity/456`. Par défaut, CloudWatch RUM génère un événement d'affichage de page pour chaque URI avec un ID de page distinct correspondant au nom du chemin, mais vous pouvez plutôt les regrouper par le même identifiant de page. Pour ce faire, désactivez l'automatisation de la consultation de page du client Web à l'aide de la configuration `disableAutoPageView`, puis utilisez la commande `recordPageView` pour définir l'ID de page souhaité. Pour plus d'informations, consultez la section Configurations spécifiques à l'application sur GitHub.

Exemple de script intégré :

```
cwr('recordPageView', { pageId: 'entityPageId' });
```

JavaScript exemple de module :

```
awsRum.recordPageView({ pageId: 'entityPageId' });
```

Activation du end-to-end traçage par rayons X

Lorsque vous créez le moniteur d'application, sélectionnez Trace my service with AWS X-Ray (Suivre mon service avec) pour activer le suivi des demandes `XMLHttpRequest` et `fetch` effectuées pendant les sessions utilisateur échantillonnées par le moniteur d'application. Vous pouvez ensuite voir les traces de ces requêtes HTTP dans le tableau de bord CloudWatch RUM, ainsi que sur les pages de détails de X-Ray Trace Map et Trace.

Par défaut, ces suivis côté client ne sont pas connectés à des suivis côté serveur en aval. Pour connecter les traces côté client aux traces côté serveur et activer le end-to-end suivi, définissez l'`addXRayTraceIdHeader`option sur `true` dans le client Web. Cela oblige le client Web CloudWatch RUM à ajouter un en-tête de trace X-Ray aux requêtes HTTP.

Le bloc de code suivant présente un exemple d'ajout de suivis côté client. Certaines options de configuration sont omises de cet exemple pour des raisons de lisibilité.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            enableXRay: true,
            telemetries: [
                'errors',
                'performance',
                [ 'http', { addXRayTraceIdHeader: true } ]
            ]
        }
    );
</script>
```

> **Warning**
>
> La configuration du client Web CloudWatch RUM pour ajouter un en-tête de trace X-Ray aux requêtes HTTP peut entraîner l'échec du partage de ressources entre origines (CORS) ou l'invalidation de la signature de la demande si celle-ci est signée avec SigV4. Pour plus d'informations, consultez la documentation du client Web CloudWatch RUM. Nous vous recommandons fortement de tester votre application avant d'ajouter un en-tête de suivi X-Ray côté client dans un environnement de production.

Pour plus d'informations, consultez la documentation du client Web CloudWatch RUM

Étape 4 : insertion de l'extrait de code dans l'application

Ensuite, vous devez insérer dans votre application l'extrait de code que vous avez créé dans la section précédente.

> **Warning**
>
> Le client web, téléchargé et configuré par l'extrait de code, utilise des cookies (ou des technologies semblables) pour vous aider à collecter les données des utilisateurs finaux. Avant d'insérer l'extrait de code, consultez Filtrage par attributs de métadonnées dans la console.

Si vous n'avez pas l'extrait de code précédemment généré, vous pouvez le trouver en suivant les instructions dans Comment puis-je trouver un extrait de code que j'ai déjà généré ?.

Pour insérer l'extrait de code CloudWatch RUM dans votre application

1. Insérez l'extrait de code que vous avez copié ou téléchargé dans la section précédente dans l'élément <head> de votre application. Insérez-le avant l'élément <body> ou toute autre identification <script>.

   L'extrait suivant représente un exemple d'extrait de code généré :

```
<script>
(function (n, i, v, r, s, c, x, z) {
    x = window.AwsRumClient = {q: [], n: n, i: i, v: v, r: r, c: c};
    window[n] = function (c, p) {
```

```
        x.q.push({c: c, p: p});
    };
    z = document.createElement('script');
    z.async = true;
    z.src = s;
    document.head.insertBefore(z, document.getElementsByTagName('script')[0]);
})('cwr',
    '194a1c89-87d8-41a3-9d1b-5c5cd3dafbd0',
    '1.0.0',
    'us-east-2',
    'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
    {
        sessionSampleRate: 1,
        identityPoolId: "us-east-2:c90ef0ac-e3b8-4d1a-b313-7e73cfd21443",
        endpoint: "https://dataplane.rum.us-east-2.amazonaws.com",
        telemetries: ["performance", "errors", "http"],
        allowCookies: true,
        enableXRay: false
    });
</script>
```

2. Si votre application est une application web multipage, vous devez répéter l'étape 1 pour chaque page HTML que vous souhaitez inclure dans la collecte de données.

Étape 5 : test de la configuration du moniteur d'application via la génération d'événements utilisateur

Une fois que vous avez inséré l'extrait de code et que votre application mise à jour est en cours d'exécution, vous pouvez effectuer un test en générant manuellement des événements utilisateur. Pour ce test, nous vous recommandons d'effectuer les opérations suivantes. Ce test entraîne des frais de CloudWatch RUM standard.

- Parcourez les pages de votre application web.
- Créez plusieurs sessions utilisateur à l'aide de différents navigateurs et appareils.
- Envoyez de demandes.
- JavaScript Provoquer des erreurs.

Après avoir généré certains événements, visualisez-les dans le tableau de bord CloudWatch RUM. Pour plus d’informations, consultez Consulter le tableau de bord CloudWatch du RUM.

L'affichage des données des sessions utilisateur dans le tableau de bord peut prendre jusqu'à 15 minutes.

Si vous ne voyez pas de données 15 minutes après avoir généré des événements dans l'application, reportez-vous à la section Résolution des problèmes liés CloudWatch à.

Configuration du client Web CloudWatch RUM

Vos applications peuvent utiliser l'un des extraits de code générés par CloudWatch RUM pour installer le client Web CloudWatch RUM. Les extraits générés prennent en charge deux méthodes d'installation : en tant que JavaScript module via NPM ou à partir d'un réseau de diffusion de contenu (CDN). Pour de meilleures performances, nous vous recommandons d'utiliser la méthode d'installation NPM. Pour plus d'informations sur l'utilisation de cette méthode, consultez la section Installation en tant que JavaScript module.

Si vous utilisez l'option d'installation du CDN, les bloqueurs de publicité peuvent bloquer le CDN par défaut fourni par RUM. CloudWatch Cela désactive la surveillance des applications pour les utilisateurs qui ont installé des bloqueurs de publicité. Pour cette raison, nous vous recommandons d'utiliser le CDN par défaut uniquement pour l'intégration initiale avec CloudWatch RUM. Pour plus d'informations sur les moyens d'atténuer ce problème, consultez Instrument the application (Instrumentation de l'application).

L'extrait de code se trouve dans l'identification <head> d'un fichier HTML et installe le client web en le téléchargeant, puis en le configurant pour l'application qu'il surveille. Cet extrait est une fonction auto-exécutable qui ressemble à l'exemple suivant. Dans cet exemple, le corps de la fonction de l'extrait de code a été omis pour des raisons de lisibilité.

```
<script>
(function(n,i,v,r,s,c,u,x,z){...})(
'cwr',
'00000000-0000-0000-0000-000000000000',
'1.0.0',
'us-west-2',
'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
{ /* Configuration Options Here */ }
);
<script>
```

Arguments

L'extrait de code accepte six arguments :

- Un espace de noms permettant d'exécuter des commandes sur le client web, par exemple `'cwr'`
- L'ID du moniteur d'application, par exemple `'00000000-0000-0000-0000-000000000000'`
- La version de l'application, par exemple `'1.0.0'`
- La AWS région du moniteur de l'application, telle que `'us-west-2'`
- L'URL du client web, par exemple `'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js'`
- Des options de configuration spécifiques à l'application. Pour plus d'informations, consultez la section suivante.

Omission des erreurs

Le client Web CloudWatch RUM écoute tous les types d'erreurs qui se produisent dans vos applications. Si votre application émet JavaScript des erreurs que vous ne souhaitez pas afficher dans le tableau de bord CloudWatch RUM, vous pouvez configurer le client Web CloudWatch RUM pour filtrer ces erreurs afin de ne voir que les événements d'erreur pertinents sur le tableau de bord CloudWatch RUM. Par exemple, vous pouvez choisir de ne pas afficher certaines JavaScript erreurs dans le tableau de bord parce que vous avez déjà identifié un correctif pour celles-ci et que le volume de ces erreurs masque d'autres erreurs. Vous pouvez également ignorer les erreurs que vous ne pouvez pas corriger car elles appartiennent à une bibliothèque appartenant à un tiers.

Pour plus d'informations sur la manière d'instrumenter le client Web pour filtrer des JavaScript erreurs spécifiques, consultez l'exemple dans la section Erreurs dans la documentation Github du client Web.

Options de configuration

Pour plus d'informations sur les options de configuration disponibles pour le client Web CloudWatch RUM, consultez la documentation du client Web CloudWatch RUM

Régionalisation

Cette section illustre les stratégies d'utilisation CloudWatch du RUM avec des applications dans différentes régions.

Mon application Web est déployée dans plusieurs AWS régions

Si votre application Web est déployée dans plusieurs AWS régions, trois options s'offrent à vous :

- Déployez un moniteur d'application dans une seule région, dans un seul compte, qui sert toutes les régions.
- Déployez des moniteurs d'application distincts pour chaque région, dans des comptes uniques.
- Déployez des moniteurs d'application distincts pour chaque région, dans un seul compte.

L'avantage d'utiliser un seul moniteur d'application est que toutes les données seront centralisées dans une seule visualisation et que tous les journaux seront écrits dans le même groupe de CloudWatch journaux dans Logs. Avec un seul moniteur d'application, il y a une petite latence supplémentaire pour les requêtes et un point unique de défaillance.

L'utilisation de plusieurs moniteurs d'application élimine le point unique de défaillance, mais empêche la combinaison de toutes les données dans une seule visualisation.

CloudWatch RUM n'a pas été lancé dans certaines régions dans lesquelles mon application est déployée

CloudWatch Le RUM est lancé dans de nombreuses régions et possède une large couverture géographique. En installant le CloudWatch RUM dans les régions où il est disponible, vous pouvez bénéficier des avantages. Les utilisateurs finaux peuvent se trouver n'importe où tout en ayant leurs sessions incluses si vous avez configuré un moniteur d'application dans la région à laquelle ils se connectent.

Cependant, le CloudWatch RUM n'est pas encore lancé dans AWS GovCloud (USA Est), AWS GovCloud (USA Ouest) ou dans aucune autre région de Chine. Vous n'êtes pas en mesure d'envoyer des données à CloudWatch RUM depuis ces régions.

Utiliser des groupes de pages

Utilisez des groupes de pages pour associer différentes pages de votre application entre elles afin de pouvoir consulter des analyses agrégées pour des groupes de pages. Par exemple, vous souhaiterez peut-être voir les temps de chargement agrégés de toutes vos pages de destination.

Vous placez les pages dans des groupes de pages en ajoutant une ou plusieurs balises aux événements d'affichage des pages dans le client Web CloudWatch RUM. Les exemples suivants placent la page `/home` dans le groupe de pages intitulé `en` et le groupe de pages intitulé `landing`.

Exemple de script intégré

```
cwr('recordPageView', { pageId: '/home', pageTags: ['en', 'landing']});
```

JavaScript exemple de module

```
awsRum.recordPageView({ pageId: '/home', pageTags: ['en', 'landing']});
```

> **Note**
>
> Les groupes de pages sont destinés à faciliter l'agrégation des analyses sur différentes pages. Pour plus d'informations sur la façon de définir et de manipuler `pageIds` pour votre application, consultez la section Manually recording page views dans (Facultatif) Étape 3 : modifiez manuellement l'extrait de code pour configurer le client Web CloudWatch RUM.

Spécifier des métadonnées personnalisées

CloudWatch RUM associe des données supplémentaires à chaque événement sous forme de métadonnées. Les métadonnées d'événement se composent d'attributs sous forme de paires clé-valeur. Vous pouvez utiliser ces attributs pour rechercher ou filtrer des événements dans la console CloudWatch RUM. Par défaut, CloudWatch RUM crée des métadonnées pour vous. Pour en savoir plus sur les métadonnées par défaut, consultez Métadonnées des événements RUM.

Vous pouvez également utiliser le client Web CloudWatch RUM pour ajouter des métadonnées personnalisées aux événements CloudWatch RUM. Les métadonnées personnalisées peuvent inclure des attributs de session et des attributs de page.

Pour ajouter des métadonnées personnalisées, vous devez utiliser la version 1.10.0 ou ultérieure du client Web CloudWatch RUM.

Exigences et syntaxe

Chaque événement peut inclure jusqu'à 10 attributs personnalisés dans les métadonnées. Les exigences syntaxiques pour les attributs personnalisés sont les suivantes :

- Clés
  - 128 caractères maximum
  - Peut inclure des caractères alphanumériques, des deux-points (:) et des traits de soulignement (_)
  - Ne doit pas commencer par `aws:`.

  - Ne peut pas être entièrement constituée de l'un des mots-clés réservés énumérés dans la section suivante. Peut utiliser ces mots-clés dans le cadre d'un nom de clé plus long.
- Valeurs
  - 256 caractères maximum
  - Doit être une chaîne de caractères, un nombre ou une valeur booléenne

Mots-clés réservés

Vous ne pouvez pas utiliser les mots-clés réservés suivants comme noms de clés complets. Vous pouvez utiliser les mots-clés suivants comme partie d'un nom de clé plus long, tel que `applicationVersion`.

- `browserLanguage`
- `browserName`
- `browserVersion`
- `countryCode`
- `deviceType`
- `domain`
- `interaction`
- `osName`
- `osVersion`
- `pageId`
- `pageTags`
- `pageTitle`
- `pageUrl`
- `parentPageId`
- `platformType`
- `referrerUrl`
- `subdivisionCode`
- `title`
- `url`
- `version`

> **Note**
>
> CloudWatch RUM supprime les attributs personnalisés des événements RUM si un attribut inclut une clé ou une valeur non valide, ou si la limite de 10 attributs personnalisés par événement est déjà atteinte.

Ajout d'un attribut de session

Si vous configurez des attributs de session personnalisés, ils sont ajoutés à tous les événements d'une session. Vous configurez les attributs de session soit lors de l'initialisation du client Web CloudWatch RUM, soit lors de l'exécution à l'aide de la `addSessionAttributes` commande.

Par exemple, vous pouvez ajouter la version de votre application comme attribut de session. Ensuite, dans la console CloudWatch RUM, vous pouvez filtrer les erreurs par version afin de déterminer si un taux d'erreur accru est associé à une version particulière de votre application.

Ajout d'un attribut de session pendant l'initialisation, exemple NPM

La section de code en gras ajoute l'attribut de session.

```
import { AwsRum, AwsRumConfig } from 'aws-rum-web';

try {
  const config: AwsRumConfig = {
    allowCookies: true,
    endpoint: "https://dataplane.rum.us-west-2.amazonaws.com",
    guestRoleArn: "arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth",
    identityPoolId: "us-west-2:00000000-0000-0000-0000-000000000000",
    sessionSampleRate: 1,
    telemetries: ['errors', 'performance'],
    sessionAttributes: {
        applicationVersion: "1.3.8"
    }
  };

  const APPLICATION_ID: string = '00000000-0000-0000-0000-000000000000';
  const APPLICATION_VERSION: string = '1.0.0';
  const APPLICATION_REGION: string = 'us-west-2';

  const awsRum: AwsRum = new AwsRum(
```

```
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    config
  );
} catch (error) {
  // Ignore errors thrown during CloudWatch RUM web client initialization
}
```

Ajout d'un attribut de session au moment de l'exécution, exemple NPM

```
awsRum.addSessionAttributes({
    applicationVersion: "1.3.8"
})
```

Ajout d'un attribut de session pendant l'initialisation, exemple de script intégré

La section de code en gras ajoute l'attribut de session.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            sessionSampleRate:1,
            guestRoleArn:'arn:aws:iam::000000000000:role/RUM-Monitor-us-
west-2-000000000000-00xx-Unauth',
            identityPoolId:'us-west-2:00000000-0000-0000-0000-000000000000',
            endpoint:'https://dataplane.rum.us-west-2.amazonaws.com',
            telemetries:['errors','http','performance'],
            allowCookies:true,
            sessionAttributes: {
                applicationVersion: "1.3.8"
            }
        }
    );
</script>
```

Ajout d'un attribut de session au moment de l'exécution, exemple de script intégré

```
<script>
    function addSessionAttribute() {
        cwr('addSessionAttributes', {
            applicationVersion: "1.3.8"
        })
    }

</script>
```

Ajout d'attributs de page

Si vous configurez des attributs de page personnalisés, ils sont ajoutés à tous les événements de la page actuelle. Vous configurez les attributs de page soit lors de l'initialisation du client Web CloudWatch RUM, soit lors de l'exécution à l'aide de la `recordPageView` commande.

Par exemple, vous pouvez ajouter votre modèle de page comme attribut de page. Ensuite, dans la console CloudWatch RUM, vous pouvez filtrer les erreurs par modèles de page afin de déterminer si un taux d'erreur accru est associé à un modèle de page particulier de votre application.

Ajout d'un attribut de page pendant l'initialisation, exemple NPM

La section de code en gras ajoute l'attribut de page.

```
const awsRum: AwsRum = new AwsRum(
    APPLICATION_ID,
    APPLICATION_VERSION,
    APPLICATION_REGION,
    { disableAutoPageView:  true // optional }
);
awsRum.recordPageView({
    pageId:'/home',
    pageAttributes: {
      template: 'artStudio'
    }
});
const credentialProvider = new CustomCredentialProvider();
if(awsCreds) awsRum.setAwsCredentials(credentialProvider);
```

Ajout d'un attribut de page au moment de l'exécution, exemple NPM

```
awsRum.recordPageView({
    pageId: '/home',
```

```
    pageAttributes: {
        template: 'artStudio'
    }
});
```

Ajout d'un attribut de page pendant l'initialisation, exemple de script intégré

La section de code en gras ajoute l'attribut de page.

```
<script>
    (function(n,i,v,r,s,c,u,x,z){...})(
        'cwr',
        '00000000-0000-0000-0000-000000000000',
        '1.0.0',
        'us-west-2',
        'https://client.rum.us-east-1.amazonaws.com/1.0.2/cwr.js',
        {
            disableAutoPageView: true //optional
        }
    );
    cwr('recordPageView', {
       pageId: '/home',
       pageAttributes: {
           template: 'artStudio'
       }
    });
    const awsCreds = localStorage.getItem('customAwsCreds');
    if(awsCreds) cwr('setAwsCredentials', awsCreds)
</script>
```

Ajout d'un attribut de page au moment de l'exécution, exemple de script intégré

```
<script>
    function recordPageView() {
        cwr('recordPageView', {
            pageId: '/home',
            pageAttributes: {
                template: 'artStudio'
            }
        });
    }
</script>
```

Filtrage par attributs de métadonnées dans la console

Pour filtrer les visualisations de la console CloudWatch RUM à l'aide d'un attribut de métadonnées intégré ou personnalisé, utilisez la barre de recherche. Dans la barre de recherche, vous pouvez spécifier jusqu'à 20 termes de filtre sous la forme clé=valeur à appliquer aux visualisations. Par exemple, pour filtrer les données concernant uniquement le navigateur Chrome, vous pouvez ajouter le terme de filtre browserName=Chrome.

Par défaut, la console CloudWatch RUM récupère les 100 clés et valeurs d'attributs les plus courants à afficher dans le menu déroulant de la barre de recherche. Pour ajouter d'autres attributs de métadonnées comme termes de filtre, saisissez la clé et la valeur complètes de l'attribut dans la barre de recherche.

Un filtre peut inclure jusqu'à 20 termes de filtre, et vous pouvez enregistrer jusqu'à 20 filtres par moniteur d'application. Lorsque vous enregistrez un filtre, il est sauvegardé dans la liste déroulante Saved filters (Filtres enregistrés). Vous pouvez également supprimer un filtre enregistré.

Envoyer des événements personnalisés

CloudWatch RUM enregistre et ingère les événements listés dansInformations collectées par le client Web CloudWatch RUM. Si vous utilisez la version 1.12.0 ou ultérieure du client Web CloudWatch RUM, vous pouvez définir, enregistrer et envoyer des événements personnalisés supplémentaires. Vous définissez le nom du type d'événement et les données à envoyer pour chaque type d'événement que vous définissez. Chaque charge utile d'événement personnalisé peut atteindre 6 Ko.

Les événements personnalisés sont ingérés uniquement si les événements personnalisés sont activés dans le moniteur d'applications. Pour mettre à jour les paramètres de configuration de votre moniteur d'applications, utilisez la console CloudWatch RUM ou l'UpdateAppMonitorAPI.

Après avoir activé les événements personnalisés, puis défini et envoyé des événements personnalisés, vous pouvez les rechercher. Pour les rechercher, utilisez l'onglet Events de la console CloudWatch RUM. Recherchez en utilisant le type d'événement.

Exigences et syntaxe

Les événements personnalisés se composent d'un type d'événement et des détails de l'événement. Les conditions requises pour ces derniers sont les suivantes :

- Type d'événement

  - Il peut s'agir du type ou du nom de votre événement. Par exemple, le type d'événement intégré CloudWatch RUM appelé JsErrorpossède un type d'événement de`com.amazon.rum.js_error_event`.
  - Doit comporter entre 1 et 256 caractères.
  - Peut être une combinaison de caractères alphanumériques, de traits de soulignement, de tirets et de points.
- Détails de l'événement
  - Contient les données réelles que vous souhaitez enregistrer dans CloudWatch RUM.
  - Doit être un objet qui se compose de champs et de valeurs.

Exemples d'enregistrement d'événements personnalisés

Il existe deux manières d'enregistrer des événements personnalisés dans le client Web CloudWatch RUM.

- Utilisez l'`recordEvent`API du client Web CloudWatch RUM.
- Utilisez un plugin personnalisé.

Envoyer un événement personnalisé à l'aide de l'API **`recordEvent`**, exemple NPM

```
awsRum.recordEvent('my_custom_event', {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
)
```

Envoyer un événement personnalisé à l'aide de l'API **`recordEvent`**, exemple de script intégré

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
```

```
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Exemple d'envoi d'un événement personnalisé à l'aide d'un plugin personnalisé

```
// Example of a plugin that listens to a scroll event, and
// records a 'custom_scroll_event' that contains the timestamp of the event.
class MyCustomPlugin implements Plugin {
    // Initialize MyCustomPlugin.
    constructor() {
        this.enabled;
        this.context;
        this.id = 'custom_event_plugin';
    }
    // Load MyCustomPlugin.
    load(context) {
        this.context = context;
        this.enable();
    }
    // Turn on MyCustomPlugin.
    enable() {
        this.enabled = true;
        this.addEventHandler();
    }
    // Turn off MyCustomPlugin.
    disable() {
        this.enabled = false;
        this.removeEventHandler();
    }
    // Return MyCustomPlugin Id.
    getPluginId() {
        return this.id;
    }
    // Record custom event.
    record(data) {
        this.context.record('custom_scroll_event', data);
    }
    // EventHandler.
```

```
    private eventHandler = (scrollEvent: Event) => {
        this.record({timestamp: Date.now()})
    }
    // Attach an eventHandler to scroll event.
    private addEventHandler(): void {
        window.addEventListener('scroll', this.eventHandler);
    }
    // Detach eventHandler from scroll event.
    private removeEventHandler(): void {
        window.removeEventListender('scroll', this.eventHandler);
    }
}
```

Consulter le tableau de bord CloudWatch du RUM

CloudWatch RUM vous aide à collecter des données à partir des sessions utilisateur concernant les performances de votre application, notamment les temps de chargement des pages, le score Apdex, les navigateurs et les appareils utilisés, la géolocalisation des sessions utilisateur et les sessions comportant des erreurs. Toutes ces informations sont affichées dans un tableau de bord.

Pour afficher le tableau de bord RUM

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.

   L'onglet Overview (Présentation) affiche les informations collectées par l'un des moniteurs d'application que vous avez créés.

   La rangée supérieure de volets affiche les informations suivantes pour ce moniteur d'application :

   - Nombre de chargements de pages
   - Vitesse et durée moyennes du chargement des pages
   - Score Apdex
   - Statut de toutes les alertes associées au moniteur d'application

   Le score Application Performance Index (indice de performance des applications, Apdex) indique le niveau de satisfaction des utilisateurs finaux. Les scores vont de 0 (satisfaction minimale) à 1 (satisfaction maximale). Les scores sont basés uniquement sur la performance de l'application.

Les utilisateurs ne sont pas invités à évaluer l'application. Pour plus d'informations sur les scores Apdex, consultez Comment CloudWatch RUM définit les scores d'Apdex.

Plusieurs de ces panneaux comprennent des liens que vous pouvez utiliser pour examiner plus avant les données. Cliquez sur l'un de ces liens pour afficher une vue détaillée comprenant les onglets Performance, Erreurs, Requêtes HTTP, Sessions, Évanements, Navigateurs et Appareils et Parcours utilisateur en haut de l'écran.

3. Pour vous concentrer davantage, choisissez l'option Vue Liste, puis choisissez le nom du moniteur d'application sur lequel vous souhaitez vous concentrer. Cela affiche les onglets suivants pour le moniteur d'application sélectionné.

   - L'onglet Performance affiche des informations sur la performance des pages, y compris les temps de chargement, les informations de session, les informations de demande, les signaux web et les chargements de page dans le temps. Cette vue comprend des commandes permettant de basculer la vue pour mettre en évidence les éléments Pages loads (Chargements de pages), Requests (Demandes) ou Location (Emplacement).
   - L'onglet Erreurs affiche des informations sur les erreurs Javascript, y compris le message d'erreur le plus fréquemment vu par les utilisateurs et les appareils, ainsi que les navigateurs présentant le plus d'erreurs. Cette vue inclut un histogramme des erreurs et une liste des erreurs. Vous pouvez filtrer la liste des erreurs par utilisateur et par événement. Choisissez un message d'erreur pour obtenir plus de détails.
   - L'onglet Requêtes HTTP affiche les informations relatives aux requêtes HTTP, notamment l'URL de la requête comportant le plus d'erreurs et les appareils et navigateurs présentant le plus grand nombre d'erreurs. Cet onglet inclut un histogramme des requêtes, une liste des requêtes et une liste des erreurs réseau. Vous pouvez filtrer les listes par utilisateur et par événement. Choisissez un code de réponse ou un message d'erreur pour obtenir plus de détails sur la requête ou l'erreur réseau, respectivement.
   - L'onglet Sessions affiche les métriques de session. Cet onglet inclut un histogramme des événements de démarrage de session et une liste des sessions. Vous pouvez filtrer la liste des sessions par type d'événement, informations sur l'utilisateur et détails de l'événement. Choisissez une sessionId pour obtenir plus de détails sur une session.
   - L'onglet Événements affiche un histogramme des événements RUM et une liste des événements. Vous pouvez filtrer la liste des événements par type d'événement, informations sur l'utilisateur et détails de l'événement. Choisissez un événement RUM pour voir l'événement brut.

- L'onglet Browsers & Devices (Navigateurs et appareils) affiche des informations telles que la performance et l'utilisation de différents navigateurs et appareils utilisés pour accéder à votre application. Cette vue comprend des commandes permettant de basculer la vue pour mettre en évidence les éléments Navigateurs et Appareils.

  Si vous limitez la portée à un seul navigateur, les données sont réparties par version du navigateur.
- L'onglet User Journey (Parcours utilisateur) affiche les chemins que vos clients utilisent pour parcourir votre application. Vous pouvez voir où vos clients entrent dans votre application et de quelle page ils la quittent. Vous pouvez également voir les chemins qu'ils empruntent et le pourcentage de clients qui suivent ces chemins. Vous pouvez faire une pause sur un nœud pour obtenir plus de détails sur la page concernée. Vous pouvez choisir un chemin spécifique pour mettre en évidence les connexions et faciliter la visualisation.

4. (Facultatif) Sur n'importe lequel des six onglets, vous pouvez choisir le bouton Pages et sélectionner une page ou un groupe de pages dans la liste. Cela permet de n'afficher les données que d'une seule page ou d'un seul groupe de pages de votre application. Vous pouvez également marquer les pages et groupes de pages de la liste comme favoris.

Comment CloudWatch RUM définit les scores d'Apdex

Apdex (Application Performance Index) est une norme ouverte qui définit une méthode pour comparer et évaluer le temps de réponse des applications et générer des rapports sur celui-ci. Un score Apdex vous aide à comprendre et à identifier l'impact sur la performance de l'application au fil du temps.

Le score Apdex indique le niveau de satisfaction des utilisateurs finaux. Ce score est compris entre 0 (satisfaction minimale) à 1 (satisfaction maximale). Les scores sont basés uniquement sur la performance de l'application. Les utilisateurs ne sont pas invités à évaluer l'application.

Chaque score Apdex individuel se trouve dans l'un des trois seuils. En fonction du seuil Apdex et du temps de réponse réel de l'application, il existe trois types de performances, à savoir :

- Satisfait : le temps de réponse réel de l'application est inférieur ou égal au seuil Apdex. Pour le CloudWatch RUM, ce seuil est inférieur ou égal à 2000 ms.
- Tolérable : le temps de réponse réel de l'application est supérieur au seuil Apdex, mais inférieur ou égal à quatre fois le seuil Apdex. Pour le CloudWatch RUM, cette plage est comprise entre 2 000 et 8 000 ms.

- Frustrant : le temps de réponse réel de l'application est supérieur à quatre fois le seuil Apdex. Pour le CloudWatch RUM, cette plage est supérieure à 8 000 ms.

Le score Apdex total, compris entre 0 et 1, est calculé à l'aide de la formule suivante :

```
(positive scores + tolerable scores/2)/total scores * 100
```

CloudWatch métriques que vous pouvez collecter avec CloudWatch RUM

Le tableau de cette section répertorie les métriques que vous collectez automatiquement avec CloudWatch RUM. Vous pouvez consulter ces statistiques dans la CloudWatch console. Pour plus d'informations, consultez Affichage des métriques disponibles.

Vous pouvez également éventuellement envoyer des métriques étendues à CloudWatch ou CloudWatch Evidently. Pour plus d'informations, consultez Métriques étendues.

Ces métriques sont publiées dans l'espace de noms de métrique nommé `AWS/RUM`. Les métriques suivantes sont publiées avec la dimension `application_name`. La valeur de cette dimension est le nom du moniteur d'application. Certaines métriques sont également publiées avec des dimensions supplémentaires, comme indiqué dans le tableau.

Mesure	Unité	Description
`HttpStatusCodeCount`	Nombre	Le nombre de réponses HTTP dans l'application, en fonction de leur code d'état de réponse. Dimensions supplémentaires : • `event_details.response.status` est le code d'état de réponse, comme par exemple 200, 400, 404, etc.

Mesure	Unité	Description
		• `event_type` le type d'événement. Actuellement, la seule valeur possible pour cette dimension est `http`.
`Http4xxCount`	Nombre	Le nombre de réponses HTTP dans l'application, avec le code d'état de réponse 4xx. Ils sont calculés en fonction des événements `http_event` RUM qui génèrent des codes 4xx.
`Http5xxCount`	Nombre	Le nombre de réponses HTTP dans l'application, avec le code d'état de réponse 5xx. Ils sont calculés en fonction des événements `http_event` RUM qui génèrent des codes 5xx.

Mesure	Unité	Description
`JsErrorCount`	Nombre	Nombre d'événements d' JavaScript erreur ingérés.
`NavigationFrustratedCount`	Nombre	Le nombre d'événements de navigation avec une `duration` plus élevée que le seuil de tolérance, qui est de 8000 ms. La durée des événements de navigation est comptabilisée dans la métrique `PerformanceNavigationDuration` .
`NavigationSatisfiedCount`	Nombre	Le nombre d'événements de navigation avec une `duration` inférieure à l'objectif Apdex, qui est de 2000 ms. La durée des événements de navigation est comptabilisée dans la métrique `PerformanceNavigationDuration` .

Mesure	Unité	Description
`NavigationToleratedCount`	Nombre	Le nombre d'événements de navigation avec une `duration` comprise entre 2000 ms et 8000 ms. La durée des événements de navigation est comptabilisée dans la métrique `PerformanceNavigationDuration` .
`PageViewCount`	Nombre	Nombre d'événements de consultation de page ingérés par le moniteur de l'application. Ceci est calculé en comptant les événements `page_view_event` RUM.

Mesure	Unité	Description
`PerformanceResourceDuration`	Millisecondes	La `duration` d'un événement de ressources. Dimensions supplémentaires : • `event_details.file.type` est le type de fichier de l'événement de ressource, tel qu'une feuille de style, un document, une image, un script ou une police. • `event_type` le type d'événement. Actuellement, la seule valeur possible pour cette dimension est `resource`.
`PerformanceNavigationDuration`	Millisecondes	La `duration` d'un événement de navigation.

Mesure	Unité	Description
`RumEventPayloadSize`	Octets	La taille de chaque événement ingéré par CloudWatch RUM. Vous pouvez également utiliser la statistique `SampleCount` de cette métrique afin de surveiller le nombre d'événements qu'un moniteur d'application ingère.
`SessionCount`	Nombre	Le nombre d'événements de démarrage de session ingérés par le moniteur d'application. En d'autres termes, le nombre de nouvelles sessions démarrées.
`WebVitalsCumulativeLayoutShift`	Aucun	Effectue le suivi de la valeur des événements Cumulative Layout Shift.
`WebVitalsFirstInputDelay`	Millisecondes	Effectue le suivi de la valeur des événements First Input Delay.
`WebVitalsLargestContentfulPaint`	Millisecondes	Effectue le suivi de la valeur des événements Largest Contentful Paint.

Des métriques personnalisées et des métriques étendues que vous pouvez envoyer à CloudWatch et CloudWatch Evidently

Par défaut, les moniteurs de l'application RUM envoient des métriques à CloudWatch. Ces mesures et dimensions par défaut sont répertoriées dans CloudWatch les métriques que vous pouvez collecter avec CloudWatch RUM.

Vous pouvez également configurer un moniteur d'application pour exporter les métriques. Le moniteur de l'application peut envoyer des métriques étendues, des métriques personnalisées, ou les deux. Il peut les envoyer à CloudWatch ou à CloudWatch Eviently, ou aux deux.

- Métriques personnalisées : les métriques personnalisées sont des métriques que vous définissez. Avec les métriques personnalisées, vous pouvez utiliser n'importe quel nom de métrique et n'importe quel espace de noms. Pour obtenir les métriques, vous pouvez utiliser des événements personnalisés, des événements intégrés, des attributs personnalisés ou des attributs par défaut.

  Vous pouvez envoyer des métriques personnalisées aux deux CloudWatch et à CloudWatch Evidently.
- Métriques étendues : vous permet d'envoyer les métriques CloudWatch RUM par défaut à CloudWatch Evidently pour qu'elles soient utilisées dans les tests Evidently. Vous pouvez également envoyer n'importe laquelle des métriques CloudWatch RUM par défaut CloudWatch avec des dimensions supplémentaires. De cette façon, ces métriques peuvent donner une vue plus fine.

Rubriques

- Métriques personnalisées
- Métriques étendues

Métriques personnalisées

Pour envoyer des métriques personnalisées, vous devez utiliser les AWS API ou AWS CLI remplacer la console. Pour plus d'informations sur l'utilisation des AWS API, reportez-vous aux PutRumMetricsDestinationsections et BatchCreateRumMetricDefinitions.

Le nombre maximum de définitions de métriques étendues et de métriques personnalisées qu'une destination peut contenir est de 2 000. Pour chaque métrique personnalisée ou étendue que vous envoyez à chaque destination, chaque combinaison de nom de dimension et de valeur de dimension

est prise en compte dans cette limite. Cela compte également comme une métrique CloudWatch personnalisée pour la tarification.

L'exemple suivant montre comment créer une métrique personnalisée dérivée d'un événement personnalisé. Voici l'exemple d'événement personnalisé utilisé :

```
cwr('recordEvent', {
    type: 'my_custom_event',
    data: {
        location: 'IAD',
        current_url: 'amazonaws.com',
        user_interaction: {
            interaction_1 : "click",
            interaction_2 : "scroll"
        },
        visit_count:10
    }
})
```

Compte tenu de cet événement personnalisé, vous pouvez créer une métrique personnalisée qui compte le nombre de visites sur l'URL `amazonaws.com` depuis des navigateurs Chrome. La définition suivante crée une métrique nommée `AmazonVisitsCount` dans votre compte, dans l'espace de noms `RUM/CustomMetrics/PageVisits`.

```
{
    "AppMonitorName":"customer-appMonitor-name",
    "Destination":"CloudWatch",
    "MetricDefinitions":[
        {
            "Name":"AmazonVisitsCount",
            "Namespace":"PageVisit",
            "ValueKey":"event_details.visit_count",
            "UnitLabel":"Count",
            "DimensionKeys":{
                "event_details.current_url": "URL"
            },
            "EventPattern":"{\"metadata\":{\"browserName\":[\"Chrome\"]},\"event_type
\":[\"my_custom_event\"],\"event_details\": {\"current_url\": [\"amazonaws.com\"]}}"
        }
    ]
}
```

Métriques étendues

Si vous configurez des métriques étendues, vous pouvez faire l'une des deux choses suivantes ou les deux :

- Envoyez les métriques CloudWatch RUM par défaut à CloudWatch Evidently pour qu'elles soient utilisées dans les expériences Evidently. Seuls les WebVitalsLargestContentfulPaintindicateurs PerformanceNavigationDurationPerformanceResourceDuration, WebVitalsCumulativeLayoutShift, WebVitalsFirstInputDelay, et peuvent être envoyés à Evidently.
- Envoyez n'importe laquelle des métriques CloudWatch RUM par défaut à CloudWatch avec des dimensions supplémentaires afin qu'elles vous offrent une vue plus précise. Par exemple, vous pouvez voir les métriques spécifiques à un certain navigateur utilisé par vos utilisateurs, ou les métriques pour les utilisateurs dans une géolocalisation spécifique.

Pour plus d'informations sur les métriques CloudWatch RUM par défaut, consultezCloudWatch métriques que vous pouvez collecter avec CloudWatch RUM.

Le nombre maximum de définitions de métriques étendues et de métriques personnalisées qu'une destination peut contenir est de 2 000. Pour chaque métrique étendue ou personnalisée que vous envoyez à chaque destination, chaque combinaison de nom de dimension et de valeur de dimension compte comme une métrique étendue pour cette limite. Cela compte également comme une métrique CloudWatch personnalisée pour la tarification.

Lorsque vous envoyez des métriques étendues à CloudWatch, vous pouvez utiliser la console CloudWatch RUM pour créer des CloudWatch alarmes sur celles-ci.

Les métriques étendues sont facturées en tant que métriques CloudWatch personnalisées. Pour plus d'informations, consultez Tarification d'Amazon CloudWatch.

Les dimensions suivantes sont prises en charge pour les métriques étendues pour tous les noms de métriques que les moniteurs d'applications peuvent envoyer. Ces noms de métriques sont répertoriés dans CloudWatch métriques que vous pouvez collecter avec CloudWatch RUM.

- `BrowserName`

  Exemples de valeurs de dimension : `Chrome`, `Firefox`, `Chrome Headless`
- `CountryCode` Ceci utilise le format ISO-3166, avec des codes à deux lettres.

  Exemples de valeurs de dimension : `US`, `JP`, `DE`

- `DeviceType`

  Exemples de valeurs de dimension : `desktop`, `mobile`, `tablet`, `embedded`
- `FileType`

  Exemples de valeurs de dimension : `Image`, `Stylesheet`
- `OSName`

  Exemples de valeurs de dimension : `Linux`, `Windows, iOS`, `Android`
- `PageId`

Configuration des métriques étendues à l'aide de la console

Pour utiliser la console pour envoyer des métriques étendues à CloudWatch, procédez comme suit.

Pour envoyer des métriques étendues à CloudWatch Evidently, vous devez utiliser les AWS API ou AWS CLI remplacer la console. Pour plus d'informations sur l'utilisation des AWS API pour envoyer des métriques étendues à l'un ou à l'autre CloudWatch ou à Evidently, consultez PutRumMetricsDestinationet BatchCreateRumMetricDefinitions.

Pour utiliser la console pour configurer un moniteur d'applications et envoyer des métriques étendues RUM à CloudWatch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.
3. Choisissez List view (Vue en liste), puis le nom du moniteur d'application qui doit envoyer les métriques.
4. Choisissez l'onglet Configuration, puis RUM extended metrics (Métriques étendues RUM).
5. Choisissez Send metrics (Envoyer les métriques).
6. Sélectionnez un ou plusieurs noms de métriques à envoyer avec des dimensions supplémentaires.
7. Sélectionnez un ou plusieurs facteurs à utiliser comme dimensions pour ces métriques. Au fur et à mesure que vous faites vos choix, le nombre de métriques étendues que vos choix créent s'affiche dans Number of extended metrics (Nombre de métriques étendues).

   Ce nombre est calculé en multipliant le nombre de noms de métriques choisis par le nombre de dimensions différentes que vous créez. Ce nombre représente le nombre de métriques

personnalisées qui vous sont facturées. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

a. Pour envoyer une métrique avec l'ID de page comme dimension, choisissez Browse for page ID (Parcourir les ID de page), puis sélectionnez les ID de page à utiliser.

b. Pour envoyer une métrique avec le type d'appareil comme dimension, choisissez soit Desktop devices (Appareils de bureau), soit Mobile and tablets (Mobiles et tablettes).

c. Pour envoyer une métrique avec le système d'exploitation comme dimension, sélectionnez un ou plusieurs systèmes d'exploitation sous Operating system (Système d'exploitation).

d. Pour envoyer une métrique avec le type de navigateur comme dimension, sélectionnez un ou plusieurs navigateurs sous Browsers (Navigateurs).

e. Pour envoyer une métrique avec la géolocalisation comme dimension, sélectionnez un ou plusieurs emplacements sous Locations (Emplacements).

   Seuls les emplacements dans lesquels ce moniteur d'application a enregistré des métriques apparaîtront dans la liste de choix.

8. Lorsque vous avez terminé vos choix, sélectionnez Send metrics (Envoyer les métriques).
9. (Facultatif) Dans la liste Extended metrics (Métriques étendues), pour créer une alarme qui surveille l'une des métriques, choisissez Create alarm (Créer une alarme) dans la ligne de cette métrique.

   Pour des informations générales sur les CloudWatch alarmes, consultez Utilisation des CloudWatch alarmes Amazon. Pour un didacticiel sur le réglage d'une alarme sur une métrique étendue CloudWatch RUM, voirTutoriel : créer une métrique étendue et la déclencher.

Arrêt de l'envoi de métriques étendues

Pour utiliser la console pour arrêter l'envoi de métriques étendues

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.
3. Choisissez List view (Vue en liste), puis le nom du moniteur d'application qui doit envoyer les métriques.
4. Choisissez l'onglet Configuration, puis RUM extended metrics (Métriques étendues RUM).
5. Sélectionnez une ou plusieurs combinaisons de noms et de dimensions de métriques à arrêter d'envoyer. Puis, choisissez Actions, Delete (Supprimer).

Tutoriel : créer une métrique étendue et la déclencher

Ce didacticiel explique comment configurer une métrique étendue à envoyer CloudWatch, puis comment définir une alarme sur cette métrique. Dans ce didacticiel, vous allez créer une métrique qui permet de suivre JavaScript les erreurs dans le navigateur Chrome.

Pour configurer cette métrique étendue et définir une alarme sur celle-ci

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.
3. Choisissez List view (Vue en liste), puis le nom du moniteur d'application qui doit envoyer la métrique.
4. Choisissez l'onglet Configuration, puis RUM extended metrics (Métriques étendues RUM).
5. Choisissez Send metrics (Envoyer les métriques).
6. Sélectionnez JS ErrorCount.
7. Sous Browsers (Navigateurs), sélectionnez Chrome.

   Cette combinaison de JS ErrorCount et Chrome enverra une métrique étendue à CloudWatch. La métrique compte JavaScript les erreurs uniquement pour les sessions utilisateur utilisant le navigateur Chrome. Le nom de la métrique sera JsErrorCountet le nom de la dimension sera Browser.
8. Choisissez Send metrics (Envoyer les métriques).
9. Dans la liste des métriques étendues, choisissez Créer une alarme dans la ligne qui s'affiche JsErrorCountsous Nom et Chrome sous BrowserName.
10. Sous Spécifier la métrique et les conditions, vérifiez que le nom de la métrique et BrowserNameles champs sont préremplis avec les valeurs correctes.
11. Pour Statistic (Statistique), sélectionnez la statistique que vous voulez utiliser pour l'alarme. Average (Moyenne) est un bon choix pour ce type de métrique de comptage.
12. Pour Période, sélectionnez 5 minutes.
13. Sous Conditions, procédez comme suit :

    - Choisissez Static.
    - Choisissez Greater (Supérieur) pour spécifier que l'alarme doit passer à l'état ALARM lorsque le nombre d'erreurs est supérieur au seuil que vous allez spécifier.

- Sous than... (à…), saisissez le nombre correspondant au seuil d'alarme. L'alarme passe à l'état ALARM lorsque le nombre d'erreurs sur une période de 5 minutes dépasse ce nombre.

14. (Facultatif) Par défaut, l'alarme passe à l'état ALARM dès que le nombre d'erreurs dépasse le nombre seuil que vous avez défini pendant une période de 5 minutes. Vous pouvez, en option, modifier ce paramètre pour que l'alarme ne passe à l'état ALARM que si ce nombre est dépassé pendant plus d'une période de 5 minutes.

    Pour ce faire, sélectionnez Additional configuration (Configuration supplémentaire), puis pour Datapoints to alarm (Points de données à déclencher), indiquez le nombre de périodes de 5 minutes pendant lesquelles le nombre d'erreurs doit dépasser le seuil pour déclencher l'alarme. Par exemple, vous pouvez sélectionner 2 sur 2 pour que l'alarme se déclenche uniquement lorsque deux périodes de 5 minutes consécutives dépassent le seuil, ou 2 sur 3 pour que l'alarme se déclenche si deux des trois périodes de 5 minutes consécutives dépassent le seuil.

    Pour plus d'informations sur ce type d'évaluation d'alarme, consultez Évaluation d'une alerte.
15. Choisissez Suivant.
16. Pour Configure actions (Configurer les actions), indiquez ce qui doit se passer lorsque l'alarme passe à l'état d'alarme. Pour recevoir une notification avec Amazon SNS, procédez comme suit :

    - Sélectionnez Ajouter une notification.
    - Choisissez En alarme.
    - Sélectionnez une rubrique SNS existante ou créez-en une nouvelle. Si vous en créez un nouveau, indiquez-lui un nom et ajoutez-lui au moins une adresse e-mail.
17. Choisissez Suivant.
18. Saisissez un nom et une description facultative pour l'alarme, puis sélectionnez Next (Suivant).
19. Passez en revue les détails et sélectionnez Create alarm (Créer l'alarme).

Protection et confidentialité des données avec CloudWatch RUM

Le modèle de responsabilité AWS partagée s'applique à la protection et à la confidentialité des données dans Amazon CloudWatch RUM. Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure globale qui gère l'ensemble du AWS cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Pour plus d'informations sur la confidentialité des données, consultez la FAQ sur la confidentialité des données. Pour plus d'informations sur la protection des données en Europe, consultez le modèle de responsabilité AWS

partagée et le billet de blog sur le RGPD sur le blog AWS de sécurité. Pour plus de ressources relatives à la conformité aux exigences du RGPD, consultez le Centre du Règlement général sur la protection des données (RGPD).

Amazon CloudWatch RUM génère un extrait de code que vous pouvez intégrer dans le code de votre site Web ou de votre application Web, en fonction des données d'utilisateur final que vous souhaitez collecter. Le client web, téléchargé et configuré par l'extrait de code, utilise des cookies (ou des technologies semblables) pour vous aider à collecter les données des utilisateurs finaux. L'utilisation de cookies (ou de technologies semblables) est soumise aux réglementations en matière de confidentialité des données dans certaines juridictions. Avant d'utiliser Amazon CloudWatch RUM, nous vous recommandons vivement d'évaluer vos obligations de conformité conformément à la législation en vigueur, y compris les exigences légales applicables, afin de fournir des avis de confidentialité juridiquement adéquats et d'obtenir les consentements nécessaires à l'utilisation de cookies et au traitement (y compris la collecte) des données des utilisateurs finaux. Pour plus d'informations sur la manière dont le client Web utilise les cookies (ou des technologies similaires) et sur les données d'utilisateur final collectées par le client Web, consultez Informations collectées par le client Web CloudWatch RUM etCloudWatch Cookies du client Web RUM (ou technologies similaires).

Nous vous recommandons vivement de ne jamais placer d'informations d'identification sensibles, telles que les numéros de compte, les adresses e-mail ou toute autre information personnelle de vos utilisateurs finaux, dans des champs à structure libre. Toutes les données que vous entrez dans Amazon CloudWatch RUM ou dans d'autres services peuvent être incluses dans les journaux de diagnostic.

CloudWatch Cookies du client Web RUM (ou technologies similaires)

Le client Web CloudWatch RUM collecte certaines données sur les sessions utilisateur par défaut. Vous pouvez choisir d'activer les cookies pour que le client Web collecte un identifiant utilisateur et un identifiant de session qui persistent après le chargement des pages. L'ID utilisateur est généré aléatoirement par RUM.

Si ces cookies sont activés, RUM est en mesure d'afficher les types de données suivants lorsque vous consultez le tableau de bord RUM de ce moniteur d'application.

- Des données agrégées basées sur les ID utilisateur, comme le nombre d'utilisateurs uniques et le nombre d'utilisateurs différents ayant rencontré une erreur.
- Des données agrégées basées sur les ID de session, comme le nombre de sessions et le nombre de sessions ayant rencontré une erreur.

- Le parcours utilisateur, qui est la séquence de pages incluse dans chaque session utilisateur échantillonnée.

> **Important**
>
> Si vous n'activez pas ces cookies (ou technologies similaires), le client Web enregistre toujours certaines informations sur les sessions de l'utilisateur final, telles que le type/la version du navigateur, le type/la version du système d'exploitation, le type d'appareil, etc. Ces informations sont collectées pour fournir des informations agrégées spécifiques aux pages, telles que les signaux web, les consultations de page et les pages présentant des erreurs. Pour de plus amples informations sur les données enregistrées, veuillez consulter Informations collectées par le client Web CloudWatch RUM.

Informations collectées par le client Web CloudWatch RUM

Cette section décrit le PutRumEventsschéma qui définit la structure des données que vous pouvez collecter à partir de sessions utilisateur à l'aide de CloudWatch RUM.

Une PutRumEventsdemande envoie une structure de données avec les champs suivants à CloudWatch RUM.

- ID de ce lot d'événements RUM
- Détails du moniteur d'application, qui comprennent les éléments suivants :
  - ID du moniteur d'application
  - Version de l'application contrôlée
- Détails sur les utilisateurs, qui comprennent les éléments suivants. Ces informations ne sont collectées que si les cookies sont activés sur le moniteur d'application.
  - ID utilisateur généré par le client web
  - ID de session
- Tableau des événements RUM dans ce lot

Schéma d'événements RUM

La structure de chaque événement RUM comprend les champs suivants.

- ID de l'événement
- Horodatage
- Type d'événement
- Agent utilisateur
- Metadonnées
- Détails de l'événement RUM

Métadonnées des événements RUM

Les métadonnées incluent les métadonnées de page, les métadonnées de l'agent utilisateur, les métadonnées de géolocalisation et les métadonnées de domaine.

Métadonnées de page

Les métadonnées de page incluent les éléments suivants :

- ID de la page
- Titre de la page
- ID de la page parent. – Ces informations ne sont collectées que si les cookies sont activés sur le moniteur d'application.
- Profondeur d'interaction – Ces informations ne sont collectées que si les cookies sont activés sur le moniteur d'application.
- Balises de page – Vous pouvez ajouter des balises aux événements de page pour regrouper les pages. Pour plus d’informations, consultez Utiliser des groupes de pages.

Métadonnées de l'agent utilisateur

Les métadonnées de l'agent utilisateur incluent les éléments suivants :

- Langue du navigateur
- Nom du navigateur
- Version du navigateur
- Nom du système d'exploitation
- Version du système d'exploitation

- Type d'appareil
- Types de plateforme

Métadonnées de géolocalisation

Les métadonnées de géolocalisation incluent les éléments suivants :

- Code pays
- Code de subdivision

Métadonnées de domaine

Les métadonnées de domaine incluent le domaine d'URL.

Détails de l'événement RUM

Les détails d'un événement suivent l'un des types de schémas suivants, en fonction du type d'événement.

Événement de démarrage de session

Cet événement ne contient aucun champ. Ces informations ne sont collectées que si les cookies sont activés sur le moniteur d'application.

Schéma de consultation de page

Un événement Page view (Consultation de page) contient les propriétés suivantes. Vous pouvez configurer le client web pour désactiver la collecte des consultations de page. Pour plus d'informations, consultez la documentation du client Web CloudWatch RUM.

Nom	Type	Description
Page ID (ID de la page)	Chaîne	Un ID qui représente cette page de manière unique au sein de l'application. Par défaut, il s'agit du chemin de l'URL.
Parent page ID (ID de la page parent)	Chaîne	L'ID de la page sur laquelle l'utilisateur se trouvait lorsqu'il a accédé à la page actuelle. Ces informations ne sont

Nom	Type	Description
		collectées que si les cookies sont activés sur le moniteur d'application.
Interaction depth (Profondeur d'interaction)	Chaîne	Ces informations ne sont collectées que si les cookies sont activés sur le moniteur d'application.

JavaScript schéma d'erreur

JavaScript les événements d'erreur générés par l'agent contiennent les propriétés suivantes. Le client web ne collecte ces événements que si vous avez choisi de collecter la télémétrie des erreurs.

Nom	Type	Description
Error type (Type d'erreur)	Chaîne	Le nom de l'erreur, le cas échéant. Pour de plus amples informations, veuillez consulter Error.prototype.name. Certains navigateurs peuvent ne pas prendre en charge les types d'erreurs.
Error message (Message d'erreur)	Chaîne	Le message de l'erreur. Pour de plus amples informati ons, veuillez consulter Error.prototype.message. Si le champ d'erreur n'existe pas, il s'agit du message de l'événement d'erreur. Pour plus d'informations, consultez ErrorEvent. Les messages d'erreur peuvent varier d'un navigateur à l'autre.
Stack trace (Suivi de pile)	Chaîne	Le suivi de pile de l'erreur, le cas échéant, tronqué à 150 caractères. Pour de plus amples informations, veuillez consulter Error.prototype.stack. Certains navigateurs peuvent ne pas prendre en charge les suivis de pile.

Schéma d'événements DOM

Les événements DOM (Document Object Model) générés par l'agent contiennent les propriétés suivantes. Par défaut, ces événements ne sont pas collectés. Ils ne sont collectés que si vous activez la télémétrie des interactions. Pour plus d'informations, consultez la documentation du client Web CloudWatch RUM.

Nom	Type	Description
Événement	Chaîne	Le type d'événement DOM, comme un clic, un défilement ou un survol. Pour plus d'informations, consultez Référence des événements.
Element	Chaîne	Type d'élément DOM
Element ID (ID de l'élément)	Chaîne	Si l'élément qui a généré l'événement possède un ID, cette propriété stocke cet ID. Pour de plus amples informations, veuillez consulter Element.id.
CSSLocator	Chaîne	Localisateur CSS utilisé pour identifier l'élément DOM.
InteractionId	Chaîne	Un identifiant unique pour l'interaction entre l'utilisateur et l'interface utilisateur.

Schéma d'événement de navigation

Les événements de navigation ne sont collectés que si la télémétrie de performance est activée sur le moniteur d'application.

Les événements de navigation utilisent les API Navigation timing Level 1 et Navigation timing Level 2. Les API Level 2 ne sont pas prises en charge sur tous les navigateurs. Ces nouveaux champs sont donc facultatifs.

> **Note**
>
> Les métriques d'horodatage sont basées sur le DOM. HighResTimestamp Avec les API Level 2, par défaut, toutes les durées sont relatives à la `startTime`. Mais pour Level 1, la

métrique `navigationStart` est soustraite des métriques d'horodatage pour obtenir des valeurs relatives. Toutes les valeurs d'horodatage sont exprimées en millisecondes.

Les événements de navigation contiennent les propriétés suivantes.

Nom	Type	Description	Remarques
initiatorType	Chaîne	Représente le type de ressource qui a initié l'événement de performance.	Valeur : "navigation" Level 1 : "navigation" Level 2 : entryData .initiatorType
navigation Type	Chaîne	Représente le type de navigation. Cet attribut n'est pas obligatoire.	Valeur : cette valeur doit être l'un des éléments suivants. • `navigate` est une navigation démarrée par le choix d'un lien, la saisie d'une URL dans la barre d'adresse d'un navigateur, l'envoi d'un

Nom	Type	Description	Remarques
			formulaire ou l'initialisation via une opération de script autre que `reload` ou `back_forward` . • `reload` est une navigation réalisée via l'opération de rechargement du navigateur ou `location.reload()` . • `back_forward` est une navigation réalisée via l'opération de balayage de l'historique du navigateur.

Nom	Type	Description	Remarques
			• `prerender` est une navigation initiée par un indicateur de prérendu. Pour de plus amples informations, veuillez consulter Prerender.
startTime	Nombre	Indique quand l'événement est déclenché.	Valeur : 0 Level 1 : entryData.navigationStart - entryData.navigationStart Level 2 : entryData.startTime

Nom	Type	Description	Remarques
unloadEventStart	Nombre	Indique l'heure à laquelle le déchargement du document précédent de la fenêtre a commencé après que l'événement `unload` a été levé.	Valeur : s'il n'y a pas de document précédent ou si le document précédent ou l'une des redirections nécessaires ne sont pas de la même origine, la valeur renvoyée est 0. Level 1 : `entryData.unloadEventStart > 0 ? entryData.unloadEventStart - entryData.navigationStart : 0` Niveau 2 : données d'entrée.

Nom	Type	Description	Remarques
			unloadEvent Start

Nom	Type	Description	Remarques
promptForUnload	Nombre	Temps nécessaire au déchargement du document. En d'autres termes, le temps entre `unloadEventStart` et `unloadEventEnd`. `UnloadEventEnd` représente le moment, en millisecondes, où le gestionnaire d'événements de déchargement se termine.	Valeur : s'il n'y a pas de document précédent ou si le document précédent ou l'une des redirections nécessaires ne sont pas de la même origine, la valeur renvoyée est 0. Niveau 1 : données d'entrée. unloadEventEnd - Données d'entrée. unloadEventStart Niveau 2 : données d'entrée. unloadEventEnd - Données d'entrée.

Nom	Type	Description	Remarques
			unloadEventStart
redirectCount	Nombre	Numéro représentant le nombre de redirections depuis la dernière navigation non redirigée dans le contexte de navigation actuel. Cet attribut n'est pas obligatoire.	Valeur : s'il n'y a pas de redirection ou s'il y a une redirection qui n'est pas de la même origine que le document de destination, la valeur renvoyée est 0. Level 1 : non disponible Level 2 : entryData.redirectCount

Nom	Type	Description	Remarques
redirectStart	Nombre	Heure de démarrage de la première redirection HTTP.	Valeur : s'il n'y a pas de redirection ou s'il y a une redirection qui n'est pas de la même origine que le document de destination, la valeur renvoyée est 0. Level 1 : `entryData.redirectStart > 0 ? entryData.redirectStart - entryData.navigationStart : 0` Level 2 : entryData.redirectStart

Nom	Type	Description	Remarques
redirectTime	Nombre	Le temps nécessaire à la redirection HTTP. Il s'agit de la différence entre `redirectStart` et `redirectEnd` .	Level 1 : entryData .redirectEnd - entryData .redirectStart Level 2 : entryData .redirectEnd - entryData .redirectStart

Nom	Type	Description	Remarques
workerStart	Nombre	Propriété de l'interface `PerformanceResourceTiming`. Elle marque le début de l'opération de thread de travail. Cet attribut n'est pas obligatoire.	Valeur : si un thread de service worker est déjà en cours d'exécution, ou immédiatement avant le démarrage du thread de service worker, cette propriété renvoie l'heure précédant immédiatement la distribution de `FetchEvent`. Elle renvoie 0 si la ressource n'est pas interceptée par un service worker. Level 1 : non disponible

Nom	Type	Description	Remarques
			Level 2 : entryData .workerStart
workerTime	Nombre	Si la ressource est interceptée par un service worker, cela renvoie le temps requis pour l'opération du thread de travail. Cet attribut n'est pas obligatoire.	Level 1 : non disponible Level 2 : `entryData .workerStart > 0 ? entryData .fetchStart - entryData .workerStart : 0`
fetchStart	Nombre	Heure à laquelle le navigateur est prêt à récupérer le document à l'aide d'une demande HTTP. Et ce, avant de vérifier n'importe quel cache d'application.	Level 1 : `: entryData .fetchStart > 0 ? entryData .fetchStart - entryData .navigationStart : 0` Level 2 : entryData .fetchStart

Nom	Type	Description	Remarques
domainLookupStart	Nombre	Heure de démarrage de la recherche de domaine.	Valeur : si une connexion persistante est utilisée ou si les informations sont stockées dans un cache ou une ressource locale, la valeur sera identique à `fetchStart` . Level 1 : `entryData.domainLookupStart > 0 ? entryData.domainLookupStart - entryData.navigationStart : 0` Niveau 2 : données d'entrée.

Nom	Type	Description	Remarques
			domainLoo kupStart
dns	Nombre	Temps nécessaire à la recherche de domaines.	Valeur : si les ressources et les enregistr ements DNS sont mis en cache, la valeur attendue est 0. Niveau 1 : données d'entrée. domainLoo kupEnd - Données d'entrée. domainLoo kupStart Niveau 2 : données d'entrée. domainLoo kupEnd - Données d'entrée. domainLoo kupStart

Nom	Type	Description	Remarques
nextHopProtocol	Chaîne	Chaîne représentant le protocole réseau utilisé pour récupérer la ressource. Cet attribut n'est pas obligatoire.	Level 1 : non disponible Niveau 2 : données d'entrée. nextHopProtocol

Nom	Type	Description	Remarques
connectStart	Nombre	Heure précédant immédiatement le lancement par l'agent utilisateur de l'établissement de la connexion au serveur pour récupérer le document.	Valeur : si une connexion persistante RFC2616 est utilisée ou si le document actuel est extrait à partir de ressources locales ou de caches d'applications pertinents, cet attribut renvoie la valeur de `domainLookupEnd` . Level 1 : `entryData.connectStart > 0 ? entryData.connectStart - entryData.navigationStart : 0`

Nom	Type	Description	Remarques
			Level 2 : entryData .connectStart
connect	Nombre	Mesure le temps nécessaire à l'établis sement des connexions de transport ou à l'authentification SSL. Il inclut également le temps bloqué qui est pris lorsqu'il y a trop de demandes simultanées émises par le navigateu r.	Level 1 : entryData .connectEnd - entryData .connectStart Level 2 : entryData .connectEnd - entryData .connectStart
secureCon nectionStart	Nombre	Si le modèle d'URL de la page actuelle est « https », cet attribut renvoie l'heure précédant immédiatement le lancement par l'agent utilisateur du processus de négociation pour sécuriser la connexion actuelle. Il renvoie 0 si HTTPS n'est pas utilisé. Pour plus d'informa tions sur les modèles d'URL, consultez URL representation.	Formule : EntryData. secureCon nectionStart

Nom	Type	Description	Remarques
tlsTime	Nombre	Le temps nécessaire pour terminer une négociation SSL.	Level 1 : `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0` Level 2 : `entryData.secureConnectionStart > 0 ? entryData.connectEnd - entryData.secureConnectionStart : 0`

Nom	Type	Description	Remarques
requestStart	Nombre	Heure précédant immédiatement le début de la demande de ressource par l'agent utilisateur auprès du serveur, des caches d'application pertinents ou des ressources locales.	Level 1 : `: entryData.requestStart > 0 ? entryData.requestStart - entryData.navigationStart : 0` Level 2 : entryData .requestStart
timeToFirstoctet	Nombre	Temps nécessaire à la réception du premier octet d'informations après l'envoi de la demande. Ce temps est relatif à la valeur `startTime` .	Level 1 : entryData .responseStart - entryData .requestStart Level 2 : entryData .responseStart - entryData .requestStart

Nom	Type	Description	Remarques
responseStart	Nombre	Heure suivant immédiatement la réception par l'analyseur HTTP de l'agent utilisateur du premier octet de la réponse des caches d'application concernés, des ressources locales ou du serveur.	Level 1 : `entryData.responseStart > 0 ? entryData.responseStart - entryData.navigationStart : 0` Level 2 : entryData.responseStart

Nom	Type	Description	Remarques
responseTime	Chaîne	Temps nécessaire à la réception d'une réponse complète sous la forme d'octets provenant des caches d'application concernés, des ressources locales ou du serveur.	Level 1 : `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0` Level 2 : `entryData.responseStart > 0 ? entryData.responseEnd - entryData.responseStart : 0`

Nom	Type	Description	Remarques
domIntera ctive	Nombre	Heure à laquelle l'analyseur a terminé son travail sur le document principal et que le DOM HTML est construit. À ce moment, la valeur `Document.readyState` passe à "interactive" (interactif) et l'événement `readystatechange` correspondant est levé.	Level 1 : `entryData .domInter active > 0 ? entryData .domInter active - entryData .navigati onStart : 0` Level 2 : entryData .domInter active

Nom	Type	Description	Remarques
domContentLoadedEventStart	Nombre	Représente la valeur temporelle égale à l'heure immédiatement avant que l'agent utilisateur ne déclenche l'ContentLoaded événement DOM sur le document en cours. L'Content Loaded événement DOM se déclenche lorsque le document HTML initial a été complètement chargé et analysé. À ce moment, le document HTML principal a terminé l'analyse, le navigateur commence à construire l'arborescence de rendu et les sous-ressources doivent encore être chargées. Cette propriété n'attend pas la fin du chargement des feuilles de style, des images et les sous-cadres.	Level 1 : `entryData.domContentLoadedEventStart > 0 ? entryData.domContentLoadedEventStart - entryData.navigationStart : 0` Niveau 2 : données d'entrée. domContentLoadedEventStart

Nom	Type	Description	Remarques
domContentLoaded	Nombre	Cette heure de début et de fin de la construction de l'arborescence de rendu est marquée par `domContentLoadedEventStart` et `domContentLoadedEventEnd`. Il permet à CloudWatch RUM de suivre l'exécution. Cette propriété correspond à la différence entre `domContentLoadedStart` et `domContentLoadedEnd`. Pendant ce temps, DOM et CSSOM sont prêts. Cette propriété attend l'exécution des scripts, à l'exception des scripts asynchrones et de ceux créés dynamiquement. Si les scripts dépendent de feuilles de style, `domContentLoaded` les attend aussi. Cette propriété n'attend pas les images. **Note** Les valeurs réelles de `domContentLoadedStart` et `domContentLoadedEnd` correspondent approximativement à `domContentLoaded` dans le panneau Réseau de Google Chrome. Celui-ci indique le temps de construction de l'arborescence de rendu DOM HTML + CSSOM depuis le début du processus de chargement de la page. Dans le cas des mesures de navigation, la valeur `domContentLoaded` représente la différence entre les valeurs de début et de fin, ce qui correspond au temps nécessaire au téléchargement des	Niveau 2 : données d'entrée. domContentLoadedEventEnd - Données d'entrée. domContentLoadedEventStart Niveau 2 : données d'entrée. domContentLoadedEventEnd - Données d'entrée. domContentLoadedEventStart

Nom	Type	Description	Remarques
		sous-ressources et à la construction de l'arborescence de rendu uniquement.	
domComplete	Nombre	Heure précédant immédiatement la définition par le navigateur de la préparation du document actuel sur terminée. À ce stade, le chargement des sous-ressources, telles que des images, est terminé. Cela inclut le temps nécessaire au téléchargement de contenus bloquants tels que le CSS et le contenu synchrone. JavaScript Cette valeur correspond approximativement à `loadTime` dans le panneau Réseau de Google Chrome.	Level 1 : `entryData.domComplete > 0 ? entryData.domComplete - entryData.navigationStart : 0` Level 2 : entryData.domComplete

Nom	Type	Description	Remarques
domProcessingTime	Nombre	Durée totale entre la réponse et le début de l'événement de chargement.	Niveau 1 : données d'entrée. loadEventStart - EntryData .ResponseEnd Niveau 2 : données d'entrée. loadEventStart - EntryData .ResponseEnd

Nom	Type	Description	Remarques
loadEvent Start	Nombre	Heure précédant immédiatement le lancement de l'événement `load` du document actif.	Level 1 : `entryData.loadEventStart > 0 ? entryData.loadEventStart - entryData.navigationStart : 0` Niveau 2 : données d'entrée. loadEvent Start

Nom	Type	Description	Remarques
loadEvent Time	Nombre	Différence entre `loadEventStart` et `loadEventEnd` . La logique ou les fonctions supplémentaires en attente de cet événement de chargement seront déclenchées pendant cette période.	Niveau 1 : données d'entrée. loadEvent End - Données d'entrée. loadEvent Start Niveau 2 : données d'entrée. loadEvent End - Données d'entrée. loadEvent Start
duration	Chaîne	Durée correspond au temps de chargement total de la page. Cette propriété enregistre la durée nécessaire au téléchargement de la page principale et de toutes ses sous-ressources synchrones, ainsi qu'au rendu de la page. Le téléchargement des ressources asynchrones telles que les scripts se poursuit ultérieurement. Cette propriété correspond à la différence entre les propriétés `loadEventEnd` et `startTime` .	Niveau 1 : données d'entrée. loadEvent End - Données d'entrée. Navigatio nStart Level 2 : entryData .duration

Nom	Type	Description	Remarques
headerSize	Nombre	Renvoie la différence entre `transferSize` et `encodedBodySize` . Cet attribut n'est pas obligatoire.	Level 1 : non disponible Niveau 2 : EntryData .TransferSize - EntryData . encodedBodySize Niveau 2 : EntryData .TransferSize - EntryData . encodedBodySize
compressionRatio	Nombre	Rapport entre `encodedBodySize` et `decodedBodySize` . La valeur de `encodedBodySize` correspond à la taille compressée de la ressource, à l'exclusion des en-têtes HTTP. La valeur de `decodedBodySize` correspond à la taille décompressée de la ressource, à l'exclusion des en-têtes HTTP. Cet attribut n'est pas obligatoire.	Level 1 : non disponible. Level 2 : `entryData.encodedBodySize > 0 ? entryData.decodedBodySize / entryData.encodedBodySize : 0`
navigationTimingLevel	Nombre	Version de l'API Navigation Timing.	Valeur : 1 ou 2

Schéma d'événement de ressource

Les événements de ressource ne sont collectés que si la télémétrie de performance est activée sur le moniteur d'application.

Les métriques d'horodatage sont basées sur le DOM typedef. HighResTimeStamp Avec les API Level 2, par défaut, toutes les durées sont relatives à la `startTime`. Mais pour les API Level 1, la métrique `navigationStart` est soustraite des métriques d'horodatage pour obtenir des valeurs relatives. Toutes les valeurs d'horodatage sont exprimées en millisecondes.

Les événements de ressource générés par l'agent contiennent les propriétés suivantes.

Nom	Type	Description	Remarques
targetUrl	Chaîne	Renvoie l'URL de la ressource.	Formule : entryData .name
initiatorType	Chaîne	Représente le type de ressource qui a initié l'événement de ressource de performance.	Valeur : "ressource" Formule : entryData .initiatorType
duration	Chaîne	Renvoie la différence entre les propriétés `responseEnd` et `startTime` . Cet attribut n'est pas obligatoire.	Formule : entryData .duration
transferSize	Nombre	Renvoie la taille (en octets) de la ressource récupérée, y compris les champs d'en-tête de réponse et le corps de la charge utile de réponse. Cet attribut n'est pas obligatoire.	Formule : entryData .transferSize
fileType	Chaîne	Extensions dérivées du modèle d'URL cible.	

Schéma d'événement Largest Contentful Paint

Les événements Largest Contentful Paint contiennent les propriétés suivantes.

Ces événements ne sont collectés que si la télémétrie de performance est activée sur le moniteur d'application.

Name (Nom)	Description		
Valeur	Pour de plus amples informations, veuillez consulter Web Vitals.		

Événement First Input Delay

Les événements First Input Delay contiennent les propriétés suivantes.

Ces événements ne sont collectés que si la télémétrie de performance est activée sur le moniteur d'application.

Name (Nom)	Description		
Valeur	Pour de plus amples informations, veuillez consulter Web Vitals.		

Événement Cumulative Layout Shift

Les événements Cumulative Layout Shift contiennent les propriétés suivantes.

Ces événements ne sont collectés que si la télémétrie de performance est activée sur le moniteur d'application.

Name (Nom)	Description		
Valeur	Pour de plus amples informations, veuillez consulter Web Vitals.		

Événement HTTP

Les événements HTTP peuvent contenir les propriétés suivantes. Ils contiendront un champ `Response` ou `Error`, mais pas les deux.

Ces événements ne sont collectés que si la télémétrie HTTP est activée sur le moniteur d'application.

Name (Nom)	Description
Demande	Le champ de requête inclut les éléments suivants : • Le champ `Method`, qui peut contenir des valeurs telles que GET, POST, etc. • L'URL.
Réponse	Le champ de réponse inclut les éléments suivants : • Statut, tel que 2xx, 4xx ou 5xx • Texte de statut
Error (Erreur)	Le champ d'erreur peut inclure les éléments suivants : • Type • Message • Nom de fichier • Numéro de ligne

Name (Nom)	Description
	• Numéro de colonne • Suivi de pile

Schéma d'événement de suivi X-Ray

Ces événements ne sont collectés que si le suivi X-Ray est activé sur le moniteur d'application.

Pour plus d'informations sur les schémas d'événements de suivi X-Ray, consultez Documents de segment AWS X-Ray.

Calendrier de changement de route pour les applications monopages

Dans une application multipage traditionnelle, lorsqu'un utilisateur demande le chargement d'un nouveau contenu, il demande en fait une nouvelle page HTML au serveur. Par conséquent, le client Web CloudWatch RUM capture les temps de chargement à l'aide des indicateurs de performance habituels de l'API.

Cependant, les applications Web à page unique utilisent JavaScript Ajax pour mettre à jour l'interface sans charger une nouvelle page depuis le serveur. Les mises à jour d'une seule page ne sont pas enregistrées par l'API de synchronisation du navigateur, mais utilisent la synchronisation des changements de route.

CloudWatch RUM prend en charge la surveillance des chargements de pages complètes depuis le serveur et des mises à jour d'une seule page, avec les différences suivantes :

- Pour la synchronisation des changements de route, il n'existe pas de mesures fournies par le navigateur, telles que `tlsTime`, `timeToFirstByte`, etc.
- Pour la synchronisation des changements de route, le champ `initiatorType` sera `route_change`.

Le client Web CloudWatch RUM écoute les interactions des utilisateurs susceptibles d'entraîner un changement d'itinéraire, et lorsqu'une telle interaction est enregistrée, le client Web enregistre un horodatage. Le minutage du changement de route commence alors si les deux conditions suivantes sont vraies :

- Une API d'historique du navigateur (à l'exception des boutons avant et arrière du navigateur) a été utilisée pour effectuer le changement de route.

- La différence entre l'heure de détection du changement de route et l'horodatage de la dernière interaction utilisateur est inférieure à 1 000 ms. Cela permet d'éviter une distorsion des données.

Ensuite, une fois que la synchronisation du changement de route commence, celle-ci se termine s'il n'y a pas de requêtes AJAX et de mutations DOM en cours. Ensuite, l'horodatage de la dernière activité terminée sera utilisé comme horodatage de fin.

Le délai de changement de route expire s'il y a des requêtes AJAX ou des mutations DOM en cours pendant plus de 10 secondes (par défaut). Dans ce cas, le client Web CloudWatch RUM n'enregistrera plus l'heure de ce changement d'itinéraire.

Par conséquent, la durée d'un événement de changement de route est calculée comme suit :

```
(time of latest completed activity) - (latest user interaction timestamp)
```

Gérez vos applications qui utilisent CloudWatch RUM

Suivez les étapes décrites dans ces sections pour gérer l'utilisation du CloudWatch RUM par vos applications.

Comment puis-je trouver un extrait de code que j'ai déjà généré ?

Pour trouver un extrait de code CloudWatch RUM que vous avez déjà généré pour une application, procédez comme suit.

Pour trouver un extrait de code que vous avez déjà généré

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.
3. Choisissez List view (Vue de liste).
4. À côté du nom du moniteur de l'application, choisissez Afficher JavaScript.
5. Dans le volet JavaScript Fragment de code, choisissez Copier dans le presse-papiers.

Modification de l'application

Pour modifier les paramètres d'un moniteur d'application, suivez ces étapes. Vous pouvez modifier tous les paramètres, à l'exception du nom du moniteur d'application.

Pour modifier la façon dont votre application utilise CloudWatch RUM

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.
3. Choisissez List view (Vue de liste).
4. Choisissez le bouton en regard du nom de l'application, puis choisissez Actions, Edit (Modifier).
5. Modifiez les paramètres, à l'exception du nom de l'application. Pour plus d'informations sur les paramètres, consultez Étape 2 : création d'un moniteur d'application.
6. Lorsque vous avez terminé, choisissez Save (Sauvegarder).

   La modification des paramètres modifie l'extrait de code. Vous devez maintenant coller l'extrait de code mis à jour dans votre application.
7. Une fois l'extrait de JavaScript code créé, choisissez Copier dans le presse-papiers ou Télécharger, puis cliquez sur OK.

   Pour démarrer la surveillance avec les nouveaux paramètres, vous devez insérer l'extrait de code dans votre application. Insérez l'extrait de code dans l'élément `<head>` de votre application, avant l'élément `<body>` ou toute autre identification `<script>`.

Arrêtez d'utiliser CloudWatch RUM ou supprimez un moniteur d'application

Pour arrêter d'utiliser CloudWatch RUM avec une application, supprimez l'extrait de code généré par RUM du code de votre application.

Pour supprimer un moniteur d'application RUM, suivez ces étapes.

Pour supprimer un moniteur d'application

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Application Signals, RUM.
3. Choisissez List view (Vue de liste).
4. Choisissez le bouton en regard du nom de l'application, puis choisissez Actions, Delete (Supprimer).
5. Dans la zone de confirmation, saisissez **`Delete`**, puis choisissez Delete (Supprimer).
6. Si ce n'est pas déjà fait, supprimez l'extrait de code CloudWatch RUM du code de votre application.

CloudWatch Quotas RUM

CloudWatch RUM a les quotas suivants.

Ressource	Quota par défaut
Moniteurs d'application	20 par compte Vous pouvez demander une augmentation de quota.
Taux d'ingestion de RUM	50 PutRumEventsdemandes par seconde (TPS). Vous pouvez demander une augmentation de quota.

Résolution des problèmes liés CloudWatch à

Cette section contient des conseils pour vous aider à résoudre les problèmes liés à CloudWatch RUM.

Aucune donnée n'est disponible pour mon application

Tout d'abord, assurez-vous que l'extrait de code a été correctement inséré dans votre application. Pour plus d’informations, consultez Étape 4 : insertion de l'extrait de code dans l'application.

Si le problème ne vient pas de là, il est possible que votre application n'ait pas encore reçu de trafic. Générez du trafic en accédant à votre application de la même manière qu'un utilisateur le ferait.

Les données ont cessé d'être enregistrées pour mon application

Votre application a peut-être été mise à jour et ne contient plus d'extrait de code CloudWatch RUM. Vérifiez le code de l'application.

Il est également possible que quelqu'un ait mis à jour l'extrait de code sans insérer ensuite l'extrait de code mis à jour dans l'application. Recherchez l'extrait de code correct actuel en suivant les instructions de la section Comment puis-je trouver un extrait de code que j'ai déjà généré ? et comparez-le à l'extrait de code collé dans votre application.

Surveillance réseau

Les rubriques de cette section décrivent les fonctionnalités de surveillance CloudWatch du réseau et d'Internet fournies par Amazon CloudWatch Internet Monitor et Amazon CloudWatch Network Monitor. Ces services vous aident à obtenir une visibilité opérationnelle sur les performances du réseau et d'Internet ainsi que sur la disponibilité de vos applications hébergées sur ce site AWS.

- Internet Monitor utilise les données de connectivité recueillies à partir AWS de son empreinte réseau mondiale pour calculer une base de référence en matière de performances et de disponibilité pour le trafic connecté à Internet. Vous pouvez avoir une vue globale des modèles de trafic et des problèmes de santé, et accéder facilement aux informations relatives aux événements. Vous pouvez également recevoir des alertes concernant des problèmes de santé liés à Internet qui affectent les clients de votre application. En outre, vous pouvez utiliser les informations fournies par Internet Monitor pour explorer les améliorations potentielles à apporter à votre expérience client, en utilisant Amazon CloudFront ou en utilisant différents itinéraires Régions AWS.
- Network Monitor utilise une approche d'agent entièrement gérée pour vous permettre de suivre et de visualiser la latence et la perte de paquets pour les connexions réseau hybrides. Pour recueillir des mesures et permettre à Network Monitor de créer des alertes d'événements médicaux pour votre application, vous devez créer des sondes qui sont envoyées depuis vos ressources hébergées AWS vers des adresses IP de destination sur site. Il n'est pas nécessaire d'installer des agents supplémentaires pour surveiller les performances de votre réseau. Comme avec Internet Monitor, vous pouvez définir des alertes et des seuils, obtenir des informations pour résoudre rapidement les problèmes, puis prendre des mesures pour améliorer l'expérience de l'utilisateur final.

Rubriques

Utilisation d'Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor fournit une visibilité sur l'impact des problèmes Internet sur les performances et la disponibilité entre vos applications hébergées sur AWS et vos utilisateurs finaux. Cela permet de réduire le temps nécessaire pour diagnostiquer les problèmes liés à Internet de plusieurs jours à quelques minutes. Internet Monitor utilise les données de connectivité recueillies

à partir AWS de son empreinte réseau mondiale pour calculer une base de référence en matière de performances et de disponibilité pour le trafic connecté à Internet. Il s'agit des mêmes données que celles AWS utilisées pour surveiller le temps de fonctionnement et la disponibilité d'Internet. Avec ces mesures comme base de référence, Moniteur Internet vous sensibilise lorsqu'il y a des problèmes significatifs pour vos utilisateurs finaux (clients) dans les différents emplacements géographiques où votre application s'exécute.

Dans la CloudWatch console Amazon, vous pouvez obtenir une vue globale des modèles de trafic et des événements sanitaires, et accéder facilement aux informations relatives aux événements, selon différentes granularités géographiques (emplacements). Vous pouvez visualiser clairement l'impact et déterminer où se trouvent les clients et les réseaux (ASN, généralement des fournisseurs de services Internet, ou FSI) concernés. Si Internet Monitor détermine qu'un problème de disponibilité ou de performance d'Internet est dû à un ASN spécifique ou au AWS réseau, il fournit ces informations.

Principales fonctionnalités du Moniteur Internet

- Moniteur Internet propose des idées et des recommandations qui peuvent vous aider à améliorer l'expérience de vos utilisateurs finaux. Vous pouvez découvrir, en temps quasi réel, comment améliorer la latence prévue de votre application en optant pour d'autres services ou en réacheminant le trafic vers votre charge de travail via différents Régions AWS moyens.
- Grâce au Moniteur Internet, vous pouvez rapidement identifier ce qui a un impact sur les performances et la disponibilité de votre application, afin de pouvoir localiser et résoudre les problèmes.
- Internet Monitor publie les mesures Internet dans CloudWatch Logs and CloudWatch Metrics, afin de faciliter l'utilisation d' CloudWatch outils contenant des informations sur l'état de santé des sites et des ASN (fournisseurs de services Internet) spécifiques à votre application. Vous pouvez également publier des mesures Internet sur Amazon S3.
- Internet Monitor envoie les événements médicaux à Amazon EventBridge afin que vous puissiez configurer des notifications. Si un problème est dû au AWS réseau, vous recevez également automatiquement une AWS Health Dashboard notification indiquant les mesures prises pour atténuer le problème. AWS

Comment utiliser Moniteur Internet

Pour utiliser Internet Monitor, vous devez créer un moniteur et y associer les ressources de votre application (VPC, équilibreurs de charge réseau, CloudFront distributions ou WorkSpaces répertoires) afin de permettre à Internet Monitor de savoir où se trouve le trafic Internet de votre

application. Internet Monitor publie ensuite des mesures Internet spécifiques aux réseaux urbains, c' AWS est-à-dire aux emplacements des clients et aux ASN (généralement des fournisseurs de services Internet ou ISP), par lesquels les clients accèdent à votre application. Pour plus d'informations, consultez Comment fonctionne Amazon CloudWatch Internet Monitor. Pour commencer à utiliser Moniteur Internet, consultez Commencer à utiliser Amazon CloudWatch Internet Monitor à l'aide de la console.

Table des matières

Compatible Régions AWS avec Amazon CloudWatch Internet Monitor

Les Régions AWS domaines dans lesquels Amazon CloudWatch Internet Monitor est pris en charge sont répertoriés dans cette section. Pour obtenir la liste actuelle des régions dans lesquelles Internet Monitor est pris en charge, y compris les régions optionnelles, consultez la section Points de

terminaison et quotas Amazon CloudWatch Internet Monitor dans le manuel Amazon Web Services General Reference.

Notez qu'Internet Monitor stocke les données d'un moniteur uniquement dans celui Région AWS dans lequel vous le créez, bien qu'un moniteur puisse inclure des ressources dans plusieurs régions.

Nom de la région (support par adhésion)	Région
Afrique (Le Cap)	`af-south-1`
Asie-Pacifique (Hong Kong)	`ap-east-1`
Asie-Pacifique (Hyderabad)	`ap-south-2`
Asie-Pacifique (Jakarta)	`ap-southeast-3`
Asie-Pacifique (Melbourne)	`ap-southeast-4`
Europe (Milan)	`eu-south-1`
Europe (Espagne)	`eu-south-2`
Europe (Zurich)	`eu-central-2`
Moyen-Orient (Bahreïn)	`me-south-1`
Moyen-Orient (EAU)	`me-central-1`

Nom de la région (support par défaut)	Région
USA Est (Ohio)	`us-east-2`
USA Est (Virginie du Nord)	`us-east-1`
USA Ouest (Californie du Nord)	`us-west-1`
USA Ouest (Oregon)	`us-west-2`
Asie-Pacifique (Mumbai)	`ap-south-1`

Nom de la région (support par défaut)	Région
Asie-Pacifique (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asie-Pacifique (Singapour)	ap-southeast-1
Asie-Pacifique (Sydney)	ap-southeast-2
Asie-Pacifique (Tokyo)	ap-northeast-1
Canada (Centre)	ca-central-1
Europe (Francfort)	eu-central-1
Europe (Irlande)	eu-west-1
Europe (Londres)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Amérique du Sud (São Paulo)	sa-east-1

Tarification d'Amazon CloudWatch Internet Monitor

Avec Amazon CloudWatch Internet Monitor, il n'y a aucun coût initial ni engagement à long terme. La tarification du Moniteur Internet comporte deux éléments : des frais par ressource surveillée et des frais par réseau urbain. Un réseau urbain est l'emplacement à partir duquel les clients accèdent aux ressources de votre application et le réseau (ASN, tel qu'un fournisseur de services Internet, ou FSI) par lequel les clients accèdent aux ressources. Notez que les CloudWatch prix standard vous sont également facturés pour les journaux et pour tous les indicateurs, tableaux de bord, alarmes ou informations supplémentaires que vous créez.

Vous choisissez un pourcentage du trafic à surveiller lorsque vous créez un moniteur. Pour mieux contrôler votre facture, vous pouvez également définir un nombre maximum de réseaux urbains à surveiller. Vous pouvez mettre à jour le pourcentage de trafic à surveiller ou la limite de réseaux urbains à tout moment en modifiant votre moniteur. Les 100 premiers réseaux urbains (tous

moniteurs confondus par compte) sont inclus. Ensuite, vous ne payez que pour les réseaux urbains supplémentaires que vous surveillez, dans la limite du nombre maximum.

Vous ne payez que pour les réseaux urbains supplémentaires que vous surveillez, dans la limite du nombre maximum, sans frais pour les 100 premiers réseaux urbains (tous moniteurs confondus par compte). Un montant forfaitaire équivalent au coût de 100 réseaux urbains est déduit de votre facture mensuelle.

Par exemple, une multinationale peut choisir de surveiller 100 % de son trafic Internet et de fixer un maximum de 50 000 réseaux urbains, pour un moniteur associé à une seule ressource. En supposant que le trafic atteigne 50 000 réseaux urbains, cette partie de sa facture s'élèverait à environ 2 700 USD/mois. Pour une autre entreprise, dans un nombre réduit de zones géographiques, avec un seul moniteur avec une seule ressource et 200 réseaux urbains, cette partie de la facture serait d'environ 13 dollars par mois. Pour plus d'informations, consultez Choisir une limite maximale de réseaux urbains.

Vous pouvez essayer différentes options avec le calculateur de prix. Pour explorer les options de tarification, sur la page Calculateur de prix, faites défiler CloudWatch la page vers le bas jusqu'à Internet Monitor.

Pour plus d'informations sur Internet Monitor et ses CloudWatch tarifs, consultez la page de CloudWatch tarification d'Amazon.

Composants et conditions d'Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor utilise ou fait référence aux éléments suivants.

Surveiller

Un moniteur inclut les ressources d'une seule application pour laquelle vous souhaitez consulter les mesures de performance et de disponibilité Internet, et pour laquelle vous souhaitez recevoir des alertes relatives aux événements de santé. Lorsque vous créez un moniteur pour une application, vous ajoutez des ressources afin que l'application définisse les villes (emplacements) que Moniteur Internet doit surveiller. Le Moniteur Internet utilise les modèles de trafic issus des ressources d'application que vous ajoutez pour publier des mesures de performance et de disponibilité Internet spécifiques aux emplacements et aux ASN (généralement des fournisseurs de services Internet, ou FSI) qui communiquent avec votre application. En d'autres termes, les ressources que vous ajoutez créent une portée des réseaux urbains que vous souhaitez que Moniteur Internet surveille et pour lesquels vous souhaitez qu'il publie des mesures.

Ressource ajoutée au moniteur (« ressource surveillée »)

Une ressource que vous ajoutez à un moniteur est une « ressource surveillée » dans Internet Monitor. C'est-à-dire :

- Chaque VPC que vous ajoutez dans une région est une ressource surveillée. Lorsque vous ajoutez un VPC, Internet Monitor surveille le trafic de toute application connectée à Internet dans le VPC, par exemple une application hébergée sur une instance Amazon EC2, derrière un Network Load Balancer ou un conteneur. AWS Fargate
- Chaque Network Load Balancer que vous ajoutez dans une région est une ressource surveillée.
- Chaque WorkSpaces répertoire que vous ajoutez dans une région est une ressource surveillée.
- Chaque CloudFront distribution que vous ajoutez est une ressource surveillée.

Numéro de système autonome (ASN)

Dans Moniteur Internet, un ASN fait généralement référence à un fournisseur de services Internet (FSI), tel que Verizon ou Comcast. Un ASN est un fournisseur de réseau qu'un client utilise pour accéder à votre application Internet. Un système autonome (AS) est un ensemble de préfixes de protocole Internet (IP) routables qui appartiennent à un réseau ou à une collection de réseaux qui sont tous gérés, contrôlés et supervisés par une organisation.

Réseau urbain (emplacement et ASN)

Un réseau urbain est l'emplacement (tel qu'une ville) à partir duquel les clients accèdent aux ressources de votre application et l'ASN, généralement un fournisseur de services Internet (ISP), par lequel les clients accèdent aux ressources. Pour vous aider à contrôler votre facture, vous pouvez définir une limite pour le nombre maximum de réseaux urbains qu'Internet Monitor doit surveiller pour chaque moniteur. Vous ne payez que pour les réseaux urbains que vous surveillez, dans la limite du nombre maximum. Pour plus d'informations, consultez Choosing a city-network maximum limit.

Mesures Internet

Internet Monitor publie les mesures Internet dans des fichiers CloudWatch journaux toutes les cinq minutes pour les 500 principaux réseaux urbains (sites clients et ASN, généralement des fournisseurs de services Internet ou des fournisseurs de services Internet) de votre compte. Ces mesures quantifient le score de performance, le score de disponibilité, le nombre d'octets transférés (octets entrants et sortants) et le temps de propagation aller et retour des réseaux urbains de votre application. Il s'agit de mesures pour les réseaux urbains spécifiques à vos VPC, à vos équilibreurs de charge réseau, à vos CloudFront distributions ou à vos annuaires.

WorkSpaces Vous pouvez choisir de publier les mesures et les événements Internet pour tous les réseaux urbains surveillés (jusqu'à la limite de service de 500 000 réseaux urbains) dans un compartiment Amazon S3.

Métriques

Internet Monitor génère des mesures agrégées pour les CloudWatch métriques, pour le trafic global vers votre application et le trafic global vers chacune d'entre elles Région AWS. Pour plus d'informations, consultez Utilisation CloudWatch des métriques avec Amazon CloudWatch Internet Monitor.

Événement de santé

Le Moniteur Internet crée un événement de santé pour vous alerter sur un problème spécifique qui affecte votre application. Le Moniteur Internet détecte les problèmes Internet, tels que l'augmentation de la latence du réseau, à travers le monde. Il utilise ensuite ses mesures Internet historiques relatives à l'ensemble de l'infrastructure AWS mondiale pour calculer l'impact des problèmes actuels sur votre application et créer des événements de santé. Par défaut, Moniteur Internet crée des événements de santé basés sur les seuils d'impact global et d'impact local. Pour plus d'informations sur la configuration des seuils, consultez Modifier les seuils relatifs aux événements de santé.

Chaque événement de santé comprend des informations sur les réseaux urbains touchés. Vous pouvez consulter les événements de santé dans la CloudWatch console, à l'aide du AWS SDK ou à l'aide AWS CLI des actions de l'API Internet Monitor. Internet Monitor envoie également des EventBridge notifications à Amazon pour les problèmes de santé. Pour plus d'informations, consultez Quand Internet Monitor crée et résout des problèmes de santé.

Événement sur Internet

Internet Monitor affiche des informations sur les récents événements sanitaires mondiaux, appelés événements Internet, sur une carte météo Internet accessible à tous les AWS clients. Il n'est pas nécessaire de créer un moniteur dans Internet Monitor pour afficher la carte météo sur Internet. Contrairement aux événements médicaux, les événements Internet ne sont pas spécifiques aux clients individuels ou au trafic de leurs applications. Pour plus d'informations, consultez Carte météo mondiale sur Internet dans Amazon CloudWatch Internet Monitor.

Seuils

Le Moniteur Internet crée des événements de santé basés à la fois sur les seuils globaux et sur les seuils locaux. Vous pouvez modifier les seuils par défaut et configurer d'autres options, telles

que la désactivation des seuils locaux. Pour plus d'informations sur la configuration des seuils, consultez Modifier les seuils relatifs aux événements de santé.

Scores de performance et de disponibilité

En analysant les données AWS collectées, Internet Monitor peut détecter les baisses de performances et de disponibilité de votre application, par rapport aux valeurs de référence estimées calculées par Internet Monitor. Pour faciliter la détection de ces baisses, Moniteur Internet vous communique ces informations sous forme de scores. Un score de performance représente le pourcentage estimé du trafic qui ne subit pas de baisse de performance. De même, un score de disponibilité représente le pourcentage estimé du trafic qui ne subit pas de baisse de disponibilité. Pour plus d'informations, voir Comment AWS calcule les scores de performance et de disponibilité.

Octets transférés et octets surveillés transférés

Le nombre d'octets transférés est le nombre total d'octets de trafic entrant et sortant entre une application AWS et le réseau urbain (c'est-à-dire l'emplacement et l'ASN, généralement le fournisseur de services Internet) où les clients accèdent à une application. Les octets surveillés transférés sont une métrique similaire, mais ils incluent uniquement les octets du trafic surveillé.

Temps de propagation aller et retour

Le temps de propagation aller et retour (RTT) est le temps qu'il faut pour qu'une demande d'un utilisateur client renvoie une réponse à l'utilisateur. Lorsque le RTT est agrégé entre les emplacements clients (villes ou autres zones géographiques), la valeur est pondérée en fonction de la part du trafic de votre application généré par chaque emplacement client.

Carte météo mondiale sur Internet dans Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor affiche une carte météo mondiale sur Internet accessible à tous les AWS clients. Pour afficher la carte, dans la CloudWatch console Amazon, accédez à Internet Monitor.

La carte met en évidence les événements Internet (« pannes ») qui affectent les AWS clients dans le monde entier, en indiquant les villes et les réseaux spécifiques (ASN, généralement des fournisseurs de services Internet) présentant des problèmes de performance ou de disponibilité. La carte météo Internet inclut les événements Internet des dernières 24 heures.

Il n'est pas nécessaire de créer un moniteur dans Internet Monitor pour afficher la carte météo sur Internet. Contrairement aux événements de santé dans Internet Monitor, les événements Internet ne sont pas spécifiques aux clients individuels ou au trafic de leurs applications.

Sur la carte météo Internet, vous pouvez choisir un événement Internet pour en savoir plus sur celui-ci. Pour un événement Internet, vous pouvez voir l'heure de début, l'heure de fin (si l'événement est terminé), l'état actuel (actif ou résolu) et le type de panne (disponibilité ou performance). Pour en savoir plus sur la façon dont la carte météo Internet est créée et sur ce qui est inclus, consultez la FAQ sur les cartes météorologiques mondiales sur Internet.

Pour consulter et utiliser des informations détaillées spécifiques au trafic de votre application et à l'emplacement des clients, vous pouvez facilement configurer un moniteur dans Internet Monitor pour votre application. Ensuite, vous découvrirez les modèles et les événements actuels et historiques en matière de performances et de disponibilité, ainsi que des alertes relatives aux événements de santé, adaptées uniquement à l'encombrement de votre application et à vos clients. La carte météo sur Internet vous donne une vue d'ensemble, tandis qu'un moniteur spécifique filtre les informations en fonction des mesures et des détails pertinents pour votre application. À l'aide d'un moniteur, vous pouvez également explorer les indicateurs historiques et obtenir des recommandations pour améliorer l'expérience client de votre application. Pour en savoir plus, veuillez consulter la section Commencer à utiliser Amazon CloudWatch Internet Monitor à l'aide de la console.

Comment fonctionne Amazon CloudWatch Internet Monitor

Cette section fournit des informations sur le fonctionnement CloudWatch d'Amazon Internet Monitor. Cela inclut des descriptions de la manière dont il AWS collecte les données qu'il utilise pour détecter les problèmes de connectivité sur Internet, et de la manière dont les scores de performance et de disponibilité sont calculés.

Table des matières

- Comment est créée la carte météo mondiale sur Internet (questions fréquemment posées)

Comment Internet Monitor se concentre uniquement sur l'empreinte du trafic de votre application

Internet Monitor concentre la surveillance uniquement sur le sous-ensemble d'Internet auquel accèdent les utilisateurs de vos AWS ressources, au lieu de surveiller globalement votre site Web dans toutes les régions du monde, comme le font d'autres outils. C'est également une solution économique, abordable pour les grandes et petites entreprises.

Internet Monitor utilise les mêmes sondes puissantes et algorithmes de détection de problèmes qui AWS tirent parti des fonctionnalités internes et vous avertissent des problèmes de connectivité affectant votre application en créant des événements de santé dans Internet Monitor. Le Moniteur Internet vous donne ensuite accès à la carte de performance et de disponibilité qui en résulte, en superposant le profil de trafic qu'il crée à partir de vos spectateurs actifs, sur la base des ressources de votre application.

À l'aide de ces informations, Moniteur Internet vous montre uniquement les événements pertinents (c'est-à-dire les événements provenant des endroits où vous avez des utilisateurs actifs) et l'impact de ces événements sur votre volume global d'utilisateurs. Ainsi, l'impact d'un événement, en pourcentage, est basé sur votre trafic total dans le monde entier.

Internet Monitor publie dans CloudWatch Logs les mesures Internet toutes les cinq minutes pour les 500 principaux réseaux urbains (sites clients et ASN, généralement des fournisseurs de services Internet ou des FAI) qui envoient du trafic vers chaque moniteur. Vous pouvez choisir de publier les mesures Internet pour tous les réseaux urbains surveillés (jusqu'à la limite de service de 500 000 réseaux urbains) dans un compartiment Amazon S3. Pour plus d’informations, consultez Publication de mesures Internet sur Amazon S3 dans Amazon CloudWatch Internet Monitor.

Les avantages du Moniteur Internet sont les suivants :

- L'utilisation du Moniteur Internet n'impose pas de charge ou de coût supplémentaire à votre application qui est hébergée par AWS.
- Vous n'avez pas besoin d'inclure un code de mesure des performances dans vos ressources côté client, ou dans votre application.
- Vous pouvez obtenir une visibilité des performances et de la disponibilité sur l'ensemble de l'Internet auquel votre application est connectée, y compris les informations sur le « dernier kilomètre ».

Notez que, dans la mesure où Internet Monitor crée des mesures en fonction de vos AWS ressources, Internet Monitor crée uniquement des événements spécifiques au trafic de votre application. Les problèmes d'Internet mondial en général ne sont pas signalés. En outre, lorsque l'emplacement du service est un Région AWS, les mesures et les événements émis sont conçus pour représenter la connectivité au niveau régional et ne représentent pas exactement la connectivité entre l'emplacement d'un utilisateur final et une zone de disponibilité.

Comment AWS mesurer les problèmes de connectivité et calculer les mesures

Amazon CloudWatch Internet Monitor utilise les données de connectivité Internet entre différents CloudFront points de présence (POP) Régions AWS et Amazon vers différents sites clients via des numéros de système autonomes (ASN), généralement des fournisseurs de services Internet (ISP). Il s'agit des données de connectivité qui sont utilisées en interne par AWS les opérateurs, au quotidien, pour détecter de manière proactive les problèmes de connectivité sur Internet mondial.

Pour chacune d'entre elles Région AWS, nous savons quelles parties d'Internet communiquent avec la Région et nous prenons les mesures suivantes :

- Nous surveillons activement ces parties d'Internet, avec une fenêtre continue de 30 jours.
- Nous utilisons à la fois des sondes de réseau et de protocole de niveau supérieur, y compris des sondes entrantes et sortantes.

AWS possède des sondes actives et passives qui mesurent la latence (performance) au 90e percentile et l'accessibilité (disponibilité) de chaque service à l' CloudFront ensemble Région AWS d'Internet. Les modèles anormaux de connectivité entre un service et l'emplacement d'un client sont surveillés, puis signalés sous forme d'alertes au client.

Calcul de la disponibilité et du RTT

Le temps d'aller-retour (RTT) est le temps qu'il faut pour qu'une demande de l'utilisateur renvoie une réponse à l'utilisateur. Lorsque le temps de propagation aller et retour est agrégé entre les emplacements des utilisateurs finaux, la valeur est pondérée par le montant de votre trafic qui est conduit par chaque emplacement d'utilisateur final.

Par exemple, avec deux emplacements d'utilisateurs finaux, l'un servant 90 % du trafic avec un RTT de 5 ms, et l'autre servant 10 % du trafic avec un RTT de 10 ms, le résultat est un RTT agrégé de 5,5 ms (qui provient de 5 ms * 0,9 + 10 ms * 0,1).

Notez qu'il existe des différences entre les ressources relatives à la mesure de la latence du dernier kilomètre. Pour les mesures de latence d'Internet Monitor, les VPC, les équilibreurs de charge réseau et les WorkSpaces annuaires n'incluent pas la latence du dernier kilomètre.

Calcul des scores de performance et de disponibilité

AWS dispose de données historiques substantielles sur les performances et la disponibilité d'Internet entre les AWS services et les différents réseaux urbains (sites et ASN). En appliquant une analyse statistique aux données, Moniteur Internet peut détecter quand les performances et la disponibilité de votre application ont baissé, par rapport à une référence estimée qu'il a calculée. Pour faciliter la détection de ces baisses, Moniteur Internet communique ces informations sous la forme de scores de santé : un score de performance et un score de disponibilité.

Les scores de l'état sont calculés à différentes granularités. À la granularité la plus fine, nous calculons le score de santé pour une région géographique, telle qu'une ville ou une zone métropolitaine, et un ASN (un réseau urbain). Nous regroupons également les scores de santé individuels en scores globaux de santé pour une application dans un moniteur. Si vous consultez les scores de performance ou de disponibilité sans filtrer pour une zone géographique ou un fournisseur de services spécifique, Moniteur Internet fournit des scores de santé globaux.

Les scores de santé globaux couvrent l'ensemble de votre application pour la période de temps spécifiée. Lorsque le score de performance ou de disponibilité des paires de réseaux urbains de votre application atteint le seuil d'événement de santé correspondant ou descend en dessous de celui-ci pour un seuil de performance ou de disponibilité, Moniteur Internet déclenche un événement de santé. Par défaut, le seuil est de 95 % pour la performance globale et la disponibilité. Moniteur Internet crée également des événements de santé basés sur des seuils locaux (si l'option est activée, comme c'est le cas par défaut) en fonction des valeurs que vous configurez. Pour en savoir plus sur la configuration des seuils d'événements de santé, consultez Modifier les seuils relatifs aux événements de santé.

Lorsque vous explorez les informations du moniteur et des fichiers journaux pour examiner les problèmes et en savoir plus, vous pouvez filtrer les données par ville (emplacement), par réseau (ASN ou fournisseurs de services Internet) ou les deux. Vous pouvez donc utiliser des filtres pour voir les scores de santé de différentes villes, ASN ou paires de réseaux urbains, selon les filtres que vous choisissez.

- Un score de disponibilité représente le pourcentage estimé du trafic qui ne subit pas de baisse de disponibilité. Le Moniteur Internet estime le pourcentage de trafic subissant

une baisse à partir du trafic total observé et des mesures métriques de disponibilité. Par exemple, un score de disponibilité de 99 % pour une paire d'utilisateurs finaux et d'emplacements de service équivaut à 1 % du trafic subissant une baisse de disponibilité pour cette paire.

- Un score de performance représente le pourcentage du trafic qui ne subit pas de baisse de performance. Par exemple, un score de performance de 99 % pour une paire d'utilisateurs finaux et d'emplacements de service équivaut à 1 % du trafic subissant une baisse de performance pour cette paire.

Calcul du TTFB et du RTT (latence)

Le délai jusqu'au premier octet (TTFB) fait référence au délai entre le moment où un client fait une demande et le moment où il reçoit le premier octet d'informations du serveur. AWS les calculs pour le TTFB mesurent le temps écoulé entre Amazon EC2 ou Amazon CloudFront et le nœud de mesure Internet Monitor (y compris le dernier kilomètre du nœud). En d'autres termes, Internet Monitor mesure le temps écoulé entre l'utilisateur et la région Amazon EC2 pour TTFB pour EC2, et entre l'utilisateur et le TTFB pour. CloudFront CloudFront

Pour le temps de propagation aller et retour (RTT), Moniteur Internet inclut le temps écoulé entre le réseau urbain (c'est-à-dire l'emplacement du client et l'ASN, généralement un fournisseur de services Internet), tel que mappé par l'adresse IP publique, jusqu'à la Région AWS. Cela signifie que Moniteur Internet n'a pas de visibilité sur le dernier kilomètre pour les utilisateurs qui accèdent à Internet via une passerelle ou un VPN.

Notez qu'il existe des différences entre les ressources relatives à la mesure de la latence du dernier kilomètre. Pour les mesures de latence d'Internet Monitor, les VPC, les équilibreurs de charge réseau et les WorkSpaces annuaires n'incluent pas la latence du dernier kilomètre.

Internet Monitor inclut des informations sur le TTFB moyen dans la section Suggestions d'optimisation du trafic de l'onglet Informations sur le trafic du tableau de CloudWatch bord, afin de vous aider à évaluer les options pour les différentes configurations de votre application susceptibles d'améliorer les performances.

Mesures et agrégation par région et par zone de disponibilité

Bien qu'Internet Monitor agrège les mesures et partage l'impact au niveau régional, il calcule l'impact au niveau de la zone de disponibilité (AZ). Cela signifie que, si, à la suite d'un événement, une seule zone est affectée et que la majeure partie de votre trafic passe par cette zone, vous constatez un impact sur votre trafic. Toutefois, pour le même événement, si le trafic

de votre application ne passe pas par une zone de disponibilité affectée, vous ne constaterez aucun impact.

Notez que cela ne s'applique qu'aux ressources qui ne sont pas WorkSpaces des annuaires. WorkSpaces les annuaires sont mesurés uniquement au niveau régional.

Précision de la géolocalisation dans Moniteur Internet

Pour les informations de localisation, Internet Monitor utilise les données de géolocalisation IP fournies par. MaxMind La précision des informations de localisation dans les mesures d'Internet Monitor dépend de la précision MaxMind des données.

Sachez que les mesures de `Metro` niveau peuvent ne pas être précises pour des sites situés en dehors des États-Unis.

Quand Moniteur Internet crée et résout des événements de santé

Moniteur Internet crée et clôture des événements de santé pour le trafic d'applications que vous surveillez en fonction des seuils actuels définis. Moniteur Internet possède une configuration de seuil par défaut, mais vous pouvez définir vos propres seuils. Moniteur Internet détermine l'impact global des problèmes de connectivité sur votre application, ainsi que l'impact sur les zones locales où votre application compte des clients, et crée des problèmes de santé lorsque les seuils sont franchis.

Internet Monitor calcule l'impact des problèmes de connectivité sur l'emplacement d'un client en fonction des données historiques relatives aux performances Internet et à la disponibilité du trafic réseau accessible au service via AWS celui-ci. Il applique les informations pertinentes à votre application, en fonction des emplacements géographiques des ASN et des services où les clients utilisent votre application : les paires de réseaux urbains concernées. Les emplacements sont déterminés à partir des ressources que vous ajoutez à votre moniteur. Moniteur Internet utilise ensuite une analyse statistique afin de déterminer si les performances et la disponibilité ont diminué, ce qui a une incidence sur l'expérience client de votre application.

Les scores de performance et de disponibilité calculés par Moniteur Internet sont représentés par le pourcentage du trafic qui ne subit pas de baisse. L'impact est le contraire de cela : c'est une représentation de l'importance du problème pour les utilisateurs finaux du client. Ainsi, s'il y a une baisse de disponibilité mondiale de 93 %, par exemple, l'impact correspondant sera de 7 %.

Lorsque le score de performance ou de disponibilité des paires de réseaux urbains de votre application atteint globalement le seuil d'événement de santé correspondant ou descend en dessous de celui-ci pour un seuil de performance ou de disponibilité, Moniteur Internet déclenche

un événement de santé. Par défaut, le seuil est de 95 % pour la performance et la disponibilité. Les valeurs pour atteindre le seuil ou descendre en dessous sont cumulatives, ce qui signifie que plusieurs événements de moindre envergure peuvent se combiner pour atteindre le pourcentage de seuil ou qu'un seul événement peut atteindre le seuil ou descendre en dessous.

Tant que les scores de performance ou de disponibilité qui ont déclenché l'événement sont égaux ou inférieurs au seuil d'impact global correspondant à l'événement de santé, l'événement de santé reste actif. Lorsque le score ou les scores combinés qui ont déclenché l'événement dépassent le seuil, Moniteur Internet résout le problème de santé.

Moniteur Internet crée également des événements de santé en fonction des seuils locaux et du pourcentage du trafic global sur lequel un problème a un impact. Vous pouvez configurer des options pour les seuils locaux ou désactiver complètement les seuils locaux.

Pour en savoir plus sur la configuration des seuils d'événements de santé, consultez Modifier les seuils relatifs aux événements de santé.

Calendrier des rapports d'événements de santé

Le Moniteur Internet utilise un agrégateur pour rassembler tous les signaux sur les problèmes Internet, afin de créer des événements de l'état dans les moniteurs en quelques minutes.

Dans la mesure du possible, Internet Monitor analyse l'origine d'un problème de santé afin de déterminer s'il a été causé par un ASN AWS ou par un ASN. L'analyse d'événement de santé se poursuit une fois que l'événement est résolu. Moniteur Internet peut mettre à jour les événements avec de nouvelles informations pendant une heure au maximum.

Comment Moniteur Internet fonctionne avec le trafic IPv4 et IPv6

Moniteur Internet mesure la santé d'un réseau uniquement via IPv4 et affiche les événements de santé, ainsi que les métriques de disponibilité et de performance, si vous acheminez du trafic vers ce réseau via n'importe quelle famille d'adresses IP (IPv4 ou IPv6). Si vous gérez du trafic à partir d'une ressource à double pile, telle qu'une CloudFront distribution à double pile, Internet Monitor signale un événement de santé et indique une baisse du score de performance ou de disponibilité uniquement si le trafic IPv4 présente les mêmes problèmes pour la ressource que le trafic IPv6.

Notez que les métriques de Moniteur Internet relatives au nombre total d'octets entrants et d'octets sortants reflètent précisément l'ensemble du trafic Internet (IPv4 et IPv6).

Comment Internet Monitor sélectionne le sous-ensemble de réseaux urbains à inclure

Lorsque vous définissez une limite maximale pour le nombre de réseaux urbains surveillés par votre moniteur ou que vous choisissez un pourcentage du trafic à surveiller, Internet Monitor

choisit les réseaux urbains à inclure (surveiller) en fonction du volume de trafic récent le plus élevé.

Par exemple, si vous définissez une limite maximale de 100 réseaux urbains, Internet Monitor surveille (jusqu'à) 100 réseaux urbains en fonction du trafic de votre application au cours d'une heure récente. Plus précisément, Internet Monitor surveille les 100 principaux réseaux urbains ayant enregistré le plus de trafic au cours de la dernière fenêtre d'une heure avant la dernière fenêtre d'une heure.

Pour illustrer cela, disons que l'heure actuelle est 14 h 30. Dans ce scénario, le trafic que vous voyez sur votre écran a été capturé entre 13 h 00 et 14 h 00, et la mesure du volume de trafic utilisée par Internet Monitor pour déterminer les 100 principaux réseaux urbains a été capturée entre 12 h 00 et 13 h 00.

Comment est créée la carte météo mondiale sur Internet (questions fréquemment posées)

La carte météo CloudWatch Internet d'Amazon Internet Monitor est disponible sur la console Internet Monitor pour tous les AWS clients authentifiés. Cette section contient des détails sur la façon dont la carte météo sur Internet est créée et comment l'utiliser.

Qu'est-ce que la carte météo Internet d'Internet Monitor ?

La carte météo d'Internet fournit une représentation visuelle des problèmes liés à Internet dans le monde entier. Il met en évidence les sites des clients concernés, c'est-à-dire les villes et les ASN (généralement des fournisseurs de services Internet). La carte montre une combinaison de problèmes de disponibilité et de performance qui ont récemment eu un impact sur l'expérience Internet des clients pour les principaux sites et AWS services clients du monde entier.

D'où proviennent les données de la carte ?

Les données sont basées sur une combinaison d'enquêtes actives et passives sur Internet. Pour en savoir plus sur la façon dont Internet Monitor mesure les données, vous pouvez consulter la section Comment AWS mesure les problèmes de connectivité.

À quelle fréquence la carte est-elle mise à jour ?

La carte météo sur Internet est mise à jour toutes les 15 minutes.

Quels réseaux sont suivis pour détecter les pannes ?

AWS suit les réseaux du monde entier qui représentent les préfixes IP importants utilisés par les clients pour établir des connexions Internet. AWS Nous évaluons les pannes sur les sites clients les plus parlants en termes de volume de trafic envoyé et reçu depuis le AWS réseau.

Qu'est-ce qui détermine si un événement Internet est inclus sur la carte ?

Voici quelques critères de haut niveau que nous utilisons pour déterminer si un événement Internet est inclus sur la carte météo Internet :

- AWS détecte un événement de disponibilité ou de performance.
- Si l'événement est de courte durée, par exemple s'il dure moins de 5 minutes, nous l'ignorons.
- Ensuite, si l'événement se produit dans un site client classé parmi les meilleurs orateurs, il est considéré comme une panne.

Quels sont les seuils utilisés pour la carte météo sur Internet ?

Les seuils permettant de déterminer les pannes ne sont pas statiques sur la carte météo d'Internet. Internet Monitor détermine ce qui constitue un événement en se basant sur la détection d'un écart par rapport aux valeurs attendues. Pour en savoir plus sur son fonctionnement, consultez la façon dont Internet Monitor détermine à quel moment créer des événements de santé pour les moniteurs que vous créez avec le service. Lorsque vous créez un moniteur, Internet Monitor génère des mesures de santé du trafic Internet spécifiques au trafic de votre propre application. Internet Monitor vous alerte également en cas de problèmes de santé affectant le trafic Internet de votre application.

Que puis-je faire avec ces données ?

La carte météo Internet fournit un résumé rapide des principaux événements Internet survenus dans le monde au cours des dernières 24 heures. Il vous permet de vous faire une idée de l'expérience de surveillance Internet, sans avoir à intégrer votre propre trafic Internet à Internet Monitor. Pour exploiter tout le potentiel des fonctionnalités de surveillance Internet AWS et les personnaliser en fonction de vos applications et services hébergés sur Internet AWS, vous pouvez créer un moniteur dans Internet Monitor.

Lorsque vous créez un moniteur, vous activez Internet Monitor pour identifier les chemins Internet spécifiques qui affectent les clients de votre application, et vous avez accès aux fonctionnalités qui peuvent vous aider à améliorer votre expérience client. Vous serez également informé de manière proactive des nouveaux problèmes Internet qui ont un impact spécifique sur le trafic de votre application et sur vos clients.

Comment puis-je obtenir plus de détails sur les événements ?

Cliquez sur une panne sur la carte pour voir les détails, notamment les dates de début et de fin d'un événement, la ville et l'ASN concernés, ainsi que le type de problème (c'est-à-dire un problème de performance ou un problème de disponibilité).

Pour obtenir des informations plus détaillées sur les événements et pour obtenir des mesures personnalisées du trafic de votre application, créez un moniteur dans Internet Monitor.

Exemples de cas d'utilisation d'Amazon CloudWatch Internet Monitor

Dans cette section, nous décrivons plusieurs exemples spécifiques, avec des liens vers des articles de blog contenant plus de détails. Ces exemples montrent comment vous pouvez utiliser les fonctionnalités d'Amazon CloudWatch Internet Monitor pour vous aider à surveiller votre application et à améliorer l'expérience de vos utilisateurs.

Configurer des alertes et déterminer les actions à mettre en œuvre

Vous pouvez utiliser Moniteur Internet pour obtenir des informations sur les métriques de performance Internet moyenne au fil du temps et sur les événements de santé par réseau urbain (emplacement client et ASN, généralement un fournisseur de services Internet). À l'aide d'Internet Monitor, vous pouvez identifier les événements qui ont un impact sur l'expérience de l'utilisateur final pour les applications hébergées sur Amazon Virtual Private Clouds (VPC), les Network Load Balancers, Amazon WorkSpaces ou Amazon. CloudFront

Une fois que vous avez créé un moniteur, plusieurs options s'offrent à vous pour être alerté en cas d'événements de santé Moniteur Internet. Il s'agit notamment de notifications basées sur des CloudWatch alarmes utilisant des indicateurs d'événements ou EventBridge des règles Amazon pour filtrer les événements médicaux. Vous pouvez choisir différentes options pour les notifications ou les actions en fonction des alarmes, y compris, par exemple, les AWS SMS notifications ou les mises à jour d'un groupe de CloudWatch journaux.

Pour voir un exemple avec des instructions détaillées, consultez le billet de blog suivant : Présentation d'Amazon CloudWatch Internet Monitor.

Identifier les problèmes de latence et améliorer le TTFB pour améliorer l'expérience de jeu multijoueur

Utilisez Moniteur Internet pour identifier rapidement les endroits où les joueurs rencontrent des problèmes de latence à travers le monde dans les applications de jeu cloud et fournir des informations sur l'amélioration des performances. En identifiant les endroits où le plus grand nombre de joueurs ont actuellement le temps au premier octet (TTFB) le plus lent, vous savez comment améliorer la latence afin de satisfaire l'ensemble de vos joueurs.

Maintenant, lorsque vous êtes prêt à déployer le prochain serveur EC2 pour votre jeu, choisissez Région AWS celui qui, selon Internet Monitor, réduira le TTFB dans la zone où la latence est élevée et où le nombre de joueurs est important.

Pour en savoir plus sur la configuration et l'utilisation d'Internet Monitor dans ce cas d'utilisation, consultez le billet de blog suivant : Utiliser Amazon CloudWatch Internet Monitor pour une meilleure expérience de jeu.

Observabilité entre comptes Internet Monitor

Grâce à l'observabilité entre comptes d'Internet Monitor, vous pouvez surveiller vos applications qui couvrent plusieurs AWS comptes au sein d'un même compte. Région AWS

Vous pouvez utiliser Amazon CloudWatch Observability Access Manager pour configurer un ou plusieurs de vos AWS comptes en tant que compte de surveillance. Vous allez permettre au compte de surveillance de consulter les données de votre compte source en créant un récepteur dans votre compte de surveillance. Un récepteur est une ressource qui représente un point d'attache dans un compte de surveillance. Pour Internet Monitor, le point d'attache de la ressource est un moniteur. Vous utilisez le récepteur pour créer un lien entre votre compte source et votre compte de surveillance. Pour plus d’informations, consultez CloudWatch observabilité entre comptes.

Ressources requises

Pour garantir le bon fonctionnement de CloudWatch l'observabilité entre comptes Application Insights, assurez-vous que les types de télémétrie suivants sont partagés via l' CloudWatch Observability Access Manager.

- Moniteurs dans Internet Monitor
- Métriques sur Amazon CloudWatch
- Groupes de journaux dans Amazon CloudWatch Logs

Commencer à utiliser Amazon CloudWatch Internet Monitor à l'aide de la console

Pour commencer à utiliser Amazon CloudWatch Internet Monitor, vous devez créer un moniteur dans Internet Monitor pour votre application en ajoutant les AWS ressources qu'elle utilise et en définissant plusieurs options de configuration. Ce chapitre décrit la procédure pour ajouter un moniteur dans

la console. Il comprend également une section avec plus de détails sur les ressources de Monitor Internet, puis des sections supplémentaires avec les descriptions et les limites des différentes options que vous pouvez ou devez configurer pour votre moniteur.

Table des matières

Création d'un moniteur dans Amazon CloudWatch Internet Monitor à l'aide de la console

Vous créez un moniteur dans Amazon CloudWatch Internet Monitor pour votre application en ajoutant les AWS ressources qu'elle utilise, puis en définissant plusieurs options de configuration. Les ressources que vous ajoutez, qu'il s'agisse d'Amazon Virtual Private Clouds (VPC), d'équilibreurs de charge réseau (NLB), de CloudFront distributions ou d' WorkSpaces annuaires, fournissent les informations permettant à Internet Monitor de cartographier les informations relatives au trafic Internet de votre application. Après avoir créé votre moniteur, attendez 15 à 30 minutes pour générer le profil de trafic spécifique à l'endroit où votre application est utilisée. Vous pouvez ensuite utiliser le moniteur Internet Monitor ou d'autres outils pour visualiser et explorer les performances et la disponibilité liées à l'utilisation de vos clients. Ces outils vous fournissent des informations à l'aide des mesures du trafic de votre application, collectées et publiées par le moniteur, par exemple dans CloudWatch Logs.

Généralement, il est plus simple de créer un moniteur dans Moniteur Internet pour une seule application. Sur le même moniteur, vous pouvez rechercher et trier les mesures et les métriques dans les fichiers journaux Moniteur Internet en fonction de différents emplacements et ASN (généralement des fournisseurs de services Internet), ou d'autres informations. Il n'est pas nécessaire de créer des moniteurs distincts pour les applications dans différents domaines, par exemple.

Les étapes décrites ici vous guident dans la configuration à l'aide de la console de votre moniteur. Pour voir des exemples d'utilisation des actions de l'API AWS Command Line Interface avec Internet Monitor, pour créer un moniteur, afficher des événements, etc., consultezExemples d'utilisation de la CLI avec Amazon CloudWatch Internet Monitor.

Pour créer un moniteur 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 de gauche, sous Surveillance du réseau, choisissez Internet Monitor.
3. Choisissez Create monitor (Créer un contrôle).
4. Pour Monitor name (Nom du moniteur), saisissez le nom que vous voulez utiliser pour ce moniteur dans Moniteur Internet.
5. Choisissez Add resources (Ajouter des ressources), puis sélectionnez les ressources pour définir les limites de surveillance à utiliser par Moniteur Internet pour ce moniteur.

   > **Note**
   >
   > Tenez compte des points suivants :
   >
   > - Pour générer des résultats significatifs avec Internet Monitor, les VPC que vous ajoutez doivent être connectés à Internet en configurant une passerelle Internet.
   > - Vous pouvez ajouter une combinaison de VPC et de CloudFront distributions, ou vous pouvez ajouter des WorkSpaces répertoires, ou vous pouvez ajouter des équilibreurs de charge réseau. Vous ne pouvez pas ajouter d'équilibreurs de charge réseau ou de WorkSpaces répertoires avec d'autres types de ressources.

6. Choisissez un pourcentage de votre trafic Internet à surveiller.
7. Spécifiez éventuellement des options supplémentaires sous Paramètres avancés.

   - Pour Nombre maximum de réseaux urbains, vous pouvez sélectionner une limite de nombre de réseaux urbains (emplacements et ASN, ou fournisseurs de services Internet) pour lesquels le Moniteur Internet surveillera le trafic. Vous pouvez la modifier à tout moment via votre moniteur. veuillez consulter Choisir une limite maximale de réseaux urbains.

     Pour rétablir les valeurs par défaut, saisissez `500000`.

Si vous définissez une limite maximale de réseaux urbains, cela limite le nombre de réseaux urbains surveillés par Moniteur Internet pour votre application, quel que soit le pourcentage de trafic que vous choisissez de surveiller.

- Vous pouvez spécifier un nom de compartiment Amazon S3 et un préfixe personnalisé pour publier les mesures Internet sur Amazon S3 pour tous les réseaux urbains surveillés.

  Internet Monitor publie les 500 meilleures mesures Internet (en termes de volume de trafic) pour votre application dans CloudWatch Logs toutes les cinq minutes. Si vous choisissez de publier des mesures dans S3, les mesures sont toujours publiées dans CloudWatch Logs. Pour plus d'informations, consultez Publication de mesures Internet sur Amazon S3 dans Amazon CloudWatch Internet Monitor.

- En option, vous pouvez ajouter une balise pour votre moniteur.

8. Choisissez Create monitor (Créer un contrôle).

Après avoir créé un moniteur, vous pouvez le modifier à tout moment, par exemple pour changer le pourcentage de trafic des applications, mettre à jour la limite maximale de réseaux urbains ou ajouter ou supprimer des ressources. Vous pouvez également supprimer le moniteur. Pour effectuer ces tâches, dans la console Moniteur Internet, sélectionnez un moniteur, puis choisissez une option dans le menu Action. Notez que vous ne pouvez pas modifier le nom d'un moniteur.

Pour afficher le tableau de bord du Moniteur Internet

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Surveillance du réseau, puis Moniteur Internet.

   L'onglet Monitors (Moniteurs) affiche une liste des moniteurs que vous avez créés.

Pour voir plus d'informations sur un moniteur spécifique, sélectionnez un moniteur.

Ajouter des ressources à votre moniteur

Lorsque vous créez un moniteur, vous devez y associer les ressources de votre application : Amazon Virtual Private Clouds (VPC), Network Load Balancers, Amazon CloudFront distributions, Network Load Balancers (NLB) ou annuaires Amazon. WorkSpaces Internet Monitor sait ensuite où se trouvent le trafic Internet et les clients de votre application, et il peut créer et gérer un profil de trafic qui détermine les mesures pertinentes à publier pour votre moniteur.

Vous pouvez ajouter les types de ressources suivants à un moniteur dans Internet Monitor en tant que « ressources surveillées ». Notez qu'Internet Monitor ne prend pas en charge l'ajout de différents types de ressources dans un seul moniteur.

- Des VPC : chaque VPC que vous ajoutez dans une région est une ressource surveillée. Lorsque vous ajoutez un VPC, Internet Monitor surveille le trafic de toute application connectée à Internet dans le VPC, par exemple une application hébergée sur une instance Amazon EC2, derrière un Network Load Balancer ou dans un conteneur. AWS Fargate
- Des Network Load Balancers : chaque Network Load Balancer que vous ajoutez est une ressource surveillée.
- CloudFront distributions : chaque CloudFront distribution que vous ajoutez est une ressource surveillée.
- WorkSpaces répertoires : chaque WorkSpaces répertoire que vous ajoutez dans une région est une ressource surveillée.

Lorsque vous surveillez le trafic de VPC, le trafic des applications hébergées sur des équilibreurs de charge situés derrière le VPC est surveillé. Vous pouvez choisir de surveiller le trafic pour des équilibreurs de charge Network Load Balancer individuels au lieu de surveiller un VPC doté de plusieurs équilibreurs de charge. Cela peut être utile, par exemple, si vous devez comprendre et configurer des fonctionnalités pour améliorer la performance ou l'efficacité au niveau de l'équilibreur de charge. Vous pouvez également avoir besoin d'informations de conformité au niveau du Network Load Balancer.

Lorsque vous ajoutez des ressources à un moniteur dans Moniteur Internet, vous devez tenir compte de ce qui suit :

- Pour générer des résultats significatifs avec Internet Monitor, les VPC que vous ajoutez doivent être connectés à Internet en configurant une passerelle Internet.
- Internet Monitor ne prend pas en charge l'ajout de différents types de ressources dans un seul moniteur.

Lorsque vous ajoutez des VPC ou des NLB en tant que ressources, il existe des différences régionales que vous devez garder à l'esprit lorsque vous ajoutez des VPC ou des NLB. Pour plus d’informations, consultez Compatible Régions AWS avec Amazon CloudWatch Internet Monitor.

Il existe également des différences entre les ressources relatives à la mesure de la latence du dernier kilomètre. Pour les mesures de latence d'Internet Monitor, les VPC, les NLB et les WorkSpaces annuaires n'incluent pas la latence du dernier kilomètre.

Choisir le pourcentage de trafic d'application à surveiller

La couverture que vous choisissez pour le pourcentage du trafic d'application à surveiller détermine le nombre de réseaux urbains (emplacements clients et ASN, généralement des fournisseurs de services Internet) surveillés pour votre application, jusqu'à une limite maximale facultative de réseaux urbains que vous pouvez également définir.

Si vous choisissez de surveiller moins de 100 % du trafic de votre application, il se peut qu'il existe un écart d'observabilité avec votre moniteur. En effet, si Amazon CloudWatch Internet Monitor crée des problèmes de santé au cours desquels vous ne surveillez pas le trafic, vous n'en serez pas conscient. Vous pouvez également avoir une couverture moins élevée pour les informations de performance et de disponibilité relatives à l'accès des clients à votre application.

Les sections suivantes décrivent les options permettant d'explorer la couverture et les paramètres du pourcentage de trafic, et de se faire une idée de l'impact d'une augmentation ou d'une diminution de la couverture.

- Étude de la possibilité de modifier le pourcentage de trafic de votre application
- Affichage du nombre de réseaux urbains surveillés en fonction de différents pourcentages de trafic

Étude de la possibilité de modifier le pourcentage de trafic de votre application

Vous pouvez tester les différentes valeurs possibles de pourcentage de trafic d'application en consultant le nombre de réseaux urbains surveillés lorsque vous modifiez le pourcentage. La procédure décrite dans cette section fournit des step-by-step informations.

Dans la console Moniteur Internet, vous pouvez essayer d'augmenter ou de diminuer le pourcentage de trafic d'application pour votre moniteur, et voir le nombre estimé de réseaux urbains qui seraient ainsi couverts. Avec cette option, vous pouvez rapidement voir comment la modification de votre pourcentage de trafic affecte le nombre de moniteurs urbains surveillés. Cela peut vous aider à avoir une idée du pourcentage de trafic d'application à choisir pour votre application.

Tester la couverture de surveillance en augmentant et en diminuant le pourcentage de trafic d'application

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le volet de navigation de gauche, sous Surveillance du réseau, choisissez Internet Monitor.
3. Dans votre liste de moniteurs, choisissez un moniteur.
4. Dans l'onglet Aperçu, dans la section Trafic surveillé, choisissez le graphique en pourcentage, puis choisissez Mettre à jour la couverture de surveillance.
5. Dans la boîte de dialogue Explorez et définissez la couverture de surveillance du trafic, cliquez sur les flèches pour augmenter ou diminuer le pourcentage de trafic à surveiller. En choisissant 100 % de trafic, vous pouvez voir combien de réseaux urbains sont surveillés avec une couverture complète pour surveiller votre application.
6. Pour en savoir plus sur la façon dont le nombre de réseaux urbains surveillés (estimé ici) peut affecter vos coûts, cliquez sur le lien vers le calculateur de CloudWatch prix, puis faites défiler la page vers le bas jusqu'à Internet Monitor.
7. Pour définir un nouveau pourcentage de trafic à surveiller, choisissez Mettre à jour la couverture du moniteur. Sinon, pour conserver le niveau de couverture actuel, choisissez Annuler.

Affichage du nombre de réseaux urbains surveillés en fonction de différents pourcentages de trafic

Vous pouvez voir le nombre de réseaux urbains qui seraient surveillés pour votre application selon différents pourcentages de trafic d'application. La procédure décrite dans cette section fournit des step-by-step informations.

Dans la console Moniteur Internet, vous pouvez consulter des graphiques illustrant l'évolution de la couverture de vos réseaux urbains en fonction des pourcentages de trafic d'application, sur un intervalle de temps que vous spécifiez. Il s'agit d'un moyen rapide de visualiser et de comparer la couverture de surveillance de votre application à des pourcentages de trafic spécifiques, le tout sur un seul graphique.

Pour afficher des graphiques du pourcentage de trafic d'application et de la couverture correspondante des réseaux urbains

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, sous Surveillance du réseau, choisissez Internet Monitor.
3. Dans votre liste de moniteurs, choisissez un moniteur.
4. Choisissez l'onglet Informations sur le trafic et faites défiler la page vers le bas jusqu'à Graphiques du trafic Internet.
5. Sous Comparer les options de couverture du trafic, dans la liste déroulante, sélectionnez un ou plusieurs pourcentages. Vous pouvez choisir un ou plusieurs pourcentages de trafic

d'application, et le graphique du Nombre total de réseaux urbains surveillés est mis à jour pour afficher la couverture de surveillance fournie par Moniteur Internet pour ce pourcentage de trafic. En choisissant des réseaux urbains à 100 % de trafic, vous pouvez voir combien de réseaux urbains sont surveillés avec une couverture complète pour surveiller votre application.

Gardez à l'esprit les points suivants :

- La couverture du trafic est calculée en fonction du nombre de réseaux urbains au cours de l'heure précédente de trafic de votre application. Cela signifie qu'une fois que vous avez choisi un pourcentage spécifique du trafic à surveiller, il est possible que le nombre de réseaux urbains surveillés pour votre application soit inférieur à ce qui est indiqué ici dans un graphique de comparaison de la couverture du trafic.
- Pour vous assurer que tout le trafic de votre application est surveillé, définissez `TrafficPercentageToMonitor` à 100 et n'affectez aucune valeur à `MaxCityNetworksToMonitor`. Vous pouvez également définir `MaxCityNetworksToMonitor` à 500 000, qui correspond à la limite supérieure dans le Moniteur Internet.
- Si vous définissez une limite maximale de réseaux urbains, le nombre total de réseaux urbains surveillés ne dépasse jamais cette limite, quelle que soit l'option de pourcentage de trafic d'application que vous sélectionnez.
- Vous pouvez en savoir plus sur la façon dont le nombre de réseaux urbains surveillés peut affecter vos coûts. Sur la page Calculateur de prix, faites défiler CloudWatch la page vers le bas jusqu'à Internet Monitor.

Pour définir un nouveau pourcentage de trafic à surveiller, sous Découvrez d'autres options de couverture du trafic, sélectionnez Mettre à jour la couverture de surveillance. Dans la boîte de dialogue, choisissez un pourcentage de trafic, puis sélectionnez Mettre à jour la couverture du moniteur.

Choisir une limite maximale de réseaux urbains

Amazon CloudWatch Internet Monitor peut surveiller le trafic de vos applications pour certains ou tous les emplacements où les clients accèdent aux ressources de vos applications, ainsi que tous les ASN (généralement des fournisseurs de services Internet) par lesquels ils accèdent à votre application, c'est-à-dire les réseaux urbains pour le trafic Internet de vos applications. Vous choisissez un pourcentage de trafic d'application à surveiller lorsque vous créez votre moniteur, que vous pouvez mettre à jour à tout moment en modifiant le moniteur.

En plus de définir un pourcentage de trafic, vous pouvez également définir une limite maximale de nombre de réseaux urbains surveillés. Cette section décrit comment la limite de réseaux urbains peut vous aider à gérer les coûts de facturation et fournit des informations et un exemple pour vous aider à déterminer une limite.

La limite maximale que vous définissez pour le nombre de réseaux urbains permet de garantir la prévisibilité de votre facture. Pour plus d'informations, consultez Amazon CloudWatch Pricing. Vous pouvez également découvrir comment les différentes valeurs du nombre de réseaux urbains réellement surveillés peuvent affecter votre facture en utilisant le calculateur de CloudWatch prix. Pour explorer les options, sur la page Calculateur de prix, faites défiler CloudWatch la page vers le bas jusqu'à Internet Monitor.

Pour mettre à jour votre moniteur et modifier la limite maximale de réseaux urbains, consultez Modifier ou supprimer un moniteur Moniteur Internet.

Comment fonctionne la facturation avec les limites maximales de réseaux urbains

Définir une limite maximale de réseaux urbains surveillés peut vous éviter des frais imprévus sur votre facture. Cela est utile, par exemple, si vos modèles de trafic varient considérablement. Les coûts de facturation augmentent pour chaque réseau urbain surveillé après les 100 premiers réseaux urbains qui sont inclus (pour tous les moniteurs par compte). Si vous définissez une limite maximale de réseaux urbains, cela limite le nombre de réseaux urbains surveillés par Moniteur Internet pour votre application, quel que soit le pourcentage de trafic que vous choisissez de surveiller.

Vous ne payez que pour le nombre de réseaux urbains réellement surveillés. La limite maximale de réseaux urbains permet de fixer un plafond sur le total qui peut être inclus lorsque Moniteur Internet surveille le trafic avec votre moniteur. Vous pouvez modifier la limite maximale à tout moment via votre moniteur.

Pour explorer les options, sur la page Calculateur de prix, faites défiler CloudWatch la page vers le bas jusqu'à Internet Monitor. Pour plus d'informations sur la tarification d'Internet Monitor, consultez la section Internet Monitor sur la page de CloudWatch tarification d'Amazon.

Comment choisir une limite maximale de réseaux urbains

Pour vous aider à déterminer une limite maximale de réseaux urbains, identifiez le volume de trafic que vous souhaitez surveiller pour votre application. Les métriques Moniteur Internet suivantes peuvent vous aider à analyser l'utilisation et la couverture de votre trafic une fois que vous avez créé votre moniteur : `CityNetworksMonitored`, `TrafficMonitoredPercent` et une ou plusieurs métriques `CityNetworksFor`*`NN`*`PercentTraffic`, où *`NN`* est un pourcentage correspondant

à l'une des valeurs suivantes : 25, 50, 90, 95, 99 ou 100. Pour consulter les définitions de ces métriques et de toutes les autres métriques de Moniteur Internet, consultez Utilisation CloudWatch des métriques avec Amazon CloudWatch Internet Monitor.

Pour voir un graphique récapitulatif de la couverture de votre trafic Internet, accédez à l'onglet Informations sur le trafic du tableau de CloudWatch bord et, dans la section Graphiques du trafic Internet, choisissez une option pour Comparer les options de couverture du trafic. Le graphique de cette section indique le nombre de réseaux urbains surveillés pour votre application et inclut des courbes représentant les différents pourcentages de trafic d'application sélectionnés dans la liste déroulante. Pour en savoir plus, consultez la section Configuration du pourcentage de trafic d'application.

Pour explorer vos options plus en détail, vous pouvez utiliser les métriques de Moniteur Internet, comme décrit dans les exemples suivants. Ces exemples montrent comment sélectionner la limite maximale de réseaux urbains qui vous convient le mieux en fonction de la portée souhaitée de couverture du trafic Internet d'application. L'utilisation des requêtes relatives aux métriques d'Internet Monitor dans CloudWatch Metrics peut vous aider à mieux comprendre la couverture du trafic Internet de votre application.

Exemple de détermination de limite maximale de réseaux urbains

Supposons que vous ayez défini une limite maximale de surveillance de 100 réseaux urbains et que les clients accèdent à votre application sur 2 637 réseaux urbains. Dans CloudWatch Metrics, vous verrez les métriques Internet Monitor suivantes renvoyées :

```
CityNetworksMonitored 100
TrafficMonitoredPercent  12.5
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

À partir de cet exemple, vous pouvez voir que vous surveillez actuellement 12,5 % de votre trafic Internet, la limite maximale étant fixée à 100 réseaux urbains. Si vous souhaitez surveiller 90 % de votre trafic, la métrique suivante fournit des informations à ce sujet : `CityNetworksFor90PercentTraffic` indique que vous devez surveiller 2 143 réseaux urbains pour une couverture de 90 %. Pour ce faire, vous devez mettre à jour votre moniteur et définir la limite maximale de réseaux urbains à 2 143.

De même, supposons que vous souhaitiez surveiller à 100 % le trafic Internet de votre application. La métrique suivante, `CityNetworksFor100PercentTraffic`, indique que vous devez mettre à jour votre moniteur pour définir la limite maximale de réseaux urbains à 2 637.

Maintenant, si vous définissez le maximum à 5 000 réseaux urbains, étant donné que ce chiffre est supérieur à 2 637, les métriques suivantes sont renvoyées :

```
CityNetworksMonitored 2637
TrafficMonitoredPercent  100
CityNetworksFor90PercentTraffic  2143
CityNetworksFor100PercentTraffic  2637
```

À partir de ces métriques, vous pouvez constater qu'avec la limite la plus élevée, vous surveillez les 2 637 réseaux urbains, soit 100 % de votre trafic Internet.

Publication de mesures Internet sur Amazon S3 dans Amazon CloudWatch Internet Monitor

Vous pouvez choisir de demander à Amazon CloudWatch Internet Monitor de publier sur Amazon S3 les mesures de votre trafic Internet vers les réseaux urbains surveillés (sites clients et ASN, généralement des fournisseurs de services Internet) sur votre moniteur, jusqu'à la limite de service des 500 000 réseaux urbains. Internet Monitor publie automatiquement les mesures Internet dans CloudWatch Logs toutes les cinq minutes pour les 500 principaux réseaux urbains (en termes de volume de trafic) pour chaque moniteur. Les mesures publiées sur S3 incluent les 500 principales mesures publiées sur CloudWatch Logs.

Vous pouvez choisir l'option de publication sur S3 et spécifier le compartiment dans lequel publier les mesures lorsque vous créez ou mettez à jour votre moniteur. Le compartiment doit déjà être créé dans S3 pour que vous puissiez le spécifier dans Moniteur Internet. Il existe une limite de service de 500 000 réseaux urbains pour les mesures Internet publiées sur S3. Moniteur Internet publie les mesures Internet dans S3 sous forme d'événements, une série d'objets fichiers journaux compressés qui sont stockés dans le compartiment.

Lorsque vous créez le compartiment S3 dans lequel Internet Monitor doit publier les mesures, assurez-vous de suivre les instructions relatives aux autorisations fournies par CloudWatch Logs. Cela garantit qu'Internet Monitor peut publier les journaux directement sur S3 et, si nécessaire, créer et modifier les politiques de ressources associées au groupe de journaux recevant les journaux. AWS Pour plus d'informations, consultez la section Logs envoyés à CloudWatch Logs dans le guide de l'utilisateur Amazon CloudWatch Logs.

Les fichiers journaux publiés sont compressés. Si vous ouvrez les fichiers journaux à l'aide de la console Amazon S3, ils sont décompressés et les événements de mesure Internet s'affichent. Si vous téléchargez les fichiers, vous devez les décompresser pour afficher les événements.

Vous pouvez également interroger les mesures Internet dans les fichiers journaux à l'aide d'Amazon Athena. Amazon Athena est un service de requête interactif qui facilite l'analyse des données dans Amazon S3 à l'aide du langage SQL standard. Pour plus d'informations, consultez Utiliser Amazon Athena pour interroger les mesures Internet dans les fichiers journaux Amazon S3.

Utiliser un moniteur Moniteur Internet

Il existe plusieurs manières d'utiliser un moniteur Amazon CloudWatch Internet Monitor après l'avoir créé : par exemple, vous pouvez consulter des informations dans le CloudWatch tableau de bord, obtenir des informations à l'aide du AWS Command Line Interface et définir des alertes de santé.

Votre moniteur fournit des informations sur votre application et vos préférences de configuration afin que Moniteur Internet puisse personnaliser les métriques et les publier dans des événements pour vous. Moniteur Internet collecte les mesures à travers l'empreinte de l'infrastructure mondiale pour AWS. Ces mesures représentent une quantité considérable d'informations provenant du monde entier sur la performance et la disponibilité réseau. En utilisant les informations provenant des ressources que vous ajoutez à votre application, Moniteur Internet publie pour vous des mesures de performance et de disponibilité qui s'appliquent aux réseaux urbains (c'est-à-dire les emplacements clients et les ASN, généralement des fournisseurs de services Internet, ou FSI) sur lesquels votre application est active. Ainsi, les mesures figurant dans le tableau de bord Internet Monitor et dans les CloudWatch journaux (concernant la disponibilité, les performances, les octets surveillés transférés et le temps d'aller-retour) sont spécifiques aux sites de vos clients et à leurs ASN.

Le Moniteur Internet détermine également les anomalies de performance et de disponibilité. Par défaut, Internet Monitor superpose votre trafic aux mesures de disponibilité et de performance collectées pour chaque paire source-destination dans vos sites clients, afin de déterminer les baisses notables de performances ou de disponibilité. AWS En cas de dégradation significative des emplacements et de la portée de votre application, Moniteur Internet génère un événement de santé et publie des informations sur le problème sur votre moniteur.

Après avoir créé un moniteur, vous pouvez l'utiliser pour accéder ou être alerté des informations fournies par Moniteur Internet, des manières suivantes :

- Utilisez le CloudWatch tableau de bord pour consulter et explorer les performances, la disponibilité et les événements liés à l'état de santé, explorer les données historiques de votre application et obtenir des informations sur les nouvelles méthodes de configuration de votre application pour de meilleures performances. Pour en savoir plus, prenez connaissance de ce qui suit :
  - Suivi des performances et de la disponibilité en temps réel dans Amazon CloudWatch Internet Monitor (onglet Aperçu)

  - Filtrage et affichage des données historiques dans Amazon CloudWatch Internet Monitor (onglet Explorateur historique)
  - Obtenir des informations pour améliorer les performances des applications dans Amazon CloudWatch Internet Monitor (onglet Traffic Insights)
- Configurez les seuils d'événements de santé pour modifier ce qui incite Moniteur Internet à créer un événement de santé pour votre application. Vous pouvez configurer des seuils globaux et des seuils locaux (réseau urbain). Pour en savoir plus, consultez la section Modifier les seuils relatifs aux événements de santé.
- Utilisez des AWS CLI commandes associées aux actions de l'API Internet Monitor pour afficher les informations relatives au profil du trafic, consulter les mesures, répertorier les événements de santé, etc. Pour en savoir plus, veuillez consulter la section Exemples d'utilisation de la CLI avec Amazon CloudWatch Internet Monitor.
- Utilisez CloudWatch des outils standard, tels que CloudWatch Contributor Insights, CloudWatch Metrics Explorer et CloudWatch Logs Insights pour visualiser les données CloudWatch. Pour en savoir plus, veuillez consulter la section Exploration de vos données à l'aide CloudWatch d'outils et de l'interface de requête Internet Monitor.
- Utilisez Athena avec les journaux S3 pour accéder aux mesures Internet de Moniteur Internet et les analyser pour votre application, si vous avez activé la publication des mesures dans S3.
- Créez EventBridge des notifications Amazon pour vous avertir lorsqu'Internet Monitor détecte un problème de santé. Pour en savoir plus, veuillez consulter la section Utilisation d'Amazon CloudWatch Internet Monitor avec Amazon EventBridge.
- Recevez une AWS Health Dashboard notification automatiquement lorsqu'Internet Monitor détermine qu'un problème est dû au AWS réseau. La notification inclut les AWS mesures prises pour atténuer le problème.

Modifier ou supprimer un moniteur Moniteur Internet

À l'aide du menu Action, vous pouvez modifier ou supprimer un moniteur dans Amazon CloudWatch Internet Monitor après l'avoir créé. Par exemple, vous pouvez modifier un moniteur pour effectuer les opérations suivantes :

- modifier le pourcentage de trafic d'application à surveiller ;
- définir ou mettre à jour la limite maximale de réseaux urbains ;
- modifier les seuils des événements de santé en fonction des scores disponibilité ou de performance ;

- ajouter ou supprimer des ressources ;
- activer ou mettre à jour les événements de publication sur Amazon S3.

Vous pouvez également supprimer un moniteur. Notez que vous ne pouvez pas modifier le nom d'un moniteur après l'avoir créé.

Pour apporter des modifications à un moniteur ou en supprimer un, appliquez l'une des procédures suivantes.

Pour modifier un moniteur

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, sous Surveillance du réseau, choisissez Internet Monitor.
3. Choisissez votre moniteur, puis le menu Action.
4. Choisissez Mettre à jour un moniteur.
5. Effectuez les mises à jour souhaitées. Par exemple, pour modifier le pourcentage de trafic à surveiller, sous Trafic d'application à surveiller, sélectionnez ou saisissez un pourcentage.
6. Choisissez Mettre à jour.

Pour supprimer un moniteur

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, sous Surveillance du réseau, choisissez Internet Monitor.
3. Choisissez votre moniteur, puis le menu Action.
4. Choisissez Désactiver.
5. Choisissez à nouveau le menu Action, puis sélectionnez Supprimer.

Pour plus d'informations sur les options que vous pouvez mettre à jour, consultez les rubriques suivantes :

- Pour en savoir plus sur les ressources que vous ajoutez dans Moniteur Internet, consultez Ajouter des ressources à votre moniteur.
- Pour en savoir plus sur le pourcentage de trafic d'application, consultez Choisir le pourcentage de trafic d'application à surveiller.

- Pour en savoir plus sur la modification du seuil pour les événements de santé, consultez Modifier les seuils relatifs aux événements de santé.
- Pour en savoir plus sur la limite maximale de réseaux urbains, consultez Choisir une limite maximale de réseaux urbains.
- Pour en savoir plus sur le choix de publier des événements sur S3, consultez Publication de mesures Internet sur Amazon S3 dans Amazon CloudWatch Internet Monitor.

Ajouter ou créer un moniteur Amazon CloudWatch Internet Monitor avec Amazon VPC

Lorsque vous créez un VPC Amazon Virtual Private Cloud dans le AWS Management Console, vous pouvez éventuellement choisir de configurer également sa surveillance dans Amazon CloudWatch Internet Monitor. Vous pouvez ajouter le VPC à un moniteur existant ou créer un moniteur pour le VPC dans la console Amazon VPC.

En utilisant le Moniteur Internet avec votre VPC, vous pouvez consulter et évaluer les mesures et les métriques relatives à la disponibilité, aux performances, aux nombre d'octets surveillés transférés ainsi qu'aux temps d'aller-retour spécifiques aux sites clients et aux ASN de votre application (généralement des fournisseurs de services Internet). Le Moniteur Internet détermine également les anomalies de performance et de disponibilité et crée des événements d'état sur votre moniteur, dont vous pouvez choisir de recevoir des notifications. Pour en savoir plus sur la façon dont vous pouvez utiliser un moniteur pour gérer et améliorer l'expérience de vos clients avec votre application, veuillez consulter Utiliser un moniteur Moniteur Internet.

> **Important**
>
> Pour créer un moniteur ou ajouter un VPC à un moniteur existant, vous devez disposer des autorisations appropriées. Pour plus d’informations, consultez Identity and Access Management pour Amazon CloudWatch Internet Monitor.

Ajout d'un VPC à un moniteur existant

Vous pouvez demander à Amazon CloudWatch Internet Monitor d'ajouter un nouveau VPC à un moniteur existant pour vous lorsque vous créez le VPC dans le. AWS Management Console Après avoir ajouté le VPC, attendez quelques minutes, puis les métriques du VPC commenceront à s'afficher sur la console Internet Monitor.

Vous pouvez modifier le moniteur à tout moment, pour supprimer le VPC ou ajouter un autre VPC ou d'autres ressources. Vous pouvez également modifier le pourcentage du trafic que vous surveillez ou apporter d'autres modifications. Si vous choisissez de supprimer le VPC du moniteur, le trafic des clients vers ce VPC n'est plus surveillé par le Moniteur Internet.

Pour en savoir plus sur la mise à jour d'un moniteur, veuillez consulter Modifier ou supprimer un moniteur Moniteur Internet.

Création d'un moniteur pour un VPC

Si vous choisissez de créer un moniteur pour un VPC, l'assistant Créer un moniteur vous explique les étapes à suivre. Vous ajoutez le VPC en tant que ressource surveillée lorsque vous créez le moniteur. Si vous le souhaitez, vous pouvez également choisir un pourcentage du trafic client que vous souhaitez surveiller pour votre application (la valeur par défaut est 100 %).

Pour en savoir plus, veuillez consulter Création d'un moniteur dans Amazon CloudWatch Internet Monitor à l'aide de la console.

Tarification

Avec Amazon CloudWatch Internet Monitor, vous ne payez que pour ce que vous utilisez. La tarification du Moniteur Internet comporte deux éléments : des frais par ressource surveillée et des frais par réseau urbain. Un réseau urbain est l'emplacement à partir duquel les clients accèdent aux ressources de votre application et le réseau (un ASN, tel qu'un fournisseur de services Internet ou un FAI) par lequel les clients accèdent aux ressources.

Pour plus d'informations, y compris des exemples de tarification, voir Tarification d'Amazon CloudWatch Internet Monitor

Arrêt de la surveillance d'un VPC

Si vous souhaitez arrêter de surveiller votre ressource VPC avec Internet Monitor, procédez comme suit dans la console Internet Monitor :

Pour supprimer une ressource d'un moniteur

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, sous Surveillance du réseau, choisissez Internet Monitor.
3. Choisissez votre moniteur, puis le menu Action.
4. Choisissez Mettre à jour un moniteur.

5. Sous Ressources ajoutées, choisissez Supprimer des ressources.
6. Choisissez le VPC à supprimer, puis choisissez Supprimer.
7. Choisissez Mettre à jour.

Ajoutez ou créez un moniteur Amazon CloudWatch Internet Monitor avec CloudFront

Sur le tableau de bord des métriques d'une distribution dans CloudFront la console Amazon, vous pouvez configurer une surveillance supplémentaire pour une distribution dans Amazon CloudWatch Internet Monitor. Vous pouvez ajouter la distribution à un moniteur existant ou créer un nouveau moniteur pour la distribution.

En utilisant Internet Monitor avec votre CloudFront distribution, vous pouvez consulter et évaluer les mesures relatives à la disponibilité, aux performances, aux octets surveillés transférés et aux temps d'aller-retour spécifiques aux sites clients et aux ASN de votre application (généralement des fournisseurs de services Internet). Le Moniteur Internet détermine également les anomalies de performance et de disponibilité et crée des événements d'état sur votre moniteur, dont vous pouvez choisir de recevoir des notifications. Pour en savoir plus sur la façon dont vous pouvez utiliser un moniteur pour gérer et améliorer l'expérience de vos clients avec votre application, veuillez consulter Utiliser un moniteur Moniteur Internet.

> **Important**
>
> Pour créer un moniteur ou ajouter une distribution à un moniteur existant, vous devez disposer des autorisations appropriées. Pour plus d'informations, consultez Identity and Access Management pour Amazon CloudWatch Internet Monitor.

Ajouter une distribution à un moniteur existant

Vous pouvez choisir de demander à Internet Monitor d'ajouter une distribution à un moniteur existant directement depuis le tableau de bord CloudFront des métriques du AWS Management Console. Après avoir ajouté la distribution, attendez quelques minutes, puis les mesures relatives à la distribution commenceront à s'afficher sur la console Internet Monitor.

Vous pouvez modifier le moniteur à tout moment pour supprimer la distribution ou ajouter une autre distribution ou d'autres ressources. Vous pouvez également modifier le pourcentage du trafic que vous surveillez ou apporter d'autres modifications. Si vous choisissez de supprimer la distribution du moniteur, le trafic des clients vers cette distribution n'est plus surveillé par Internet Monitor.

Pour en savoir plus sur la mise à jour d'un moniteur, veuillez consulter Modifier ou supprimer un moniteur Moniteur Internet.

Création d'un moniteur pour une distribution

Si vous choisissez de créer un moniteur pour une distribution, l'assistant de création de moniteur vous explique les étapes à suivre. Vous ajoutez la distribution en tant que ressource surveillée lorsque vous créez le moniteur. Si vous le souhaitez, vous pouvez également choisir un pourcentage du trafic client que vous souhaitez surveiller pour votre application (la valeur par défaut est 100 %).

Pour en savoir plus, veuillez consulter Création d'un moniteur dans Amazon CloudWatch Internet Monitor à l'aide de la console.

Tarification

Avec Amazon CloudWatch Internet Monitor, vous ne payez que pour ce que vous utilisez. La tarification du Moniteur Internet comporte deux éléments : des frais par ressource surveillée et des frais par réseau urbain. Un réseau urbain est l'emplacement à partir duquel les clients accèdent aux ressources de votre application et le réseau (un ASN, tel qu'un fournisseur de services Internet ou un FAI) par lequel les clients accèdent aux ressources.

Pour plus d'informations, y compris des exemples de tarification, voir Tarification d'Amazon CloudWatch Internet Monitor

Arrêter de surveiller une distribution

Si vous souhaitez arrêter de surveiller votre ressource de distribution avec Internet Monitor, procédez comme suit dans la console Internet Monitor :

Pour supprimer une ressource d'un moniteur

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation de gauche, sous Surveillance du réseau, choisissez Internet Monitor.
3. Choisissez votre moniteur, puis le menu Action.
4. Choisissez Mettre à jour un moniteur.
5. Sous Ressources ajoutées, choisissez Supprimer des ressources.
6. Choisissez la distribution à supprimer, puis choisissez Supprimer.
7. Choisissez Mettre à jour.

Exemples d'utilisation de la CLI avec Amazon CloudWatch Internet Monitor

Cette section contient des exemples d'utilisation des opérations AWS Command Line Interface avec Amazon CloudWatch Internet Monitor.

Avant de commencer, assurez-vous de vous connecter pour utiliser le AWS CLI même AWS compte qui possède les Amazon Virtual Private Clouds (VPC), les Network Load Balancers, les CloudFront distributions Amazon ou les WorkSpaces annuaires Amazon que vous souhaitez surveiller. Le Moniteur Internet ne prend pas en charge l'accès aux ressources entre comptes. Pour plus d'informations sur l'utilisation du AWS CLI, consultez la référence des AWS CLI commandes. Pour plus d'informations sur l'utilisation des actions d'API avec Amazon CloudWatch Internet Monitor, consultez le guide de référence des API Amazon CloudWatch Internet Monitor.

Rubriques

Création d'une surveillance

Lorsque vous créez un moniteur dans Moniteur Internet, vous fournissez un nom et associez des ressources au moniteur pour indiquer où se trouve le trafic Internet de votre application. Vous spécifiez un pourcentage de trafic qui définit la part du trafic de votre application qui est surveillée. Cela détermine également le nombre de réseaux urbains, c'est-à-dire les emplacements et les ASN (généralement des fournisseurs de services Internet, ou FSI) client qui sont surveillés. Vous pouvez également définir un nombre maximum de réseaux urbains à surveiller pour vos ressources d'application afin de mieux contrôler votre facture. Pour plus d’informations, consultez Choisir une limite maximale de réseaux urbains.

Enfin, vous pouvez choisir de publier toutes les mesures Internet de votre application sur Amazon S3. Les mesures Internet pour les 500 principaux réseaux urbains (par volume de trafic) sont

automatiquement publiées dans CloudWatch Logs by Internet Monitor, mais vous pouvez également choisir de publier toutes les mesures sur S3.

Pour créer un moniteur avec le AWS CLI, vous devez utiliser la `create-monitor` commande. La commande suivante crée un moniteur qui surveille 100 % du trafic, mais fixe une limite maximale de 10 000 réseaux urbains, ajoute une ressource VPC et choisit de publier les mesures Internet sur Amazon S3.

> **Note**
>
> Internet Monitor publie dans CloudWatch Logs les mesures Internet toutes les cinq minutes pour les 500 principaux réseaux urbains (sites clients et ASN, généralement des fournisseurs de services Internet ou des FAI) qui envoient du trafic vers chaque moniteur. Vous pouvez choisir de publier les mesures Internet pour tous les réseaux urbains surveillés (jusqu'à la limite de service de 500 000 réseaux urbains) dans un compartiment Amazon S3. Pour plus d'informations, consultez Publication de mesures Internet sur Amazon S3 dans Amazon CloudWatch Internet Monitor.

```
aws internetmonitor --create-monitor monitor-name "TestMonitor" \
    --traffic-percentage-to-monitor 100 \
    --max-city-networks-to-monitor 10000 \
    --resources "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --internet-measurements-log-delivery
 S3Config="{BucketName=MyS3Bucket,LogDeliveryStatus=ENABLED}"
```

```
{
"Arn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
 "Status": "ACTIVE"
}
```

> **Note**
>
> Vous ne pouvez pas modifier le nom d'un moniteur.

Afficher les détails du contrôle

Pour afficher les informations relatives à un moniteur doté du AWS CLI, vous devez utiliser la `get-monitor` commande.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor"
```

```
{
    "ClientLocationType": "city",
    "CreatedAt": "2022-09-22T19:27:47Z",
    "ModifiedAt": "2022-09-22T19:28:30Z",
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "MonitorName": "TestMonitor",
    "ProcessingStatus": "OK",
    "ProcessingStatusInfo": "The monitor is actively processing data",
    "Resources": [
        "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889"
    ],
    "MaxCityNetworksToMonitor": 10000,
    "Status": "ACTIVE"
}
```

Lister les événements de l'état

Lorsque les performances se dégradent pour le trafic Internet de votre application, Moniteur Internet crée des événements de l'état dans votre moniteur. Pour consulter la liste des problèmes de santé actuels associés au AWS CLI, utilisez la `list-health-events` commande

```
aws internetmonitor list-health-events --monitor-name "TestMonitor"
```

```
{
    "HealthEvents": [
        {
            "EventId": "2022-06-20T01-05-05Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:15:14Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
```

```
            "PercentOfTotalTrafficImpacted": 1.21,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 60.370000000000005,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                }
            ],
            "StartedAt": "2022-06-20T01:05:05Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-05-05Z/latency"
        },
        {
            "EventId": "2022-06-20T01-17-56Z/latency",
            "Status": "RESOLVED",
            "EndedAt": "2022-06-20T01:30:23Z",
            "ServiceLocations": [
                {
                    "Name": "us-east-1"
                }
            ],
            "PercentOfTotalTrafficImpacted": 1.29,
            "ClientLocations": [
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 75.32,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
```

```
                            "NetworkName": "Amazon.com"
                        }
                    ],
                    "NetworkEventType": "AWS"
                },
                "NetworkName": "DIGITALOCEAN-ASN"
            },
            {
                "City": "Lockport",
                "PercentOfClientLocationImpacted": 22.91,
                "PercentOfTotalTraffic": 2.01,
                "Country": "United States",
                "Longitude": -78.6913,
                "AutonomousSystemNumber": 26101,
                "Latitude": 43.1721,
                "Subdivision": "New York",
                "NetworkName": "YAHOO-BF1"
            },
            {
                "City": "Hangzhou",
                "PercentOfClientLocationImpacted": 2.88,
                "PercentOfTotalTraffic": 0.7799999999999999,
                "Country": "China",
                "Longitude": 120.1612,
                "AutonomousSystemNumber": 37963,
                "Latitude": 30.2994,
                "Subdivision": "Zhejiang",
                "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
            }
        ],
        "StartedAt": "2022-06-20T01:17:56Z",
        "ImpactType": "PERFORMANCE",
        "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-17-56Z/latency"
    },
    {
        "EventId": "2022-06-20T01-34-20Z/latency",
        "Status": "RESOLVED",
        "EndedAt": "2022-06-20T01:35:04Z",
        "ServiceLocations": [
            {
                "Name": "us-east-1"
            }
        ],
```

```
            "PercentOfTotalTrafficImpacted": 1.15,
            "ClientLocations": [
                {
                    "City": "Lockport",
                    "PercentOfClientLocationImpacted": 39.45,
                    "PercentOfTotalTraffic": 2.01,
                    "Country": "United States",
                    "Longitude": -78.6913,
                    "AutonomousSystemNumber": 26101,
                    "Latitude": 43.1721,
                    "Subdivision": "New York",
                    "NetworkName": "YAHOO-BF1"
                },
                {
                    "City": "Toronto",
                    "PercentOfClientLocationImpacted": 29.770000000000003,
                    "PercentOfTotalTraffic": 1.05,
                    "Country": "Canada",
                    "Longitude": -79.3623,
                    "AutonomousSystemNumber": 14061,
                    "Latitude": 43.6547,
                    "Subdivision": "Ontario",
                    "CausedBy": {
                        "Status": "ACTIVE",
                        "Networks": [
                            {
                                "AutonomousSystemNumber": 16509,
                                "NetworkName": "Amazon.com"
                            }
                        ],
                        "NetworkEventType": "AWS"
                    },
                    "NetworkName": "DIGITALOCEAN-ASN"
                },
                {
                    "City": "Hangzhou",
                    "PercentOfClientLocationImpacted": 2.88,
                    "PercentOfTotalTraffic": 0.7799999999999999,
                    "Country": "China",
                    "Longitude": 120.1612,
                    "AutonomousSystemNumber": 37963,
                    "Latitude": 30.2994,
                    "Subdivision": "Zhejiang",
                    "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
```

```
                }
            ],
            "StartedAt": "2022-06-20T01:34:20Z",
            "ImpactType": "PERFORMANCE",
            "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/
TestMonitor/health-event/2022-06-20T01-34-20Z/latency"
        }
    ]
}
```

Afficher un événement de l'état spécifique

Pour obtenir des informations plus détaillées sur un événement de l'état spécifique avec la CLI, exécutez la commande `get-health-event` avec le nom de votre moniteur et un ID d'événement de l'état.

```
aws internetmonitor get-monitor --monitor-name "TestMonitor" --event-id "health-event/
TestMonitor/2021-06-03T01:02:03Z/latency"
```

```
{
    "EventId": "2022-06-20T01-34-20Z/latency",
    "Status": "RESOLVED",
    "EndedAt": "2022-06-20T01:35:04Z",
    "ServiceLocations": [
        {
            "Name": "us-east-1"
        }
    ],
    "EventArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor/
health-event/2022-06-20T01-34-20Z/latency",
    "LastUpdatedAt": "2022-06-20T01:35:04Z",
    "ClientLocations": [
        {
            "City": "Lockport",
            "PercentOfClientLocationImpacted": 39.45,
            "PercentOfTotalTraffic": 2.01,
            "Country": "United States",
            "Longitude": -78.6913,
            "AutonomousSystemNumber": 26101,
            "Latitude": 43.1721,
            "Subdivision": "New York",
            "NetworkName": "YAHOO-BF1"
```

```
        },
        {
            "City": "Toronto",
            "PercentOfClientLocationImpacted": 29.770000000000003,
            "PercentOfTotalTraffic": 1.05,
            "Country": "Canada",
            "Longitude": -79.3623,
            "AutonomousSystemNumber": 14061,
            "Latitude": 43.6547,
            "Subdivision": "Ontario",
            "CausedBy": {
                "Status": "ACTIVE",
                "Networks": [
                    {
                        "AutonomousSystemNumber": 16509,
                        "NetworkName": "Amazon.com"
                    }
                ],
                "NetworkEventType": "AWS"
            },
            "NetworkName": "DIGITALOCEAN-ASN"
        },
        {
            "City": "Shenzhen",
            "PercentOfClientLocationImpacted": 4.07,
            "PercentOfTotalTraffic": 0.61,
            "Country": "China",
            "Longitude": 114.0683,
            "AutonomousSystemNumber": 37963,
            "Latitude": 22.5455,
            "Subdivision": "Guangdong",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        },
        {
            "City": "Hangzhou",
            "PercentOfClientLocationImpacted": 2.88,
            "PercentOfTotalTraffic": 0.7799999999999999,
            "Country": "China",
            "Longitude": 120.1612,
            "AutonomousSystemNumber": 37963,
            "Latitude": 30.2994,
            "Subdivision": "Zhejiang",
            "NetworkName": "Hangzhou Alibaba Advertising Co.,Ltd."
        }
```

```
    ],
    "StartedAt": "2022-06-20T01:34:20Z",
    "ImpactType": "PERFORMANCE",
    "PercentOfTotalTrafficImpacted": 1.15
}
```

Afficher la liste des moniteurs

Pour voir une liste de tous les moniteurs de votre compte avec la CLI, exécutez la commande `list-monitors`.

```
aws internetmonitor list-monitors
```

```
{
    "Monitors": [
        {
            "MonitorName": "TestMonitor",
            "ProcessingStatus": "OK",
            "Status": "ACTIVE"
        }
    ],
    "NextToken": " zase12"
}
```

Modifier le moniteur

Pour mettre à jour les informations sur votre moniteur à l'aide de la CLI, utilisez la commande `update-monitor` et spécifiez le nom du moniteur à mettre à jour. Vous pouvez mettre à jour le pourcentage de trafic à surveiller, la limite maximale de réseaux urbains à surveiller, ajouter ou supprimer les ressources utilisées par Moniteur Internet pour surveiller le trafic et modifier le statut du moniteur de `ACTIVE` à `INACTIVE`, ou vice versa. Notez que vous ne pouvez pas modifier le nom du moniteur.

La réponse à un appel `update-monitor` renvoie uniquement les valeurs `MonitorArn` et `Status`.

L'exemple suivant montre comment utiliser la commande `update-monitor` pour définir le nombre maximal de réseaux urbains à surveiller sur `50000` :

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --max-city-networks-to-monitor 50000
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": " ACTIVE "
}
```

L'exemple suivant montre comment ajouter et supprimer des ressources :

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" \
    --resources-to-add "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-11223344556677889" \
    --resources-to-remove "arn:aws:ec2:us-east-1:111122223333:vpc/vpc-2222444455556666"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "ACTIVE"
}
```

L'exemple suivant montre comment utiliser la commande `update-monitor` pour changer le statut du moniteur en `INACTIVE` :

```
aws internetmonitor update-monitor --monitor-name "TestMonitor" --status "INACTIVE"
```

```
{
    "MonitorArn": "arn:aws:internetmonitor:us-east-1:111122223333:monitor/TestMonitor",
    "Status": "INACTIVE"
}
```

Supprimer un moniteur

Vous pouvez supprimer un moniteur avec la CLI en utilisant la commande `delete-monitor`. Tout d'abord, vous devez configurer le moniteur pour qu'il soit inactif. Pour ce faire, utilisez la commande `update-monitor` pour changer le statut en `INACTIVE`. Confirmez que le moniteur est inactif en utilisant la commande `get-monitor` et en vérifiant le statut.

Lorsque le statut du moniteur est `INACTIVE`, vous pouvez alors utiliser la CLI pour exécuter la commande `delete-monitor` afin de supprimer le moniteur. La réponse à un appel `delete-monitor` réussi est vide.

```
aws internetmonitor delete-monitor --monitor-name "TestMonitor"
```

```
{}
```

Surveillance et optimisation avec le tableau de bord du Moniteur Internet

Les informations contenues dans cette section expliquent comment filtrer et afficher les informations sur le tableau de bord Amazon CloudWatch Internet Monitor afin de visualiser et d'obtenir des informations sur le trafic Internet et la configuration de votre AWS application.

Une fois que vous avez créé un moniteur pour surveiller les performances et la disponibilité Internet de votre application, Amazon CloudWatch Internet Monitor publie des CloudWatch journaux contenant des mesures Internet pour les paires location-réseau du client (ville-réseau) et publie des CloudWatch mesures agrégées pour le trafic vers votre application, ainsi que vers chaque Région AWS emplacement périphérique. Vous pouvez filtrer, explorer et obtenir des suggestions orientées action à partir de ces informations du Moniteur Internet de différentes manières.

Pour commencer, sur la CloudWatch console, sous Surveillance du réseau, sélectionnez Internet Monitor.

Cette section décrit principalement comment filtrer et afficher les métriques d'Internet Monitor à l'aide du AWS Management Console. Vous pouvez également utiliser les opérations de l'API Internet Monitor avec le SDK AWS CLI ou un SDK pour travailler directement avec les événements Internet Monitor stockés dans CloudWatch des fichiers journaux. Pour plus d'informations, consultez la section relative à l'utilisation de votre moniteur et des mesures. Pour plus d'informations sur l'utilisation des opérations d'API, consultez Exemples d'utilisation de la CLI avec Amazon CloudWatch Internet Monitor et le manuel Amazon CloudWatch Internet Monitor API Reference.

Le tableau de bord Moniteur Internet comporte trois onglets :

- Dans l'onglet Overview (Vue d'ensemble), vous pouvez voir les informations actuelles et historiques sur les performances et la disponibilité de votre application, ainsi que les événements de l'état ayant un impact sur les emplacements de vos clients.
- Dans l'onglet suivant, Explorateur historique, vous pouvez filtrer par emplacement, ASN, date, etc., et visualiser les métriques de votre trafic Internet au fil du temps, à l'aide des graphiques.
- Dans l'onglet Informations sur le trafic, en plus d'afficher des informations sur le trafic le plus surveillé, résumées de plusieurs façons personnalisables, vous pouvez obtenir des suggestions de configurations optimisées pour améliorer les performances pour différentes paires d'emplacements et d'ASN. Internet Monitor prédit l'amélioration des performances de votre application, en fonction

de vos modèles de trafic et de vos performances passées, lorsque vous modifiez la façon dont vous acheminez votre trafic ou les AWS ressources que vous utilisez. Vous pouvez également consulter un graphique pour comparer le nombre de réseaux urbains inclus dans votre couverture de surveillance, en fonction du pourcentage de trafic d'application que vous avez choisi pour votre moniteur.

En outre, étant donné qu'Internet Monitor génère et publie des fichiers journaux contenant les mesures relatives à votre trafic, vous pouvez utiliser d'autres CloudWatch outils de la console pour mieux visualiser les données publiées par Internet Monitor, notamment les informations sur les CloudWatch contributeurs, CloudWatch les métriques et les informations sur CloudWatch les journaux. Pour plus d'informations, consultez Exploration de vos données à l'aide CloudWatch d'outils et de l'interface de requête Internet Monitor.

Découvrez comment utiliser Moniteur Internet pour explorer vos mesures de performance et de disponibilité dans les sections suivantes.

Rubriques

- Suivi des performances et de la disponibilité en temps réel dans Amazon CloudWatch Internet Monitor (onglet Aperçu)
- Filtrage et affichage des données historiques dans Amazon CloudWatch Internet Monitor (onglet Explorateur historique)
- Obtenir des informations pour améliorer les performances des applications dans Amazon CloudWatch Internet Monitor (onglet Traffic Insights)

Suivi des performances et de la disponibilité en temps réel dans Amazon CloudWatch Internet Monitor (onglet Aperçu)

Utilisez l'onglet Vue d'ensemble de la CloudWatch console, sous Internet Monitor, pour obtenir une vue globale des performances et de la disponibilité du trafic suivi par votre moniteur. Cet onglet affiche également une carte de synthèse du trafic Internet, avec des clusters de trafic qui peuvent vous aider à visualiser le trafic mondial de votre application, ainsi que l'emplacement et l'impact des événements de santé.

Scores de santé

Le graphique des scores de santé vous indique les informations de performance et de disponibilité pour votre trafic mondial. AWS dispose de données historiques substantielles sur les

performances Internet et la disponibilité du trafic réseau entre les emplacements géographiques pour différents ASN et AWS services. Internet Monitor utilise les données de connectivité capturées à partir de son réseau mondial pour calculer une base de référence en matière de performances et de disponibilité pour le trafic Internet. AWS Il s'agit des mêmes données que celles que nous utilisons AWS pour surveiller notre propre disponibilité et disponibilité d'Internet.

Avec ces mesures comme base de référence, Moniteur Internet peut détecter quand les performances et la disponibilité de votre application ont baissé, par rapport à la base de référence. Pour faciliter la détection de ces baisses, nous vous communiquons ces informations sous la forme d'un score de performance et d'un score de disponibilité. Pour plus d'informations, consultez Exploration de vos données à l'aide CloudWatch d'outils et de l'interface de requête Internet Monitor.

Le graphique Scores d'état inclut les événements de santé qui se sont produits au cours d'une période que vous choisissez. Lorsqu'il y a un événement de l'état, vous voyez une baisse dans la ligne de performance ou de disponibilité sur le graphique. Si vous sélectionnez l'événement, vous verrez plus de détails et des bandes apparaissent sur le graphique, avec des informations de date et d'heure indiquant la durée de l'événement.

Vous pouvez également examiner ces métriques en accédant directement aux fichiers journaux pour chaque point de données. Dans le menu Actions, choisissez Afficher CloudWatch les journaux.

Vue d'ensemble du trafic Internet

La carte Aperçu du trafic Internet vous montre le trafic Internet et les événements de santé qui sont spécifiques aux emplacements et aux ASN depuis lesquels vos utilisateurs accèdent à votre application. Les pays qui sont en gris sur la carte sont ceux qui incluent le trafic pour votre application.

Chaque cercle sur la carte indique un événement de santé dans une zone, pour une période de temps que vous sélectionnez. Internet Monitor crée des événements de santé lorsqu'il détecte un problème, à un seuil spécifique, lié à la connectivité entre l'une de vos ressources hébergées AWS et le réseau de la ville où un utilisateur accède à votre application. La sélection d'un cercle sur la carte affiche plus de détails sur l'événement de l'état pour cet emplacement. En outre, pour les clusters qui ont des événements de l'état, vous pouvez voir des informations détaillées dans le tableau Health events (Événements de l'état) sous la carte.

Notez que Moniteur Internet crée les événements de santé dans un moniteur lorsqu'il détermine qu'un événement a un impact global significatif sur votre application. Si aucun événement de

santé n'a eu un impact supérieur au seuil sur le trafic pour les emplacements des clients au cours de la période que vous avez sélectionnée, la carte sera vide. Pour plus d'informations, consultez Quand Internet Monitor crée et résout des problèmes de santé.

Modifier les seuils des événements de santé

Vous pouvez choisir comment et quand Moniteur Internet crée des événements de santé pour votre application. Choisissez Mettre à jour les seuils pour apporter des modifications.

Vous pouvez modifier le seuil global qui déclenche Moniteur Internet pour créer un événement de santé. Le seuil d'événements de santé par défaut, à la fois pour les scores de performance et les scores de disponibilité, est de 95 %. Autrement dit, lorsque le score global de performance ou de disponibilité de votre application est de 95 % ou moins, Moniteur Internet crée un événement de santé. En ce qui concerne le seuil global, l'événement de santé peut être déclenché par un seul problème important ou par la combinaison de plusieurs problèmes de moindre importance.

Vous pouvez également modifier le seuil local, c'est-à-dire le réseau urbain, qui, combiné à un pourcentage de niveau d'impact global, déclenchera un problème de santé. En définissant un seuil qui crée un événement de santé lorsqu'un score passe en dessous du seuil pour un ou plusieurs réseaux urbains (emplacements et ASN, généralement des FSI), vous pouvez savoir quand des problèmes surviennent dans les emplacements à faible trafic, par exemple.

Une option de seuil local supplémentaire fonctionne conjointement avec le seuil local des scores de disponibilité ou de performance. Le deuxième facteur est le pourcentage de votre trafic global qui doit être impacté avant que Moniteur Internet ne crée un événement de santé basé sur le seuil local.

En configurant les options de seuil pour le trafic global et le trafic local, vous pouvez affiner la fréquence à laquelle les événements de santé sont créés, en fonction de l'utilisation de votre application et de vos besoins. Sachez que lorsque vous définissez un seuil local inférieur, un plus grand nombre d'événements de santé sont généralement créés, en fonction de votre application et des autres valeurs de configuration de seuil que vous définissez.

En résumé, vous pouvez configurer les seuils des événements de santé (pour les scores de performance, les scores de disponibilité ou les deux) des manières suivantes :

- en choisissant différents seuils globaux pour déclencher un événement de santé ;
- en choisissant différents seuils locaux pour déclencher un événement de santé ; avec cette option, vous pouvez également modifier le pourcentage d'impact global sur votre application qui doit être dépassé avant que Moniteur Internet ne crée un événement ;

- en désactivant le déclenchement d'événements de santé en fonction de seuils locaux ou en activant les options de seuils locaux.

Vous pouvez également configurer des options pour les scores de performance, les scores de disponibilité ou les deux. Vous pouvez configurer une combinaison de ces options ou uniquement l'une d'entre elles.

Pour mettre à jour les seuils et les autres options de configuration pour les scores de performance, les scores de disponibilité, ou les deux, procédez comme suit :

Pour modifier les options de configuration des seuils

1. Dans AWS Management Console le volet de CloudWatch navigation de gauche, naviguez vers, puis sélectionnez Internet Monitor.
2. Dans l'onglet Aperçu, dans la section Chronologie des événements de santé, choisissez Mettre à jour les seuils.
3. Sur la page de dialogue qui s'ouvre, choisissez les nouvelles valeurs et options que vous souhaitez pour les seuils et les autres options qui déclencheront Moniteur Internet pour créer un événement de santé. Vous pouvez effectuer les actions suivantes :
   - Choisissez une nouvelle valeur pour le Seuil de score de disponibilité, Seuil de score de performance, ou les deux.

     Les graphiques des sections relatives à chaque paramètre indiquent le seuil actuel et les scores de disponibilité ou de performance des événements de santé récents pour votre application. En regardant les valeurs typiques, vous pouvez vous faire une idée des valeurs pour lesquelles vous pourriez modifier un seuil.

     Conseil : pour afficher un graphique plus grand et modifier le calendrier, cliquez sur le bouton d'extension dans le coin supérieur droit du graphique.
   - Activez ou désactivez un seuil local de disponibilité ou de performance, ou les deux. Lorsqu'une option est activée, vous pouvez définir le seuil et le niveau d'impact pour que Moniteur Internet crée un événement de santé.
4. Après avoir configuré les options de seuil, enregistrez vos mises à jour en choisissant Mettre à jour les seuils d'événements de santé.

Pour en savoir plus sur le fonctionnement des problèmes de santé, consultez Quand Moniteur Internet crée et résout des événements de santé.

Table des événements de santé

La table Événements de santé répertorie les emplacements des clients qui ont été affectés par des événements de santé, ainsi que des informations sur ces événements. Les colonnes suivantes sont incluses dans la table.

	Description
Emplacement du client	L'emplacement des utilisateurs finaux qui ont été affectés par l'événement, qui ont connu une latence accrue ou une disponibilité réduite. Pour en savoir plus sur la précision de l'emplacement des clients dans Moniteur Internet, consultez Geolocation information and accuracy in Internet Monitor.
Impact sur le trafic	Le degré d'impact causé par l'événement, en termes de latence accrue ou de disponibilité réduite. Pour la latence, il s'agit du pourcentage de la latence accrue pendant l'événement par rapport aux performances habituelles du trafic, entre cet emplacement client et cet AWS emplacement en utilisant ce réseau client.
Réseau client	Le réseau par lequel le trafic a transité. Généralement, il s'agit du fournisseur de services Internet (FSI) ou du Numéro de système autonome (ASN) pour le trafic réseau.
AWS emplacement	L' AWS emplacement du trafic réseau, qui peut être un emplacement périphérique Région AWS ou un emplacement périphérique d'Internet.

	Description
Type d'impact	Le type d'impact pour l'événement de l'état. Les événements de l'état sont généralement causés par des augmentations de latence (problèmes de performance) ou d'accessibilité (problèmes de disponibilité). Vous pouvez également cliquer sur le type d'impact pour voir la cause de la dégradation. Dans la mesure du possible, Internet Monitor analyse l'origine d'un problème de santé afin de déterminer s'il a été causé par un ASN (fournisseur de services Internet) AWS ou par un ASN (fournisseur de services Internet). Notez que cette analyse se poursuit une fois l'événement résolu. Moniteur Internet peut mettre à jour les événements avec de nouvelles informations pendant une heure au maximum.

Si vous cliquez sur l'un des emplacements client dans la table Événements de santé, vous pouvez voir plus de détails sur l'événement de santé à cet emplacement. Par exemple, vous pouvez voir quand l'événement a commencé, quand il s'est terminé, et l'impact sur le trafic local.

Visualisation du chemin réseau

L'analyse des déficiences qui est terminée inclut un chemin d'accès au réseau complet sous Visualisation du chemin d'accès au réseau. Le chemin complet indique chaque nœud situé le long du chemin réseau de votre application pour l'événement médical, entre le AWS site et le client, pour une paire client-emplacement.

Si Moniteur Internet détermine la cause d'une déficience, celle-ci est marquée d'un cercle rouge en pointillé. Les déficiences peuvent être causées par les ASN, qui sont généralement des fournisseurs de services Internet (FSI), ou par AWS. Si la cause d'une déficience est multiple, plusieurs nœuds sont encerclés.

Filtrage et affichage des données historiques dans Amazon CloudWatch Internet Monitor (onglet Explorateur historique)

Utilisez l'onglet Explorateur historique de la CloudWatch console, sous Internet Monitor, pour filtrer et afficher les données de votre application qui se trouvent dans CloudWatch Logs. Internet Monitor publie dans des CloudWatch journaux des mesures spécifiques à votre application concernant la disponibilité, les performances, le nombre d'octets surveillés transférés (ou le nombre de connexions client, pour les WorkSpaces annuaires uniquement) et le temps de trajet aller-retour entre vos réseaux urbains surveillés. Régions AWS

> **Note**
>
> Internet Monitor publie des mesures Internet dans des CloudWatch journaux toutes les cinq minutes pour les 500 principaux réseaux urbains (en termes de volume de trafic) (c'est-à-dire les sites des clients et les ASN, généralement des fournisseurs de services Internet ou des ISP) qui envoient du trafic vers chaque moniteur. Vous pouvez choisir de publier les mesures Internet pour tous les réseaux urbains surveillés (jusqu'à la limite de service de 500 000 réseaux urbains) dans un compartiment Amazon S3. Pour plus d'informations, consultez Publication de mesures Internet sur Amazon S3 dans Amazon CloudWatch Internet Monitor.

Pour commencer à explorer les données de votre application, sélectionnez une période. Choisissez ensuite un emplacement géographique spécifique, comme une ville, et d'autres filtres (facultatif). Moniteur Internet applique les filtres aux données des journaux de mesures Internet qu'il a publiés pour les réseaux urbains afin de déterminer le trafic de votre application. Vous pouvez ensuite voir des graphiques des données qui indiquent le score de performance, le score de disponibilité, le nombre d'octets surveillés transférés (pour les VPC, les équilibreurs de charge réseau et les CloudFront distributions) ou le nombre de connexions client (pour les WorkSpaces annuaires) et le temps d'aller-retour (RTT) de votre application au fil du temps.

La table All events (Tous les événements) située sous les graphiques vous montre les événements de l'état que votre filtre renvoie pour le trafic de votre application, avec des informations sur chaque événement. Elle inclut les colonnes suivantes.

	Description
Début de l'événement	L'heure à laquelle l'événement de l'état a commencé.
Statut	Si l'événement est toujours actif ou est résolu.
Emplacement du client	L'emplacement des utilisateurs finaux qui ont été impactés par l'événement, qui ont connu une augmentation de la latence ou une réduction des performances. Pour en savoir plus sur la précision de l'emplacement des clients dans Moniteur Internet, consultez Geolocation information and accuracy in Internet Monitor.
Impact sur le trafic	L'impact pondéré de l'événement sur l'emplacement de l'événement de santé. Il s'agit, par exemple, de l'impact sur la latence, par rapport aux performances habituelles du trafic entre un site client et un AWS emplacement via l'ASN client, généralement un fournisseur de services Internet (ISP). De même, pour un événement qui affecte la disponibilité, vous pouvez constater l'impact sur la disponibilité par rapport à la disponibilité typique de l'emplacement du AWS client par rapport à l'ASN du client.
Durée de l'événement	Combien de temps l'événement a duré. Le Moniteur Internet met fin aux événements de l'état lorsqu'ils n'affectent plus de 5 % (au total) des emplacements clients de votre application.
FSI du client	L'ASN, généralement le fournisseur de services Internet (FSI), qui était le transporteur du trafic réseau.

	Description
Emplacement du service	Emplacement du service d'où provient le trafic réseau, qui peut être un emplacement périphérique Région AWS ou Internet.

Vous pouvez également consulter les mesures de votre application en accédant aux journaux directement pour chaque point de données. Dans le menu Actions, choisissez Afficher CloudWatch les journaux. Notez que, dans la mesure où les événements de mesure sont publiés sur votre compte lors de leur création, vous pouvez également créer d'autres CloudWatch tableaux de bord ou alarmes en fonction de ceux-ci. Pour plus d'informations, consultez Obtenir des informations pour améliorer les performances des applications dans Amazon CloudWatch Internet Monitor (onglet Traffic Insights) et Création d'alarmes avec Amazon CloudWatch Internet Monitor.

En plus de l'exploration et de l'analyse des mesures et métriques Moniteur Internet, ainsi que de la création éventuelle de tableaux de bord et d'alertes basés sur celles-ci, vous pouvez utiliser Moniteur Internet pour vous aider à comprendre comment vous pourriez améliorer les performances de votre application. L'onglet Traffic insights (Informations sur le trafic) offre plusieurs moyens de vous aider à explorer les options. Pour plus d'informations, consultez la section Suggestions d'optimisation du trafic de l'onglet Informations sur le trafic. Vous pouvez également consulter les exemples spécifiques dans le chapitre Cas d'utilisation de Moniteur Internet.

Obtenir des informations pour améliorer les performances des applications dans Amazon CloudWatch Internet Monitor (onglet Traffic Insights)

Utilisez l'onglet Informations sur le trafic de la CloudWatch console, sous Internet Monitor, pour consulter des informations récapitulatives sur le trafic le plus important (par volume) pour votre application. Vous pouvez filtrer et trier le trafic de votre application de différentes manières. Ensuite, faites défiler vers le bas et sélectionnez différentes combinaisons de configurations pour votre application afin de voir ce que Moniteur Internet suggère comme alternatives optimales pour obtenir les meilleures performances de temps jusqu'au premier octet (TTFB).

Internet Monitor publie dans CloudWatch Logs les mesures Internet toutes les cinq minutes pour les 500 principaux réseaux urbains (en termes de volume de trafic) (c'est-à-dire les sites des clients et les ASN, généralement des fournisseurs de services Internet ou des ISP) qui envoient du trafic vers chaque moniteur. Vous pouvez choisir de publier les mesures Internet pour tous les réseaux urbains surveillés (jusqu'à la limite de service de 500 000 réseaux urbains) dans un compartiment

Amazon S3. Pour plus d’informations, consultez Publication de mesures Internet sur Amazon S3 dans Amazon CloudWatch Internet Monitor.

Résumés de haut niveau du trafic

Vous pouvez commencer par afficher des résumés de haut niveau du trafic global et de la performance de votre application, sur une période spécifique, filtrés par emplacement du client. Vous pouvez également examiner la performance de votre application pour les principaux emplacements clients (ou les derniers) en matière de volume de trafic, en les filtrant et en les triant de plusieurs façons. Par exemple, vous pouvez les trier par granularité (c'est-à-dire par ville, subdivision, pays ou zone métropolitaine), par trafic total, par temps moyen jusqu'au premier octet (TTFB) et par d'autres facteurs.

Pour en savoir plus sur la précision de l'emplacement des clients dans Moniteur Internet, consultez Geolocation information and accuracy in Internet Monitor.

> **Note**
>
> Les filtres que vous utilisez s'appliquent à l'ensemble de la page. Ils ont donc une incidence sur les réseaux urbains inclus dans les graphiques récapitulatifs et les informations relatives au trafic total, ainsi que sur les réseaux urbains inclus dans la section Suggestions d'optimisation du trafic qui suit.

Suggestions d'optimisation du trafic

La section Suggestions d'optimisation du trafic affiche un ensemble filtré de réseaux urbains surveillés (emplacements et ASN, fournisseurs de services Internet) pour votre trafic, ainsi que le trafic client total pour chacun d'entre eux. Les entrées de la table sont basées sur les filtres que vous avez choisis pour le trafic de votre application pour Informations sur le trafic en haut de la page. Par défaut, ce sont les 10 premières villes en matière de volume de trafic. Vous voyez généralement plus de 10 lignes dans la table, car il y a une entrée pour chaque paire de réseaux urbains. En d'autres termes, il existe une ligne pour chaque combinaison d'emplacement (ville) et d'ASN (fournisseur de services Internet) via laquelle les clients accèdent à votre application, par exemple Dallas aux États-Unis et Comcast.

> **Note**
>
> Pour consulter les suggestions d'optimisation du trafic pour tous les réseaux urbains que vous surveillez, vous pouvez exécuter une requête directement dans CloudWatch Insights. Pour voir un exemple de requête qui n'inclut pas le filtre de granularité géographique qui limite la liste des réseaux urbains sur cette page, consultez Utilisation de CloudWatch Logs Insights avec Amazon CloudWatch Internet Monitor.

Dans cette section, sélectionnez différentes options : Amazon EC2 ou CloudFront les deux. Cela vous permet de voir quelles sont les valeurs du délai moyen prévu jusqu'au premier octet (TTFB) pour les clients lorsque vous utilisez votre application avec ces services dans différentes AWS régions, par rapport au TTFB actuel. Pour plus d'informations sur les calculs de TTFB, consultez la rubrique relative aux calculs de TTFB et de latence AWS.

En sélectionnant différentes options, puis en affichant les résultats dans la table, vous pouvez commencer à planifier des configurations et des déploiements susceptibles d'améliorer les performances de vos clients. Notez que vous pouvez voir un tiret (-) au lieu d'une valeur dans une colonne lorsque les données ne peuvent pas être affichées. Pour consulter un exemple spécifique d'amélioration des performances, consultez Utiliser Amazon CloudWatch Internet Monitor pour une meilleure expérience de jeu.

Par exemple, pour commencer, pour un réseau urbain spécifique (emplacement du client et paire ASN), essayez de sélectionner l' CloudFront option EC2 ou les deux. Pour chaque réseau urbain répertorié dans le tableau, Internet Monitor vous montre les améliorations de performances potentielles du TTFB, sur la base d'un choix de routage du trafic (via une option spécifique Région AWS) avec cette option, par rapport à la configuration actuelle. (Notez que, par souci d'exhaustivité, la table inclut également les routes déjà optimisées.) Par exemple, vous pouvez voir un TTFB prédit de 50 ms pour l'utilisation du routage EC2 via `us-east-1` par rapport à votre configuration actuelle avec un TTFB de 100 ms où vous utilisez le routage EC2 via `us-west-2`. Vous pourriez donc envisager le routage via `us-west-2`.

Autre exemple, vous pouvez sélectionner EC2, puis constater que cela ne fait pas de différence de performance mesurable pour un site client et un ASN, mais noter ensuite que lorsque vous sélectionnez CloudFront avec la même région, cela réduit quelque peu le TTFB. Cela suggère que si vous ajoutez une CloudFront distribution devant votre application, cela peut entraîner une amélioration des performances et vaut peut-être la peine d'essayer, pour cet emplacement client et cet ASN.

Exploration de vos données à l'aide CloudWatch d'outils et de l'interface de requête Internet Monitor

Outre la visualisation des performances et de la disponibilité de votre application à l'aide du tableau de bord Amazon CloudWatch Internet Monitor, vous pouvez utiliser plusieurs méthodes pour approfondir les données générées par Internet Monitor pour vous. Ces méthodes incluent l'utilisation CloudWatch d'outils utilisant les données Internet Monitor stockées dans des fichiers CloudWatch journaux et l'interface de requête Internet Monitor. Les outils que vous pouvez utiliser incluent CloudWatch Logs Insights, CloudWatch Metrics, CloudWatch Contributor Insights et Amazon Athena. Vous pouvez utiliser certains ou tous ces outils, ainsi que le tableau de bord, pour explorer les données du Moniteur Internet, en fonction de vos besoins.

Internet Monitor agrège les CloudWatch mesures relatives au trafic vers votre application et vers chacune d'entre elles Région AWS, et inclut des données telles que l'impact total sur le trafic, la disponibilité et le temps d'aller-retour. Ces données sont publiées dans CloudWatch Logs et peuvent également être utilisées avec l'interface de requête Internet Monitor. Les détails concernant la granularité géographique et d'autres aspects de l'information disponible à explorer varient pour chacun d'entre eux.

Amazon CloudWatch Internet Monitor publie des données pour votre moniteur à intervalles de 5 minutes, puis les met à disposition de différentes manières. Le tableau suivant répertorie les scénarios d'accès aux données du Moniteur Internet et décrit les caractéristiques des données collectées pour chacun d'entre eux.

Fonctionnalité	CloudWatch Journaux	Exporter vers S3	Interface de requête	CloudWatch tableau de bord
Activée par défaut.	Oui	Non	Oui	Oui
Nombre de réseaux urbains pour lesquels les données sont collectées	Top 500 (voir note ci-dessous)	Tous	Tous	Tous
Conservation des données	Contrôlé par l'utilisateur	Contrôlé par l'utilisateur	30 jours	30 jours

Fonctionnalité	CloudWatch Journaux	Exporter vers S3	Interface de requête	CloudWatch tableau de bord
Granularités géographiques pour lesquelles les données sont collectées	Tout (réseau urbain, réseau métropoli tain, réseau départemental, réseau national)	Réseau urbain	Tout (réseau urbain, réseau métropoli tain, réseau départemental, réseau national)	Tout (réseau urbain, réseau métropoli tain, réseau départemental, réseau national)
Comment interroger et filtrer les données	Utilisation de CloudWatch Logs Insights avec Amazon CloudWatch Internet Monitor	Utiliser Amazon Athena pour interroger les mesures Internet dans les fichiers journaux Amazon S3	Utilisation de l'interface de requête Amazon CloudWatch Internet Monitor	Surveillance et optimisation avec le tableau de bord du Moniteur Internet

Remarque : les 500 meilleures mesures sont capturées pour les réseaux urbains ; les 250 meilleures pour les réseaux métropolitains, les 100 meilleures pour les réseaux départementaux et les 50 meilleures pour les réseaux nationaux.

Ce chapitre décrit comment interroger et explorer vos données à l'aide d' CloudWatch outils ou de l'interface de requête d'Internet Monitor, ainsi que des exemples pour chaque méthode.

Table des matières

Utilisation de CloudWatch Logs Insights avec Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor publie des mesures détaillées de la disponibilité et du temps de trajet aller-retour dans CloudWatch Logs, et vous pouvez utiliser les requêtes CloudWatch

Logs Insights pour filtrer un sous-ensemble de journaux pour une ville ou une zone géographique spécifique (localisation du client), l'ASN du client (ISP) et l'emplacement source. AWS

Pour en savoir plus sur la précision de l'emplacement des clients dans Moniteur Internet, consultez Geolocation information and accuracy in Internet Monitor.

Les exemples présentés dans cette section peuvent vous aider à créer des requêtes CloudWatch Logs Insights pour en savoir plus sur les mesures et métriques du trafic de vos applications. Si vous utilisez ces exemples dans CloudWatch Logs Insights, remplacez *MonitorName par le nom* de votre propre moniteur.

Suggestions d'optimisation du trafic

Dans l'onglet Informations sur le trafic de Moniteur Internet, vous pouvez consulter les suggestions d'optimisation du trafic, filtrées par emplacement. Pour consulter les mêmes informations que celles affichées dans la section Suggestions d'optimisation du trafic de cet onglet, mais sans le filtre de granularité de localisation, vous pouvez utiliser la requête CloudWatch Logs Insights suivante.

1. Dans le AWS Management Console, accédez à CloudWatch Logs Insights.
2. Pour Log Group (Groupe de journaux), sélectionnez `/aws/internet-monitor/`*`monitorName`*`/byCity` et `/aws/internet-monitor/`*`monitorName`*`/byCountry`, puis spécifiez une plage de temps.
3. Ajoutez la requête suivante, puis exécutez-la.

```
fields @timestamp,
clientLocation.city as @city, clientLocation.subdivision as @subdivision,
 clientLocation.country as @country,
`trafficInsights.timeToFirstByte.currentExperience.serviceName` as @serviceNameField,
concat(@serviceNameField, ` (`, `serviceLocation`, `)`) as @currentExperienceField,
concat(`trafficInsights.timeToFirstByte.ec2.serviceName`, ` (`,
 `trafficInsights.timeToFirstByte.ec2.serviceLocation`, `)`) as @ec2Field,
`trafficInsights.timeToFirstByte.cloudfront.serviceName` as @cloudfrontField,
concat(`clientLocation.networkName`, ` (AS`, `clientLocation.asn`, `)`) as @networkName
| filter ispresent(`trafficInsights.timeToFirstByte.currentExperience.value`)
| stats avg(`trafficInsights.timeToFirstByte.currentExperience.value`) as @averageTTFB,
avg(`trafficInsights.timeToFirstByte.ec2.value`) as @ec2TTFB,
avg(`trafficInsights.timeToFirstByte.cloudfront.value`) as @cloudfrontTTFB,
sum(`bytesIn` + `bytesOut`) as @totalBytes,
latest(@ec2Field) as @ec2,
latest(@currentExperienceField) as @currentExperience,
```

```
latest(@cloudfrontField) as @cloudfront,
count(*) by @networkName, @city, @subdivision, @country
| display @city, @subdivision, @country, @networkName, @totalBytes, @currentExperience,
 @averageTTFB, @ec2, @ec2TTFB, @cloudfront, @cloudfrontTTFB
| sort @totalBytes desc
```

Afficher la disponibilité d'Internet et le RTT (p50, p90 et p95)

Pour consulter la disponibilité d'Internet et le temps d'aller-retour (p50, p90 et p95) du trafic, vous pouvez utiliser la requête Logs Insights suivante. CloudWatch

Zone géographique de l'utilisateur final : Chicago, IL, États-Unis

Réseau de l'utilisateur final (ASN) : AS7018

AWS Emplacement du service : Région de l'Est des États-Unis (Virginie du Nord)

Pour afficher les journaux, procédez comme suit :

1. Dans le AWS Management Console, accédez à CloudWatch Logs Insights.
2. Pour Log Group (Groupe de journaux), sélectionnez `/aws/internet-monitor/`*`monitorName`*`/byCity` et `/aws/internet-monitor/`*`monitorName`*`/byCountry`, puis spécifiez une plage de temps.
3. Ajoutez la requête suivante, puis exécutez-la.

La requête renvoie toutes les données de performance pour les utilisateurs se connectant depuis AS7018 à Chicago, IL vers la région USA Est (Virginie du Nord) sur la plage de temps sélectionnée.

```
fields @timestamp,
internetHealth.availability.experienceScore as availabilityExperienceScore,
internetHealth.availability.percentageOfTotalTrafficImpacted as
 percentageOfTotalTrafficImpacted,
internetHealth.performance.experienceScore as performanceExperienceScore,
internetHealth.performance.roundTripTime.p50 as roundTripTimep50,
internetHealth.performance.roundTripTime.p90 as roundTripTimep90,
internetHealth.performance.roundTripTime.p95 as roundTripTimep95
 | filter clientLocation.country == `United States`
 and clientLocation.city == `Chicago`
 and serviceLocation == `us-east-1`
 and clientLocation.asn == 7018
```

Pour plus d'informations, consultez la section Analyse des données des CloudWatch journaux avec Logs Insights.

Utilisation de Contributor Insights avec Amazon CloudWatch Internet Monitor

CloudWatch Contributor Insights peut vous aider à identifier les principaux sites clients et réseaux (ASN ou fournisseurs de services Internet) pour votre application. Utilisez les exemples de règles Contributor Insights suivants pour commencer à utiliser des règles utiles avec Amazon CloudWatch Internet Monitor. Pour plus d'informations, consultez Création d'une règle Contributor Insights.

Pour en savoir plus sur la précision de l'emplacement des clients dans Moniteur Internet, consultez Geolocation information and accuracy in Internet Monitor.

> **Note**
>
> Le Moniteur Internet publie des données toutes les cinq minutes, donc après avoir configuré une règle Contributor Insights, vous devez ajuster la période à cinq minutes pour voir un graphique.

Afficher les principaux emplacements et ASN affectés par un impact sur la disponibilité

Pour afficher les principaux emplacements clients et ASN affectés par une baisse de disponibilité, vous pouvez utiliser la règle Contributor Insights suivante dans l'éditeur de syntaxe. Remplacez *monitor-name* par votre propre nom de moniteur.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
```

```
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.availability.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Afficher les principaux emplacements clients et ASN affectés par un impact sur la latence

Pour afficher les principaux emplacements clients et ASN affectés par une augmentation de temps de propagation aller et retour (latence), vous pouvez utiliser la règle Contributor Insights suivante dans l'éditeur de syntaxe. Remplacez *monitor-name* par votre propre nom de moniteur.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.awsInternetHealth.performance.percentageOfTotalTrafficImpacted"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Afficher les principaux emplacements clients et ASN concernés par le pourcentage total du trafic

Pour afficher les emplacements clients et les ASN concernés par le pourcentage total du trafic, vous pouvez utiliser la règle Contributor Insights suivante dans l'éditeur de syntaxe. Remplacez *monitor-name* par votre propre nom de moniteur.

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Sum",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.clientLocation.city",
                "IsPresent": true
            }
        ],
        "Keys": [
            "$.clientLocation.city",
            "$.clientLocation.networkName"
        ],
        "ValueOf": "$.percentageOfTotalTraffic"
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "/aws/internet-monitor/monitor-name/byCity"
    ]
}
```

Utilisation CloudWatch des métriques avec Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor publie des statistiques sur votre compte, notamment des mesures relatives aux performances, à la disponibilité, au temps d'aller-retour et au débit (octets par seconde), que vous pouvez consulter dans la section CloudWatch Mesures de la console. CloudWatch Pour trouver toutes les métriques de votre moniteur, dans le tableau de bord CloudWatch des métriques, consultez l'espace de noms `AWS/InternetMonitor` personnalisé.

Les métriques sont agrégées sur l'ensemble du trafic Internet vers vos VPC, vos équilibreurs de charge réseau, vos CloudFront distributions ou vos WorkSpaces répertoires sur le moniteur, ainsi que sur l'ensemble du trafic vers chaque Région AWS emplacement périphérique d'Internet surveillé. Les

régions sont définies par l'emplacement du service, qui peut être soit tous les emplacements, soit une région spécifique, telle que `us-east-1`.

Remarque : les réseaux urbains sont des emplacements clients et des ASN (généralement des fournisseurs de services Internet, ou FSI).

Moniteur Internet fournit les métriques suivantes.

Métrique	Description
PerformanceScore	Un score de performance représente le pourcentage estimé du trafic qui ne subit pas de baisse de performance.
AvailabilityScore	Un score de disponibilité représente le pourcentage estimé du trafic qui ne subit pas de baisse de disponibilité.
BytesIn	Octets transférés dans le trafic Internet de votre application sur tous les réseaux urbains de l'application.
BytesOut	Octets transférés hors du trafic Internet de votre application sur tous les réseaux urbains d'applications.
BytesInMonitored	Octets transférés dans le trafic Internet de votre application sur les réseaux urbains surveillés.
BytesOutMonitored	Octets transférés hors du trafic Internet de votre application sur les réseaux urbains surveillés.
Round-trip time (RTT)	Temps aller-retour entre les ASN (générale ment les Régions AWS fournisseurs de services Internet ou ISP) et les emplaceme nts (tels que les villes) spécifiques à vos VPC, équilibreurs de charge réseau, distributions ou annuaires. CloudFront WorkSpaces

Métrique	Description
CityNetworksMonitored	Nombre de réseaux urbains surveillés par Moniteur Internet pour le trafic Internet de votre application. Ce nombre n'est jamais plus élevé que la limite maximale de réseaux urbains que vous avez définie pour le moniteur.
TrafficMonitoredPercent	Pourcentage du trafic Internet total d'applica tion pour ce moniteur qui est représenté (inclus) par les réseaux urbains surveillés par Moniteur Internet. Ce chiffre est inférieur à 100 (c'est-à-dire moins de 100 %) si les clients accèdent à votre application dans un plus grand nombre de réseaux urbains que la limite maximale de réseaux urbains que vous avez définie pour le moniteur.
CityNetworksFor100 PercentTraffic	Chiffre auquel vous devez définir la limite maximale de réseaux urbains si vous souhaitez surveiller 100 % du trafic Internet de votre application dans Moniteur Internet.
CityNetworksFor99 PercentTraffic	Chiffre auquel vous devez définir la limite maximale de réseaux urbains si vous souhaitez surveiller 99 % du trafic Internet de votre application dans Moniteur Internet.
CityNetworksFor95 PercentTraffic	Chiffre auquel vous devez définir la limite maximale de réseaux urbains si vous souhaitez surveiller 95 % du trafic Internet de votre application dans Moniteur Internet.
CityNetworksFor90 PercentTraffic	Chiffre auquel vous devez définir la limite maximale de réseaux urbains si vous souhaitez surveiller 90 % du trafic Internet de votre application dans Moniteur Internet.

Métrique	Description
CityNetworksFor75 PercentTraffic	Chiffre auquel vous devez définir la limite maximale de réseaux urbains si vous souhaitez surveiller 75 % du trafic Internet de votre application dans Moniteur Internet.
CityNetworksFor50 PercentTraffic	Chiffre auquel vous devez définir la limite maximale de réseaux urbains si vous souhaitez surveiller 50 % du trafic Internet de votre application dans Moniteur Internet.
CityNetworksFor25 PercentTraffic	Chiffre auquel vous devez définir la limite maximale de réseaux urbains si vous souhaitez surveiller 25 % du trafic Internet de votre application dans Moniteur Internet.

> **Note**
>
> Pour voir des exemples d'utilisation de ces mesures et déterminer la valeur maximale de réseaux urbains pour votre moniteur, consultez Choosing a city-network maximum value.

Pour plus d’informations, consultez Utiliser les CloudWatch métriques Amazon.

Utiliser Amazon Athena pour interroger les mesures Internet dans les fichiers journaux Amazon S3

Vous pouvez utiliser Amazon Athena pour interroger et consulter les mesures Internet publiées par Amazon CloudWatch Internet Monitor dans un compartiment Amazon S3. Moniteur Internet propose une option permettant de publier les mesures Internet de votre application dans un compartiment S3 pour le trafic Internet des réseaux urbains que vous surveillez (emplacements clients et ASN, généralement des fournisseurs de services Internet, FSI). Que vous choisissiez ou non de publier les mesures sur S3, Internet Monitor publie automatiquement les mesures Internet dans CloudWatch Logs toutes les cinq minutes pour les 500 principaux réseaux urbains (en termes de volume de trafic) pour chaque moniteur.

Ce chapitre explique comment créer une table dans Athena pour les mesures Internet situées dans un fichier journal S3, puis fournit des exemples de requêtes pour voir les mesures de différentes façons. Par exemple, vous pouvez rechercher les 10 réseaux urbains les plus affectés par un impact sur la latence.

Utiliser Amazon Athena pour créer une table des mesures Internet dans Moniteur Internet

Pour commencer à utiliser Athena avec les fichiers journaux S3 de Moniteur Internet, vous devez d'abord créer une table des mesures Internet.

Suivez les étapes de cette procédure pour créer une table dans Athena à partir des fichiers journaux S3. Vous pouvez ensuite exécuter des requêtes Athena sur la table, comme ces exemples de requêtes de mesures Internet, pour obtenir des informations sur vos mesures.

Pour créer une table Athéna

1. Ouvrez la console à l'adresse https://console.aws.amazon.com/athena/.
2. Dans l'éditeur de requêtes Athena, saisissez une instruction de requête pour générer une table contenant les mesures Internet Moniteur Internet. Remplacez la valeur du paramètre LOCATION par l'emplacement du compartiment S3 dans lequel sont stockées les mesures Internet de Moniteur Internet.

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

3. Saisissez une instruction pour créer une partition afin de lire les données. Par exemple, la requête suivante crée une partition unique pour une date et un emplacement spécifiques :

```
ALTER TABLE internet_measurements
ADD PARTITION (year = 'YYYY', month = 'MM', day = 'dd')
LOCATION
's3://bucket_name/bucket_prefix/AWSLogs/account_id/internetmonitor/AWS_Region/YYYY/
MM/DD';
```

4. Cliquez sur Exécuter.

Exemples d'instructions Athéna pour les mesures Internet

Voici un exemple d'instruction permettant de générer une table :

```
CREATE EXTERNAL TABLE internet_measurements (
    version INT,
    timestamp INT,
    clientlocation STRING,
    servicelocation STRING,
    percentageoftotaltraffic DOUBLE,
    bytesin INT,
    bytesout INT,
    clientconnectioncount INT,
    internethealth STRING,
    trafficinsights STRING
)
PARTITIONED BY (year STRING, month STRING, day STRING)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/'
TBLPROPERTIES ('skip.header.line.count' = '1');
```

Voici un exemple d'instruction permettant de créer une partition pour lire les données :

```
ALTER TABLE internet_measurements
ADD PARTITION (year = '2023', month = '04', day = '07')
LOCATION 's3://internet-measurements/TestMonitor/AWSLogs/1111222233332/internetmonitor/
us-east-2/2023/04/07/'
```

Exemples de requêtes Amazon Athena à utiliser avec les mesures Internet dans Moniteur Internet

Cette section inclut des exemples de requêtes que vous pouvez utiliser avec Amazon Athena pour obtenir des informations sur les mesures Internet de votre application publiées sur Amazon S3.

Interroger les 10 emplacements clients et ASN les plus affectés (par le pourcentage total du trafic)

Exécutez cette requête Athena pour obtenir les 10 réseaux urbains les plus affectés (par le pourcentage total du trafic), c'est-à-dire les emplacements clients et les ASN, généralement des fournisseurs de services Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(percentageoftotaltraffic) as percentageoftotaltraffic
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageoftotaltraffic desc
limit 10
```

Rechercher les 10 emplacements clients et ASN les plus affectés (par la disponibilité)

Exécutez cette requête Athena pour obtenir les 10 réseaux urbains les plus affectés (par le pourcentage total du trafic), c'est-à-dire les emplacements clients et les ASN, généralement des fournisseurs de services Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.availability.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Rechercher les 10 emplacements clients et ASN les plus affectés (par la latence)

Exécutez cette requête Athena pour obtenir les 10 réseaux urbains les plus affectés (par l'impact sur la latence), c'est-à-dire les emplacements clients et les ASN, généralement des fournisseurs de services Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.networkname') as networkName,
    sum(
        cast(
            json_extract_scalar(
                internetHealth,
                '$.performance.percentageoftotaltrafficimpacted'
            )
        as double )
    ) as percentageOfTotalTrafficImpacted
FROM internet_measurements
GROUP BY json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.networkname')
ORDER BY percentageOfTotalTrafficImpacted desc
limit 10
```

Interroger les informations sur le trafic pour les emplacements clients et les ASN

Exécutez cette requête Athena pour obtenir des informations sur le trafic, notamment le score de disponibilité, le score de performance et le temps jusqu'au premier octet pour vos réseaux urbains, c'est-à-dire les emplacements clients et les ASN, généralement des fournisseurs de services Internet.

```
SELECT json_extract_scalar(clientLocation, '$.city') as city,
    json_extract_scalar(clientLocation, '$.subdivision') as subdivision,
    json_extract_scalar(clientLocation, '$.country') as country,
    avg(cast(json_extract_scalar(internetHealth, '$.availability.experiencescore') as
 double)) as availabilityScore,
    avg(cast(json_extract_scalar(internetHealth, '$.performance.experiencescore') as
 double)) performanceScore,
    avg(cast(json_extract_scalar(trafficinsights,
 '$.timetofirstbyte.currentexperience.value') as double)) as averageTTFB,
    sum(bytesIn) as bytesIn,
    sum(bytesOut) as bytesOut,
    sum(bytesIn + bytesOut) as totalBytes
FROM internet_measurements
where json_extract_scalar(clientLocation, '$.city') != 'N/A'
GROUP BY
json_extract_scalar(clientLocation, '$.city'),
    json_extract_scalar(clientLocation, '$.subdivision'),
    json_extract_scalar(clientLocation, '$.country')
ORDER BY totalBytes desc
```

```
limit 100
```

Pour plus d'informations sur l'utilisation d'Athena, consultez le Guide de l'utilisateur Amazon Athena.

Utilisation de l'interface de requête Amazon CloudWatch Internet Monitor

Pour mieux comprendre le trafic Internet de votre AWS application, vous pouvez utiliser l'interface de requête Amazon CloudWatch Internet Monitor. Pour utiliser l'interface de requête, vous créez une requête avec les filtres de données de votre choix, puis vous exécutez la requête pour renvoyer un sous-ensemble de vos données du Moniteur Internet. L'exploration des données renvoyées par la requête peut vous donner un aperçu des performances de votre application sur Internet.

Vous pouvez interroger et explorer toutes les mesures capturées par le Moniteur Internet avec votre moniteur, notamment les scores de disponibilité et de performance, les octets transférés, les temps d'aller-retour et le temps écoulé jusqu'au premier octet (TTFB).

Le Moniteur Internet utilise l'interface de requête pour fournir les données que vous pouvez explorer dans le tableau de bord de la console du Moniteur Internet. En utilisant les options de recherche du tableau de bord (dans l'onglet Explorateur historique ou dans l'onglet Informations sur le trafic), vous pouvez interroger et filtrer les données Internet pour votre application.

Si vous souhaitez explorer et filtrer vos données avec plus de flexibilité que celle fournie par le tableau de bord, vous pouvez utiliser vous-même l'interface de requête, en utilisant les opérations de l'API Internet Monitor avec le AWS Command Line Interface ou avec un AWS SDK. Cette section présente les types de requêtes que vous pouvez utiliser avec l'interface de requête, ainsi que les filtres que vous pouvez spécifier pour créer un sous-ensemble de données, afin d'obtenir des informations sur le trafic Internet de votre application.

Rubriques

Comment utiliser l'interface de requête

Vous créez une requête à l'aide de l'interface de requête en choisissant un type de requête, puis en spécifiant des valeurs de filtre, pour renvoyer un sous-ensemble spécifique souhaité des données

de votre fichier journal. Vous pouvez ensuite utiliser le sous-ensemble de données pour filtrer et trier davantage, créer des rapports, etc.

Le processus de requête fonctionne de la manière suivante :

1. Lorsque vous exécutez une requête, le Moniteur Internet renvoie un `query ID` qui est unique à la requête. Cette section décrit les types de requêtes disponibles et les options de filtrage des données dans les requêtes. Pour comprendre comment cela fonctionne, vous pouvez également consulter la section sur les exemples de requêtes.
2. Vous spécifiez l'ID de requête avec le nom de votre moniteur lors de l'opération d'GetQueryResultsAPI pour renvoyer les résultats de la requête. Chaque type de requête renvoie un ensemble de champs de données différent. Pour en savoir plus, veuillez consulter Obtention des résultats de requêtes.

L'interface de requête fournit les trois types de requêtes suivants. Chaque type de requête renvoie un ensemble d'informations différent à propos de votre trafic à partir des fichiers journaux, comme indiqué.

- Mesures : fournit le score de disponibilité, le score de performance, le trafic total et les temps d'aller-retour, à des intervalles de 5 minutes.
- Principaux emplacements : fournit le score de disponibilité, le score de performance, le trafic total et les informations sur le délai d'obtention du premier octet (TTFB), pour les meilleures combinaisons d'emplacements et d'ASN que vous surveillez, par volume de trafic.
- Informations sur les principaux sites : fournit le TTFB pour Amazon CloudFront, votre configuration actuelle et la configuration Amazon EC2 la plus performante, à intervalles d'une heure.

Avec chacun de ces types de requêtes, vous pouvez filtrer davantage les données en spécifiant un ou plusieurs des critères suivants :

- AWS emplacement : pour AWS l'emplacement, vous pouvez spécifier CloudFront ou un Région AWS`us-east-2`, `us-west-2`, tel que,, etc.
- ASN : spécifiez un ASN, qui est généralement un fournisseur de services Internet (FSI).
- Emplacement du client : pour l'emplacement, spécifiez une ville, une métropole, un département ou un pays.
- Geo : spécifiez `geo` pour certaines requêtes. Cela est obligatoire pour les requêtes qui utilisent le type de requête `Top locations`, mais n'est pas autorisé pour les autres types de requête. Pour

savoir quand spécifier `geo` pour les paramètres du filtre, veuillez consulter la section des exemples de requêtes.

Les opérateurs que vous pouvez utiliser pour filtrer vos données sont `EQUALS` et `NOT_EQUALS`. Pour plus de détails sur les paramètres de filtrage, consultez le fonctionnement de l'FilterParameterAPI.

Pour en savoir plus sur les opérations de l'interface de requête, consultez les opérations d'API suivantes dans le guide de référence des API Amazon CloudWatch Internet Monitor :

- Pour créer et exécuter une requête, consultez le fonctionnement de l'StartQueryAPI.
- Pour arrêter une requête, consultez le fonctionnement de StopQueryl'API.
- Pour renvoyer des données pour une requête que vous avez créée, consultez le fonctionnement de l'GetQueryResultsAPI.
- Pour récupérer le statut d'une requête, consultez le fonctionnement de l'GetQueryStatusAPI.

Exemples de requêtes

Pour créer une requête que vous pouvez utiliser pour récupérer un ensemble de données filtré à partir du fichier journal de votre moniteur, vous utilisez l'opération StartQueryAPI. Vous spécifiez un type de requête et des paramètres de filtre pour la requête. Ensuite, lorsque vous utilisez l'opération d'API de l'interface de requête du Moniteur Internet pour obtenir les résultats d'une requête à l'aide de la requête, elle récupère le sous-ensemble de données avec lequel vous souhaitez travailler.

Pour illustrer le fonctionnement des types de requêtes et des paramètres de filtre, examinons quelques exemples.

Exemple 1

Supposons que vous souhaitiez récupérer toutes les données du fichier journal de votre moniteur pour un pays spécifique, à l'exception d'une ville. L'exemple suivant montre les paramètres de filtre pour une requête que vous pourriez créer avec l'opération `StartQuery` correspondant à ce scénario.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "MEASUREMENTS"
   FilterParameters: [
```

```
      {
       Field: "country",
       Operator: "EQUALS",
       Values: ["Germany"]
      },
      {
       Field: "city",
       Operator: "NOT_EQUALS",
       Values: ["Berlin"]
      },
    ]
}
```

Exemple 2

Autre exemple, supposons que vous souhaitiez afficher les emplacements les plus importants par zone métropolitaine. Vous pouvez utiliser l'exemple de requête suivant pour ce scénario.

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["metro"]
      },
    ]
}
```

Exemple 3

Supposons maintenant que vous souhaitiez connaître les meilleures combinaisons de réseaux urbains de la région métropolitaine de Los Angeles. Pour ce faire, spécifiez `geo=city`, puis définissez `metro` à Los Angeles. Désormais, la requête renvoie les principaux réseaux urbains de la région métropolitaine de Los Angeles au lieu des principaux réseaux métropolitains leur ensemble.

Voici l'exemple de requête que vous pourriez utiliser :

```
{
   MonitorName: "TestMonitor"
```

```
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATIONS"
   FilterParameters: [
      {
       Field: "geo",
       Operator: "EQUALS",
       Values: ["city"]
      },
      {
       Field: "metro",
       Operator: "EQUALS",
       Values: ["Los Angeles"]
      }
    ]
}
```

Exemple 4

Enfin, supposons que vous souhaitiez récupérer des données TTFB pour une subdivision spécifique (par exemple, un État américain).

Voici un exemple de requête pour ce scénario :

```
{
   MonitorName: "TestMonitor"
   StartTime: "2023-07-12T20:00:00Z"
   EndTime: "2023-07-12T21:00:00Z"
   QueryType: "TOP_LOCATION_DETAILS"
   FilterParameters: [
      {
       Field: "subdivision",
       Operator: "EQUALS",
       Values: ["California"]
      },
    ]
}
```

Obtention des résultats de requêtes

Après avoir défini une requête, vous pouvez renvoyer un ensemble de résultats avec la requête en exécutant une autre opération d'API Internet Monitor, GetQueryResults. Lorsque vous exécutez `GetQueryResults`, vous spécifiez l'ID de requête pour la requête que vous avez définie, ainsi que

le nom de votre moniteur. `GetQueryResults` extrait les données de la requête spécifiée dans un jeu de résultats.

Lorsque vous exécutez une requête, assurez-vous que l'exécution de la requête est terminée avant d'utiliser `GetQueryResults` pour consulter les résultats. Vous pouvez déterminer si la requête est terminée à l'aide de l'opération GetQueryStatusAPI. Lorsque le `Status` de la requête est `SUCCEEDED`, vous pouvez procéder à la consultation des résultats.

Lorsque votre requête est terminée, vous pouvez suivre ces conseils pour consulter les résultats. Chaque type de requête que vous utilisez pour créer une requête inclut un ensemble unique de champs de données provenant des fichiers journaux, comme décrit dans la liste suivante :

Mesures

Le type de requête `measurements` renvoie les données suivantes :

`timestamp, availability, performance, bytes_in, bytes_out, rtt_p50, rtt_p90, rtt_p95`

Meilleurs emplacements

Le type de requête `top locations` regroupe les données par emplacement et fournit la moyenne des données sur la période. Les données qu'il renvoie comprennent les éléments suivants :

`aws_location, city, metro, subdivision, country, asn, availability, performance, bytes_in, bytes_out, current_fbl, best_ec2, best_ec2_region, best_cf_fbl`

Notez que `city`, `metro` et `subdivision` ne sont renvoyés que si vous choisissez ce type d'emplacement pour le champ `geo`. Les champs d'emplacement suivants sont renvoyés, en fonction du type d'emplacement que vous spécifiez `geo` :

```
city = city, metro, subdivision, country
metro = metro, subdivision, country
subdivision = subdivision, country
country = country
```

Détails sur les meilleurs emplacements

Le type de requête `top locations details` renvoie des données groupées heure par heure. La requête renvoie les données suivantes :

```
timestamp, current_service, current_fbl, best_ec2_fbl, best_ec2_region,
best_cf_fbl
```

Lorsque vous exécutez l'opération d'API `GetQueryResults`, le Moniteur Internet renvoie ce qui suit dans la réponse :

- Un tableau de chaînes de données contenant les résultats renvoyés par la requête. Les informations sont renvoyées dans des tableaux alignés sur le champ `Fields`, également renvoyés par l'appel d'API. À l'aide du champ `Fields`, vous pouvez analyser les informations du référentiel `Data`, puis les filtrer ou les trier en fonction de vos besoins.
- Un tableau de champs répertoriant les champs pour lesquels la requête a renvoyé des données (dans la réponse au champ `Data`). Chaque élément du tableau est une paire nom-type de données, telle que `availability_score-float`.

Résolution des problèmes

Si des erreurs sont renvoyées lorsque vous utilisez les opérations de l'API de l'interface de requête, vérifiez que vous disposez des autorisations requises pour utiliser Amazon CloudWatch Internet Monitor. Plus précisément, assurez-vous que vous disposez des autorisations suivantes :

```
internetmonitor:StartQuery
internetmonitor:GetQueryStatus
internetmonitor:GetQueryResults
internetmonitor:StopQuery
```

Ces autorisations sont incluses dans la AWS Identity and Access Management politique recommandée pour utiliser le tableau de bord Internet Monitor dans la console. Pour plus d'informations, consultez Autorisations IAM pour Amazon CloudWatch Internet Monitor.

Création d'alarmes avec Amazon CloudWatch Internet Monitor

Vous pouvez créer des CloudWatch alarmes Amazon en fonction des métriques Amazon CloudWatch Internet Monitor, comme vous le pouvez pour les autres CloudWatch métriques Amazon.

Par exemple, vous pouvez créer une alerte basée sur la métrique `PerformanceScore` Moniteur Internet et la configurer pour envoyer une notification lorsque la métrique est inférieure à la valeur que vous avez choisie. Vous configurez les alarmes pour les métriques Internet Monitor en suivant les mêmes directives que pour les autres CloudWatch métriques.

Voici des exemples de métriques Moniteur Internet pour lesquelles vous pouvez créer une alerte :

- PerformanceScore
- AvailabilityScore
- RoundtripTime

Pour voir toutes les métriques disponibles pour Moniteur Internet, consultez Utilisation CloudWatch des métriques avec Amazon CloudWatch Internet Monitor.

La procédure suivante fournit un exemple d'activation d'une alarme PerformanceScoreen accédant à la métrique dans le CloudWatch tableau de bord. Ensuite, vous suivez les CloudWatch étapes standard pour créer une alarme en fonction d'un seuil que vous choisissez, puis vous configurez une notification ou choisissez d'autres options.

Pour créer une alarme PerformanceScoredans CloudWatch Metrics

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Métriques, puis Toutes les métriques.
3. Filtrez pour Moniteur Internet en choisissant `AWS/InternetMonitor`.
4. Choisissez MeasurementSource, MonitorName.
5. Dans la liste, sélectionnez PerformanceScore.
6. GraphedMetricsDans l'onglet, sous Actions, choisissez l'icône en forme de cloche pour créer une alarme basée sur un seuil statique.

Suivez maintenant les CloudWatch étapes standard pour choisir les options de l'alarme.
Par exemple, vous pouvez choisir d'être averti par un message Amazon SNS si le chiffre PerformanceScoreest inférieur à un certain seuil. Vous pouvez également, à la place ou en complément, ajouter l'alerte à un tableau de bord.

Gardez à l’esprit les points suivants :

- Les métriques du Moniteur Internet sont généralement calculées et publiées en 20 minutes.
- Lorsque vous créez une alarme basée sur les métriques du Moniteur Internet, assurez-vous de prendre en compte le court délai avant la publication lorsque vous définissez la période de rétrospective d'une alarme. Nous vous recommandons de configurer les Périodes d'évaluation avec une période rétrospective d'au moins 25 minutes.

Pour en savoir plus sur l'utilisation des CloudWatch alarmes avec Internet Monitor, consultez le billet de blog suivant : Utilisation d'Amazon CloudWatch Internet Monitor pour une meilleure observabilité sur Internet.

Pour plus d'informations sur les options disponibles lors de la création CloudWatch d'une alarme, consultezCréation d'une CloudWatch alarme basée sur un seuil statique.

Utilisation d'Amazon CloudWatch Internet Monitor avec Amazon EventBridge

Les événements de santé créés par Amazon CloudWatch Internet Monitor pour des problèmes de réseau sont publiés sur Amazon EventBridge, afin que vous puissiez envoyer des notifications en cas de dégradation de l'expérience utilisateur pour votre application.

EventBridge Pour utiliser Internet Monitor Health Events, suivez les instructions ici.

Pour configurer une règle pour Internet Monitor dans EventBridge

1. Dans AWS Management Console, dans EventBridge, choisissez Règles, puis entrez un nom et une description. Créez la règle sur le bus d'événements Default (Par défaut).
2. À l'étape 2, sélectionnez Autre comme source d'événement, puis, sous Modèle d'événement, faites correspondre la source suivante.

   ```
   {
     "source": ["aws.internetmonitor"]
   }
   ```

3. À l'étape 3, pour la cible, sélectionnez AWS Service et groupe de CloudWatch journaux, puis sélectionnez un groupe de journaux existant ou créez-en un nouveau.
4. Ajoutez les balises souhaitées, puis créez la règle. Cela devrait remplir le groupe de CloudWatch journaux sélectionné avec des événements provenant de EventBridge.

Pour plus d'informations sur le fonctionnement EventBridge des règles avec les modèles d'événements, consultez la section Modèles EventBridge d'événements Amazon dans le guide de EventBridge l'utilisateur Amazon.

Résoudre les erreurs CloudWatch d'accès aux journaux et aux métriques

Pour prendre en charge certaines fonctionnalités, Amazon CloudWatch Internet Monitor doit interagir avec certaines CloudWatch ressources Amazon, notamment les journaux et les statistiques. Si Internet Monitor ne peut pas accéder aux CloudWatch ressources auxquelles il doit accéder, Internet Monitor définit un code d'état `FAULT_ACCESS_CLOUDWATCH` pour le moniteur.

Il existe plusieurs raisons pour lesquelles votre moniteur peut présenter l'état `FAULT_ACCESS_CLOUDWATCH`. Les sections suivantes répertorient les causes possibles de ces erreurs et proposent des étapes de résolution des problèmes.

Internet Monitor n'a pas pu accéder aux CloudWatch journaux de votre compte

Le Moniteur Internet publie des journaux de diagnostic concernant le trafic des applications suivi par votre moniteur. Il publie ces journaux dans des groupes de CloudWatch journaux dans Logs à l'emplacement suivant :`/aws/internet-monitor/`*`monitor_name`*`/[byCity|byMetro|bySubdivision|byCountry]`. Internet Monitor n'a pas pu accéder à ces groupes de journaux.

États d'erreur et solutions potentielles :

- PutLogEvents erreur de limitation : le service Internet Monitor a peut-être été limité lorsqu'il a essayé de publier les journaux de votre moniteur sur. CloudWatch Passez en revue les limites de limitation de votre compte et, si nécessaire, demandez une augmentation de la limite.
- Groupe de journaux introuvable : désactivez puis réactivez votre moniteur. L'activation d'un moniteur redémarre la création de groupes de journaux, ce qui peut corriger le problème.
- PutLogEvents erreur d'accès refusé : contactez le AWS support pour obtenir de l'aide.
- PutLogEvents erreur inconnue ou générale : contactez le AWS support pour obtenir de l'aide.

Internet Monitor n'a pas pu accéder aux CloudWatch statistiques de votre compte

Internet Monitor fournit des CloudWatch mesures spécifiques concernant le trafic des applications qui est suivi par un moniteur. Une erreur s'est produite lorsqu'Internet Monitor a essayé de transmettre ces mesures à CloudWatch.

États d'erreur et solutions potentielles :

- PutMetricData erreur de limitation : le service Internet Monitor a peut-être été limité lorsqu'il a essayé de publier les statistiques de votre moniteur sur. CloudWatch Passez en revue les limites de limitation de votre compte et, si nécessaire, demandez une augmentation de la limite.

- PutMetricData erreur d'accès refusé : contactez le AWS support pour obtenir de l'aide.
- PutMetricData erreur inconnue ou générale : contactez le AWS support pour obtenir de l'aide.

Protection et confidentialité des données avec Amazon CloudWatch Internet Monitor

Le modèle de responsabilité AWS partagée s'applique à la protection et à la confidentialité des données dans Amazon CloudWatch Internet Monitor. Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure globale qui gère l'ensemble du AWS cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Pour plus d'informations sur la confidentialité des données, consultez la FAQ sur la confidentialité des données. Pour plus d'informations sur la protection des données en Europe, consultez le modèle de responsabilité AWS partagée et le billet de blog sur le RGPD sur le blog sur la AWS sécurité. Pour plus de ressources relatives à la conformité aux exigences du RGPD, consultez le Centre du Règlement général sur la protection des données (RGPD).

Nous vous recommandons vivement de ne jamais placer d'informations d'identification sensibles, telles que les numéros de compte, les adresses e-mail ou toute autre information personnelle de vos utilisateurs finaux, dans des champs à structure libre. Toutes les données que vous entrez dans Amazon CloudWatch Internet Monitor ou dans d'autres services peuvent être incluses dans les journaux de diagnostic.

Identity and Access Management pour Amazon CloudWatch Internet Monitor

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Des administrateurs IAM contrôlent les personnes qui s'authentifient (sont connectées) et sont autorisées (disposent d'autorisations) à utiliser des ressources Moniteur Internet. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

> **Important**
>
> Mises à jour relatives aux ressources Moniteur Internet le 24 février 2023

Si vous avez créé des politiques IAM incluant des ressources Moniteur Internet avant le 24 février 2023, prenez note des mises à jour suivantes relatives aux ressources et aux types de ressources Moniteur Internet.

- HealthEventsla ressource a été renommée en HealthEvent.
- Les formats ARN et Regex de la HealthEventressource ont été mis à jour.
- Les formats ARN et Regex de la ressource Monitor ont été mis à jour.
- Les autorisations au niveau des ressources pour l'GetHealthEventaction ne sont désormais prises en charge que pour le type de HealthEventressource. Elles ne sont pas prises en charge sur la ressource Monitor.
- TagResourceUntagResource, et ListTagsForResourcepour le type de ressource Monitor ont été mis à jour pour être obligatoires.

Pour plus d'informations sur les actions, les ressources et les clés de condition que vous pouvez spécifier dans les politiques pour gérer l'accès aux AWS ressources dans Internet Monitor, consultez Actions, ressources et clés de condition pour Amazon CloudWatch Internet Monitor.

Table des matières

Comment Amazon CloudWatch Internet Monitor fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à Moniteur Internet, découvrez les fonctions IAM que vous pouvez utiliser avec Moniteur Internet.

Pour consulter des tableaux présentant une vue d'ensemble similaire du fonctionnement des AWS services avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon CloudWatch Internet Monitor

Fonction IAM	Support Moniteur Internet
Politiques basées sur l’identité	Oui
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACL	Non
ABAC (identifications dans les politiques)	Partielle
Informations d’identification temporaires	Oui
Autorisations de principal	Oui
Fonctions du service	Non
Rôles liés à un service	Oui

Politiques basées sur l'identité pour Moniteur Internet

Prend en charge les politiques basées sur l’identité	Oui

Les politiques basées sur l’identité sont des documents de politique d’autorisations JSON que vous pouvez attacher à une identité telle qu’un utilisateur, un groupe d’utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l’identité, consultez Création de politiques IAM dans le Guide de l’utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources dans Moniteur Internet

Prend en charge les politiques basées sur les ressources	Non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique.

Actions de politique pour Moniteur Internet

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions d'Internet Monitor, consultez la section Actions définies par Amazon CloudWatch Internet Monitor dans le Service Authorization Reference.

Les actions de politique dans Moniteur Internet utilisent le préfixe suivant avant l'action :

```
internetmonitor
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "internetmonitor:action1",
      "internetmonitor:action2"
         ]
```

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante :

```
"Action": "internetmonitor:Describe*"
```

Ressources de politique pour Moniteur Internet

Prend en charge les ressources de politique	Oui

Dans la Référence de l'autorisation de service, vous pouvez consulter les informations suivantes relatives à Moniteur Internet :

- Pour consulter la liste des types de ressources Internet Monitor et leurs ARN, consultez la section Ressources définies par Amazon CloudWatch Internet Monitor.
- Pour en savoir plus sur les actions que vous pouvez spécifier avec l'ARN de chaque ressource, consultez la section Actions définies par Amazon CloudWatch Internet Monitor.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir

une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Clés de condition politique pour Moniteur Internet

Prend en charge les clés de condition de politique spécifiques au service	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition d'Internet Monitor, consultez la section Clés de condition pour Amazon CloudWatch Internet Monitor dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon CloudWatch Internet Monitor.

ACL dans Moniteur Internet

Prend en charge les listes ACL	Non

Les listes de contrôle d’accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l’autorisation d’accéder à une ressource. Les listes de contrôle d’accès sont similaires aux politiques basées sur les ressources, bien qu’elles n’utilisent pas le format de document de politique JSON.

ABAC avec Moniteur Internet

Prise en charge d’ABAC (identifications dans les politiques)	Partielle

Moniteur Internet prend partiellement en charge les balises dans les politiques. Il prend en charge le balisage pour une ressource, les moniteurs.

Pour utiliser des balises avec Internet Monitor, utilisez le AWS Command Line Interface ou un AWS SDK. Le balisage pour Internet Monitor n'est pas pris en charge avec le AWS Management Console.

Pour en savoir plus sur l'utilisation des balises dans les politiques en général, consultez les informations suivantes.

Le contrôle d’accès par attributs (ABAC) est une stratégie d’autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L’étiquetage des entités et des ressources est la première étape d’ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l’identification du principal correspond à celle de la ressource à laquelle il tente d’accéder.

L’ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l’accès basé sur des étiquettes, vous devez fournir les informations d’étiquette dans l’élément de condition d’une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d’informations sur l’ABAC, consultez Qu’est-ce que le contrôle d’accès basé sur les attributs (ABAC) ? dans le Guide de l’utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l’ABAC, consultez Utilisation du contrôle d’accès par attributs (ABAC) dans le Guide de l’utilisateur IAM.

Utiliser les informations d'identification temporaires avec Moniteur Internet

Prend en charge les informations d’identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d’identification temporaires lorsque vous vous connectez à la console en tant qu’utilisateur, puis changez de rôle. Pour plus d’informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l’utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d’informations, consultez Informations d’identification de sécurité temporaires dans IAM.

Autorisations de principal entre services pour Moniteur Internet

Prend en charge les sessions d'accès direct (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d’autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d’accès.

Fonctions de service pour Moniteur Internet

Prend en charge les fonctions de service	Non

Une fonction du service est un rôle IAM qu’un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d’IAM. Pour plus d’informations, consultez Création d’un rôle pour la délégation d’autorisations à un Service AWS dans le Guide de l’utilisateur IAM.

Rôle lié à un service pour Moniteur Internet

Prend en charge les rôles liés à un service.	Oui

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d’effectuer une action en votre nom. Les rôles liés au service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour en savoir plus sur le rôle lié à service pour Moniteur Internet, consultez Rôle lié à un service pour Amazon Internet Monitor CloudWatch .

Pour plus de détails sur la création ou la gestion des rôles liés à un service en général dans AWS, consultez la section AWS Services qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un Yes dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

AWS politiques gérées pour Amazon CloudWatch Internet Monitor

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d’utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle politique Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d’informations, consultez la section Politiques gérées par AWS dans le Guide de l’utilisateur IAM.

AWS politique gérée : CloudWatchInternetMonitorServiceRolePolicy

Cette politique est associée au rôle lié au service nommé AWSServiceRoleForInternetMonitorpour permettre à Internet Monitor d'accéder aux ressources de votre compte, telles que les ressources Amazon Virtual Private Cloud ou les Network Load Balancers, afin que vous puissiez les sélectionner lorsque vous créez un moniteur. Pour plus d’informations, consultez Rôle lié à un service pour Amazon Internet Monitor CloudWatch .

Autorisations IAM pour Amazon CloudWatch Internet Monitor

Pour accéder aux actions permettant d'utiliser des moniteurs et des données dans Amazon CloudWatch Internet Monitor, les utilisateurs doivent disposer des autorisations appropriées.

Pour plus d'informations sur la sécurité sur Amazon CloudWatch, consultezGestion des identités et des accès pour Amazon CloudWatch.

Autorisations d'accès en lecture seule dans Amazon Internet Monitor CloudWatch

Pour accéder aux actions en lecture seule permettant d'utiliser les moniteurs et les données d'Amazon CloudWatch Internet Monitor, les utilisateurs doivent être connectés en tant qu'utilisateur ou en tant que rôle disposant des autorisations suivantes :

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

Autorisations pour un accès complet à Amazon CloudWatch Internet Monitor

Pour créer un moniteur dans Amazon CloudWatch Internet Monitor et bénéficier d'un accès complet aux actions permettant d'utiliser les moniteurs et les données dans Internet Monitor, les utilisateurs doivent être connectés avec un utilisateur ou un rôle disposant des autorisations suivantes :

- Autorisations pour créer un rôle lié à un service associé au Moniteur Internet. Pour plus d'informations, consultez Rôle lié à un service pour Amazon Internet Monitor CloudWatch .
- Autorisations relatives aux actions qui permettent un accès complet pour utiliser les moniteurs et les données dans le Moniteur Internet.

> **Note**
>
> Si vous créez une politique d'autorisations basée sur l'identité qui est plus restrictive, les utilisateurs tributaires de cette politique risquent de ne pas disposer d'un accès complet pour créer et utiliser des moniteurs et des données dans le Moniteur Internet.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "internetmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "internetmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
internetmonitor.amazonaws.com/AWSServiceRoleForInternetMonitor"
        },
        {
            "Action": [
                "ec2:DescribeVpcs",
                "elasticloadbalancing:DescribeLoadBalancers",
                "workspaces:DescribeWorkspaceDirectories",
```

```
                "cloudfront:GetDistribution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Rôle lié à un service pour Amazon Internet Monitor CloudWatch

Amazon CloudWatch Internet Monitor utilise un rôle lié à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est un type unique de rôle IAM qui est lié directement au Moniteur Internet. Le rôle lié au service est prédéfini par Internet Monitor et inclut toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Le Moniteur Internet définit les autorisations du rôle lié à un service, et sauf définition contraire, seul Moniteur Internet peut assumer le rôle. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous ne pouvez supprimer le rôle qu'après avoir d'abord supprimé ses ressources liées. Cette restriction protège vos ressources de Moniteur Internet, car vous ne pouvez pas involontairement supprimer d'autorisations pour accéder aux ressources.

Pour plus d'informations sur les autres services prenant en charge les rôles liés à un service, consultez les services AWS opérationnels avec IAM et recherchez les services présentant la mention Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Choisissez un Yes (oui) ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations de rôle lié à un service pour Moniteur Internet

Internet Monitor utilise le rôle lié au service nommé. AWSServiceRoleForInternetMonitor Ce rôle permet à Internet Monitor d'accéder aux ressources de votre compte, telles que les ressources Amazon Virtual Private Cloud, les CloudFront distributions Amazon, les WorkSpaces annuaires Amazon et les Network Load Balancers, afin que vous puissiez les sélectionner lorsque vous créez un moniteur.

Ce rôle lié à un service utilise la politique gérée.
`CloudWatchInternetMonitorServiceRolePolicy`

Le rôle AWSServiceRoleForInternetMonitorlié à un service fait confiance au service suivant pour assumer le rôle :

- `internetmonitor.amazonaws.com`

Pour consulter les autorisations associées à cette politique, reportez-vous CloudWatchInternetMonitorServiceRolePolicyà la référence des politiques AWS gérées.

Création d'un rôle lié à un service pour Moniteur Internet

Vous n'avez pas besoin de créer manuellement le rôle lié à un service pour Moniteur Internet. La première fois que vous créez un moniteur, Internet Monitor le crée AWSServiceRoleForInternetMonitorpour vous.

Pour plus d'informations, consultez Création d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Modification d'un rôle lié à un service pour Moniteur Internet

Après que Moniteur Internet a créé un rôle lié à un service dans votre compte, vous ne pouvez pas modifier le nom du rôle car diverses entités pourraient faire référence au rôle. Vous pouvez modifier la description du rôle en utilisant IAM. Pour plus d'informations, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour Moniteur Internet

Si vous n'avez plus besoin d'utiliser une fonction 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 du rôle lié à un service avant de pouvoir les supprimer manuellement.

Après avoir retiré vos ressources de vos moniteurs dans Internet Monitor, puis supprimé les moniteurs, vous pouvez supprimer le rôle lié au service. AWSServiceRoleForInternetMonitor

> **Note**
>
> Si le service du Moniteur Internet utilise le rôle lorsque vous essayez de le supprimer, la suppression peut échouer. Si cela se produit, attendez quelques minutes puis réessayez.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, le AWS CLI, ou l' AWS API pour supprimer le rôle lié au AWSServiceRoleForInternetMonitorservice. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Mises à jour du rôle lié à un service de Moniteur Internet

Pour les mises à AWSServiceRoleForInternetMonitorjour de la stratégie AWS gérée pour le rôle lié au service Internet Monitor, voir CloudWatch mises à jour des politiques AWS gérées. Pour recevoir des alertes automatiques concernant les modifications de politique gérées dans CloudWatch, abonnez-vous au flux RSS sur la page Historique du CloudWatch document.

Quotas dans Amazon CloudWatch Internet Monitor

Amazon CloudWatch Internet Monitor applique les quotas suivants.

Ressource	Quota par défaut
Moniteurs par région	50
Ressources par moniteur	50
Jours pendant lesquels les événements de l'état résolu du Moniteur Internet sont conservés	400

Utilisation d'Amazon CloudWatch Network Monitor

Amazon CloudWatch Network Monitor fournit une visibilité sur les performances du réseau connectant vos applications AWS hébergées à vos destinations sur site et vous permet d'identifier la source de toute dégradation des performances du réseau en quelques minutes. Le Moniteur réseau est entièrement géré par AWS. Par conséquent, vous n'avez pas besoin d'installer d'agents supplémentaires pour surveiller les performances de votre réseau. Vous pouvez rapidement visualiser la perte de paquets et la latence de vos connexions réseau hybrides, définir des alertes et des seuils, puis prendre des mesures pour améliorer l'expérience réseau de vos utilisateurs finaux.

Le Moniteur réseau est destiné aux opérateurs réseau et aux développeurs d'applications qui souhaitent obtenir des informations en temps réel sur les performances du réseau.

Fonctions principales

- Utilisez le Moniteur réseau pour évaluer l'évolution de votre environnement réseau hybride à l'aide de mesures continues de perte de paquets et de latence en temps réel.
- Lorsque vous vous connectez en utilisant AWS Direct Connect, Network Monitor diagnostique rapidement la dégradation du réseau en écrivant le AWS Network Health Indicator sur votre CloudWatch compte. Cette métrique fournit un score probabiliste permettant de déterminer si la dégradation du réseau s'est produite au sein d' AWS.
- Le Moniteur réseau fournit une surveillance fluide grâce à une approche d'agent entièrement gérée, ce qui signifie que vous n'avez pas besoin d'installer d'agents sur des VPC ou sur site. Il vous suffit de spécifier un sous-réseau VPC et une adresse IP sur site pour commencer.
- Network Monitor publie les métriques dans CloudWatch Metrics. Vous pouvez créer des tableaux de bord pour consulter vos indicateurs et créer des seuils et des alarmes exploitables sur les indicateurs spécifiques à votre application.

Pour en savoir plus, consultez the section called "Fonctionnement du Moniteur réseau".

Terminologie et composants du Moniteur réseau

- Moniteur : un moniteur inclut les ressources d'une seule application pour laquelle vous souhaitez consulter les mesures de performance et de disponibilité Internet, et pour laquelle vous souhaitez recevoir des alertes relatives aux événements de santé. Lorsque vous créez un moniteur pour une application, vous ajoutez des ressources AWS hébergées en tant que source réseau. Network Monitor crée ensuite une liste de toutes les sondes possibles entre les ressources AWS hébergées et vos adresses IP de destination.
- Sondes : une sonde est le trafic envoyé depuis la ressource AWS hébergée vers votre adresse IP de destination sur site. Les métriques du moniteur réseau sont enregistrées dans votre CloudWatch compte pour chaque sonde configurée dans un moniteur.
- AWS source réseau : il s'agit de la AWS source d'origine d'une sonde de surveillance réseau, qui sera un sous-réseau de n'importe lequel de vos VPC.
- Destination : il s'agit de la cible de la source réseau dans votre réseau AWS sur site. La destination est une combinaison de vos adresses IP sur site, de vos protocoles réseau, de vos ports et de la taille de vos paquets réseau. Les protocoles IPv4 et IPv6 sont tous deux pris en charge.

Limitations et exigences du Moniteur réseau

- Le Moniteur réseau prend en charge un maximum de quatre adresses IP de destination et jusqu'à 24 sondes par moniteur.
- Vous pouvez avoir jusqu'à 100 moniteurs par compte et par région.
- Les sous-réseaux du moniteur doivent appartenir au même compte que le moniteur.
- Le Moniteur réseau ne fournit pas de basculement automatique du réseau en cas de problème AWS réseau.
- Chaque sonde que vous créez est facturée. Pour plus de détails sur les prix, veuillez consulter the section called "Tarification".

Comment fonctionne Amazon CloudWatch Network Monitor

Le Moniteur réseau facilite la surveillance en fournissant une solution entièrement gérée et sans agent. Lorsque vous créez un moniteur dans votre ressource AWS hébergée, vous AWS créez et gérez toute l'infrastructure en arrière-plan pour effectuer des mesures de temps aller-retour et de perte de paquets. Ainsi, vous pouvez rapidement étendre votre surveillance sans avoir à installer ou à désinstaller d'agents au sein de votre AWS infrastructure.

Network Monitor concentre la surveillance sur les itinéraires empruntés par les flux provenant de vos ressources AWS hébergées au lieu de surveiller globalement tous les flux provenant de vos ressources Région AWS. Si vos charges de travail sont réparties sur plusieurs zones de disponibilité (AZ), le Moniteur réseau peut surveiller les itinéraires depuis chacun de vos sous-réseaux privés.

Le Moniteur réseau publie les mesures relatives au temps de propagation aller et retour et aux pertes de paquets sur votre compte Amazon CloudWatch en fonction de l'intervalle d'agrégation défini lors de la création d'un moniteur. Vous pouvez également définir des seuils de latence et de perte de paquets individuels pour chaque moniteur utilisant CloudWatch. Par exemple, vous pouvez créer une alarme pour vous avertir si votre moyenne de pertes de paquets est supérieure au seuil statique de 0,1 % pour une charge de travail sensible aux pertes de paquets. Vous pouvez également utiliser la détection des CloudWatch anomalies pour déclencher une alarme en cas de perte de paquets ou de mesures de latence en dehors des plages souhaitées.

Mesures de disponibilité et de performance

Network Monitor envoie régulièrement des sondes actives depuis votre AWS ressource vers vos destinations sur site. Lorsque vous créez un moniteur, vous spécifiez ce qui suit :

- L’intervalle d’agrégation. Durée, en secondes, pendant laquelle les CloudWatch résultats mesurés sont reçus. Ce sera toutes les 30 ou 60 secondes. La période d’agrégation que vous choisissez pour le moniteur s’applique à toutes les sondes de ce moniteur.
- Le protocole de la sonde. Chaque sonde ajoutée à un moniteur doit utiliser le protocole ICMP (Internet Control Message Protocol) ou le protocole TCP (Transmission Control Protocol). Pour plus d’informations, consultez the section called “Protocoles de communication”.
- La taille de paquet. Taille, en octets, de chaque paquet transmis entre votre ressource AWS hébergée et votre destination sur une seule sonde. Chaque sonde d’un moniteur peut avoir sa propre taille de paquet.

Pour les métriques,

- La métrique du temps de propagation aller et retour, mesurée en millisecondes, mesure et enregistre une mesure des performances et enregistre le temps nécessaire pour que la sonde soit transmise à l’adresse IP de destination et pour que la réponse associée soit reçue.
- La métrique de perte de paquets mesure le pourcentage du total des paquets envoyés et enregistre le nombre de sondes transmises qui n’ont pas reçu de réponse associée, ce qui implique que ces paquets ont été effectivement perdus le long du chemin réseau.

Protocoles de communication pris en charge

Les sondes ICMP transportent les demandes d'écho ICMP provenant de vos ressources AWS hébergées vers l'adresse de destination et attendent une réponse d'écho ICMP depuis l'adresse de destination. Le Moniteur réseau utilise les informations relatives aux messages de demande d’écho et de réponse ICMP pour calculer le temps de propagation aller et retour et les mesures de perte de paquets.

Les sondes TCP transportent les paquets TCP SYN de vos ressources AWS hébergées vers l'adresse et le port de destination et attendent un paquet TCP SYN+ACK ou RST en retour depuis l'adresse et le port de destination. Le Moniteur réseau utilise les informations relatives aux messages TCP SYN et TCP SYN+ACK ou RST pour calculer le temps de propagation aller et retour et les mesures de perte de paquets. En outre, le Moniteur réseau change régulièrement de port TCP source pour augmenter la couverture réseau, ce qui peut augmenter la probabilité de détecter une perte de paquets.

AWS Indicateur de santé du réseau

Le Moniteur réseau publie un indicateur de santé du réseau (NHI, Network Health Indicator), qui fournit des informations sur les performances et la disponibilité du réseau pour les destinations connectées via AWS Direct Connect. La métrique est une mesure statistique de l'état du chemin réseau AWS contrôlé entre la ressource AWS hébergée, sur laquelle le moniteur est déployé, et l'emplacement Direct Connect.

Le Moniteur réseau utilise la détection des anomalies pour calculer les baisses de disponibilité ou la dégradation des performances le long des chemins de votre réseau.

> **Note**
>
> Chaque fois que vous créez un nouveau moniteur, que vous ajoutez une sonde ou que vous réactivez une sonde, le NHI correspondant à ce moniteur est retardé de quelques heures afin de permettre la AWS collecte des données nécessaires à la détection des anomalies.

Pour fournir la métrique NHI, le Moniteur réseau applique une corrélation statistique entre des échantillons de données AWS , ainsi qu'aux mesures de perte de paquets et de latence de propagation aller et retour pour le trafic simulant le chemin de votre réseau. La métrique peut être l'une des deux variables suivantes : 1 ou 0. La valeur 1 indique que Network Monitor a observé une dégradation du réseau dans le chemin réseau AWS contrôlé. La valeur 0 indique que le Moniteur réseau n'a observé aucune dégradation du réseau le long du chemin. Cela vous permet de résoudre les problèmes de réseau plus rapidement. Vous pouvez définir des alertes sur la métrique NHI pour être informé des problèmes récurrents sur les chemins de votre réseau.

Prise en charge des adresses IPv4 et IPv6

Le Moniteur réseau fournit des mesures de disponibilité et de performance sur les réseaux IPv4 ou IPv6 et peut surveiller les adresses IPv4 ou IPv6 à partir de VPC à double pile. Le Moniteur réseau n'autorise pas la configuration des destinations IPv4 et IPv6 sur le même moniteur, mais vous pouvez créer des destinations distinctes pour les destinations IPv4 uniquement et IPv6 uniquement.

Disponibilité dans les Régions

Network Monitor est actuellement disponible dans les versions suivantes Régions AWS :

Région	
Asie-Pacifique (Hong Kong)	ap-east-1
Asie-Pacifique (Mumbai)	ap-south-1
Asia Pacific (Seoul)	ap-northeast-2
Asie-Pacifique (Singapour)	ap-southeast-1
Asie-Pacifique (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada Ouest (Calgary)	ca-west-1
Europe (Francfort)	eu-central-1
Europe (Irlande)	eu-west-1

Région	
Europe (Londres)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Moyen-Orient (Bahreïn)	me-south-1
Amérique du Sud (São Paulo)	sa-east-1
US East (Virginie du Nord)	us-east-1
USA Est (Ohio)	us-east-2
USA Ouest (Californie du Nord)	us-west-1
US West (Oregon)	us-west-2

Création d’un Moniteur réseau

Les étapes suivantes décrivent la création d’un moniteur, puis l’ajout des sondes requises. Pour une sonde, vous choisirez le sous-réseau source et jusqu’à quatre adresses IP de destination pour un maximum de 24 sondes par moniteur. Vous pouvez créer un package de modèle à l’aide de la console Amazon CloudWatch ou à l’aide de la ligne de commande ou de l’API.

Rubriques

Création d’un Moniteur réseau à l’aide de la console

Les étapes suivantes décrivent la création d’un moniteur à l’aide de la console Amazon CloudWatch . Vous allez choisir vos sous-réseaux sources, puis ajouter jusqu’à quatre destinations pour créer jusqu’à 24 sondes par moniteur. Vous pouvez créer un package de modèle à l’aide de la console Amazon CloudWatch ou à l’aide de la ligne de commande ou du kit SDK.

> **Important**
>
> Ces étapes sont conçues pour être effectuées en une seule fois. Vous ne pourrez pas enregistrer les travaux en cours pour les poursuivre plus tard.

Définir les détails du moniteur

La première étape de la création d’un moniteur consiste à définir les détails de base. Cela inclut l’attribution d’un nom au moniteur et la définition de la période d’agrégation. Si vous le souhaitez, vous pouvez ajouter des balises au moniteur.

Pour définir les détails du moniteur

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Choisissez Create monitor (Créer un contrôle).
3. Pour Nom du moniteur, saisissez le nom que vous voulez utiliser pour ce moniteur.
4. Pour la période d'agrégation, choisissez la fréquence à laquelle vous souhaitez envoyer les métriques CloudWatch. Les périodes d’agrégation disponibles sont les suivantes :

- 30 secondes
- 60 secondes

> **Note**
>
> Une période d’agrégation plus courte permet de détecter plus rapidement les problèmes de réseau ; toutefois, la période d’agrégation que vous choisissez peut affecter votre structure de facturation. Pour plus d'informations sur les tarifs, consultez la page de CloudWatch tarification d'Amazon.

5. (Facultatif) Dans la section Balises, ajoutez des paires Clé et Valeur pour mieux identifier cette ressource et vous permettre de rechercher ou de filtrer des informations spécifiques.

   1. Sélectionnez Ajouter une nouvelle balise.
   2. Saisissez le nom de la Clé et la Valeur associée.
   3. Choisissez Ajouter une nouvelle balise pour ajouter une balise.

      Vous pouvez ajouter plusieurs balises en choisissant Ajouter une nouveau balise, ou vous pouvez supprimer n’importe quelle balise en choisissant Supprimer.
   4. Si vous souhaitez associer vos balises au moniteur, maintenez la case Ajouter des balises aux sondes créées par le moniteur cochée. Cela ajoute les balises aux sondes du moniteur, ce qui peut être utile si vous utilisez l’authentification ou la mesure basée sur des balises.

6. Choisissez Suivant vers the section called “Choisissez la source et la destination”.

Choisissez la source et la destination

Un moniteur réseau utilise une AWS source pour les VPC et les sous-réseaux associés dans les régions où fonctionne votre réseau. Une destination de moniteur est la combinaison de vos adresses IP sur site, de vos protocoles réseau, de vos ports et de la taille de vos paquets réseau.

La combinaison de la source et de la destination est appelée sonde. Vous pouvez avoir jusqu’à quatre sondes par sous-réseau, et un total de 24 sondes par moniteur.

> **Important**
>
> Ces étapes sont conçues pour être effectuées en une seule fois. Vous ne pourrez pas enregistrer les travaux en cours pour les poursuivre plus tard.

Pour choisir une source et une destination

1. Sous Source du réseau AWS, choisissez un ou plusieurs sous-réseaux à inclure dans le moniteur. Vous pouvez choisir un seul VPC, qui choisira ensuite tous les sous-réseaux de ce VPC, ou vous pouvez choisir des sous-réseaux spécifiques. Les VPC et les sous-réseaux que vous choisissez seront la source du moniteur réseau.
2. Pour Destination 1, saisissez l'adresse IP de destination du réseau sur site. Les protocoles IPv4 et IPv6 sont tous deux pris en charge.
3. Choisissez Advanced Settings (Paramètres avancés).
4. Pour cette destination gérée par le client, choisissez le Protocole réseau. Il peut s'agir de :
   - ICMP
   - TCP
5. Si le Protocole est TCP, saisissez les informations suivantes. Sinon, passez à l'étape suivante :
   1. Saisissez le Port que votre réseau utilise pour se connecter. Le port doit être un nombre compris entre 1 et 65 535.
   2. Saisissez la Taille de paquet. Il s'agit de la taille, en octets, de chaque paquet envoyé sur la sonde entre la source et la destination. La taille de paquet doit être un nombre compris entre 56 et 8 500.
6. Choisissez Ajouter une destination pour ajouter une autre destination sur site à ce moniteur. Répétez ces étapes pour chaque destination que vous souhaitez ajouter.
7. Choisissez Suivant lorsque vous avez terminé pour confirmer les sondes.

Confirmation des sondes

La confirmation des sondes vous permet de passer en revue la combinaison de sondes réseau pour le moniteur. Cette page affiche toutes les combinaisons possibles des sources et destinations que vous avez choisies. Par exemple, si vous avez six sous-réseaux sources et quatre adresses IP de destination, vous aurez un total de 24 combinaisons de sondes possibles.

> **Important**
>
> - Ces étapes sont conçues pour être effectuées en une seule fois. Vous ne pourrez pas enregistrer les travaux en cours pour les poursuivre plus tard.
> - La page Confirmer les sondes n'indique pas si une sonde est valide. Par conséquent, nous vous recommandons de passer en revue cette page et de supprimer toutes les sondes non valides. Si vous ne supprimez pas les sondes non valides, elles peuvent vous être facturées.

Pour confirmer les sondes de moniteur

1. Prérequis : the section called "Choisissez la source et la destination".
2. Sur la page Confirmer les sondes, consultez la liste des combinaisons source et destination.
3. Choisissez une ou plusieurs sondes que vous souhaitez supprimer du moniteur, puis sélectionnez Supprimer.

   > **Note**
   >
   > Vous n'êtes pas invité à confirmer la suppression. Une fois qu'une sonde est supprimée, elle doit être reconfigurée. Vous pouvez ajouter une sonde à nouveau à un moniteur depuis la section Moniteurs réseau de la page Moniteur réseau. Pour plus d'informations, consultez the section called "Ajouter une sonde à un moniteur".

4. Choisissez Suivant pour consulter les détails du moniteur avant de le créer.

Vérifier et créer

La dernière étape de la création d'un moniteur et de sondes consiste à examiner les détails du moniteur et des sondes. Vous pouvez modifier toutes les informations à ce stade. Lorsque vous avez terminé la révision et créé le moniteur, et que les métriques commencent à être suivies, vous commencerez à être facturé pour toutes les sondes.

> **⚠ Important**
>
> - Cette étape est conçue pour être réalisée en une seule fois lors de la création d'un moniteur et d'une sonde. Vous ne pourrez pas enregistrer les travaux en cours pour les poursuivre plus tard.
> - Si vous choisissez de modifier une section, vous devrez procéder à la création du moniteur dès le moment où vous la modifiez. Vous n'aurez toutefois pas besoin de recréer les étapes suivantes. Ces pages conservent les informations précédemment renseignées.

Pour examiner et créer un moniteur

1. Sur la page Réviser et créer des sondes, choisissez Modifier pour chaque section dans laquelle vous souhaitez apporter des modifications.
2. Apportez les modifications nécessaires dans cette section.
3. Choisissez Suivant.
4. Effectuez l'une des actions suivantes :
   - Apportez les modifications souhaitées sur les pages de surveillance supplémentaires, puis cliquez sur Suivant jusqu'à ce que vous reveniez à la page Révision et création.
   - Si aucune autre page n'a besoin d'être modifiée, choisissez Suivant jusqu'à ce que vous reveniez à la page Révision et création.
5. Choisissez Create monitor (Créer un contrôle).

   La page Moniteur réseau affiche l'état actuel de la création du moniteur dans la section Moniteurs réseau. Lors de la création du moniteur, l'état est En attente. Lorsque l'état devient Actif, vous pouvez accéder au tableau de bord du moniteur pour consulter CloudWatch les métriques.

   Pour plus d'informations sur l'utilisation du tableau de bord du moniteur, veuillez consulter la rubrique the section called "Tableaux de bord du Moniteur réseau".

> **Note**
>
> Le moniteur réseau récemment ajouté peut prendre plusieurs minutes pour commencer à collecter les métriques réseau.

Création d'un Moniteur réseau à l'aide de la ligne de commande ou de l'API

Utilisez la ligne de commande ou l'API pour afficher et créer un moniteur réseau.

Pour créer un moniteur réseau à l'aide de la ligne de commande ou de l'API

1. Créez un moniteur réseau à l'aide de la commande create-monitor.
2. Créez une sonde du moniteur réseau à l'aide de la commande create-probe.

Utilisation des moniteurs et des sondes du Moniteur réseau

Vous pouvez effectuer l'une des tâches suivantes avec vos moniteurs et sondes, soit à l'aide de la console Amazon CloudWatch , soit à l'aide de la ligne de commande ou de l'API.

Rubriques :

- Modifier un moniteur
- Suppression d'un moniteur
- Activation ou désactivation d'une sonde
- Ajouter une sonde à un moniteur
- Modification d'une sonde
- Suppression d'une sonde
- Balisage ou annulation de balisage des ressources à l'aide de la ligne de commande ou de l'API

Modifier un moniteur

Vous pouvez modifier toutes les informations relatives à un Moniteur réseau, notamment le renommer, définir une nouvelle période d'agrégation ou ajouter ou supprimer des balises. La modification des informations d'un moniteur ne modifie aucune des sondes associées. Vous pouvez modifier un moniteur à l'aide de la console Amazon CloudWatch , ou à l'aide de la ligne de commande ou de l'API.

Modification d’un moniteur à l’aide de la console

Utilisez la CloudWatch console pour modifier un moniteur.

Pour modifier un moniteur à l’aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Dans la section Moniteurs réseau, choisissez le moniteur que vous souhaitez modifier.
3. Sur la page du tableau de bord du moniteur, choisissez Modifier.
4. Pour le Nom du moniteur, saisissez le nouveau nom du moniteur.
5. Pour la période d'agrégation, choisissez la fréquence à laquelle vous souhaitez envoyer les métriques CloudWatch. Les périodes valides sont les suivantes :

   - 30 secondes
   - 60 secondes

   > **Note**
   >
   > Une période d’agrégation plus courte permet de détecter plus rapidement les problèmes de réseau ; toutefois, la période d’agrégation que vous choisissez peut affecter votre structure de facturation. Pour plus d'informations sur les tarifs, consultez la page de CloudWatch tarification d'Amazon.

6. (Facultatif) Dans la section Balises, ajoutez des paires Clé et Valeur pour mieux identifier cette ressource et vous permettre de rechercher ou de filtrer des informations spécifiques. Vous pouvez également simplement modifier la Valeur de n’importe quelle Clé actuelle.

   1. Sélectionnez Ajouter une nouvelle balise.
   2. Saisissez le nom de la Clé et la Valeur associée.
   3. Choisissez Ajouter une nouvelle balise pour ajouter une balise.

      Vous pouvez ajouter plusieurs balises en choisissant Ajouter une nouveau balise, ou vous pouvez supprimer n’importe quelle balise en choisissant Supprimer.
   4. Si vous souhaitez associer vos balises au moniteur, maintenez la case Ajouter des balises aux sondes créées par le moniteur cochée. Cela ajoute les balises aux sondes du moniteur, ce qui peut être utile si vous utilisez l’authentification ou la mesure basée sur des balises.

7. Sélectionnez Enregistrer les modifications.

Modification d’un moniteur à l’aide de la CLI ou de l’API

Utilisez la ligne de commande ou l’API pour afficher et modifier un moniteur.

Pour modifier un moniteur à l’aide de la ligne de commande ou de l’API

1. Utilisez la commande list-monitors pour obtenir la liste de vos moniteurs si vous ne connaissez pas le nom du moniteur. Notez le nom du moniteur que vous souhaitez modifier.
2. Utilisez la commande edit-monitor en utilisant le nom du moniteur indiqué à l’étape précédente.

Suppression d’un moniteur

Avant de supprimer un moniteur, vous devez désactiver ou supprimer toutes les sondes associées à ce moniteur, quel que soit son état. Une fois qu’un moniteur est désactivé ou supprimé, vous ne serez plus facturé pour les sondes. Il n’est pas possible de récupérer un moniteur supprimé. Vous pouvez supprimer un moniteur à l'aide de la Amazon CloudWatch console ou de la ligne de commande/API.

Bien qu'une sonde puisse être supprimée ou désactivée, elle CloudWatch conserve les indicateurs pendant 15 jours.

Suppression d’un moniteur à l’aide de la console

Utilisez la CloudWatch console pour supprimer un moniteur.

Pour supprimer un moniteur à l’aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Dans la section Moniteurs réseau, choisissez le moniteur que vous souhaitez supprimer.
3. Choisissez Actions, puis Supprimer.
4. Si vous avez des sondes actives, il vous sera demandé de les désactiver. Choisissez Désactiver les sondes.

> **Note**
>
> Vous ne pouvez pas annuler cette action après avoir sélectionné Désactiver les sondes. Les sondes désactivées ne sont toutefois pas retirées du moniteur. Vous pouvez la réactiver par la suite. veuillez consulter the section called “Activation ou désactivation d’une sonde”.

5. Saisissez **`confirm`** dans le champ de confirmation, puis choisissez Supprimer.

Suppression d’un moniteur à l’aide de la ligne de commande ou de l’API

Supprimez un moniteur à l’aide de la ligne de commande ou de l’API.

Pour supprimer un moniteur réseau à l’aide de la ligne de commande ou de l’API

1. Vous aurez besoin du nom du moniteur que vous souhaitez supprimer. Si vous ne connaissez pas le nom, utilisez la commande list-monitors pour obtenir la liste de vos moniteurs. Notez le nom du moniteur que vous souhaitez supprimer.
2. Vérifiez si ce moniteur contient des sondes. Utilisez a commande get-monitor avec le nom du moniteur indiqué à l’étape précédente. Cela renvoie une liste de toutes les sondes associées à ce moniteur.
3. Si le moniteur contient des sondes, vous devez d’abord les désactiver ou les supprimer.
   - Pour rendre une sonde inactive, utilisez la commande update-probe et définissez l’état sur `INACTIVE`.
   - Pour supprimer une sonde, utilisez la commande delete-probe.
4. Une fois que les sondes sont définies sur `INACTIVE` ou supprimées, utilisez la commande delete-monitor pour supprimer le moniteur. Les sondes inactives ne sont pas supprimées.

Activation ou désactivation d’une sonde

Vous pouvez activer ou désactiver une sonde de moniteur selon vos besoins. Vous pouvez désactiver une sonde si vous ne l’utilisez pas actuellement, mais que vous souhaitez peut-être l’utiliser à nouveau à l’avenir. En désactivant une sonde, vous n’aurez pas besoin de perdre du temps à la configurer à nouveau. Les sondes désactivées ne vous sont pas facturées.

Vous pouvez modifier l'état d'un moniteur à l'aide de la Amazon CloudWatch console, de la ligne de commande ou de l'API.

Activation ou désactivation d’une sonde à l’aide de la console

Utilisez la CloudWatch console pour activer ou désactiver une sonde.

Pour activer ou désactiver une sonde à l’aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Choisissez l’onglet Détails du moniteur.
3. Dans la section Sondes, choisissez la sonde que vous souhaitez activer ou désactiver.
4. Choisissez Actions, puis sélectionnez Activer ou Désactiver.

> **Note**
>
> Si vous réactivez une sonde désactivée, vous commencerez à payer des frais de facturation pour cette sonde.

Activation ou désactivation d’une sonde à l’aide de la ligne de commande ou de l’API

Activez ou désactivez une sonde à l’aide de la ligne de commande ou de l’API. Vous ne pouvez utiliser cette commande que pour une seule sonde.

Pour activer ou désactiver une sonde à l’aide de la ligne de commande ou de l’API

1. Utilisez la commande list-monitors pour obtenir la liste de vos moniteurs si vous ne connaissez pas le nom du moniteur. Notez le nom du moniteur dont vous souhaitez modifier l’état de sonde.
2. Utilisez a commande get-monitor avec le nom du moniteur indiqué à l’étape précédente. Cela renvoie une liste de toutes les sondes associées à ce moniteur. Notez l’ID des sondes dont vous souhaitez modifier l’état.
3. Utilisez la commande update-probe et configurez la sonde dont vous souhaitez modifier l’état pour qu’elle soit `ACTIVE` ou `INACTIVE`.

Ajouter une sonde à un moniteur

Vous pouvez ajouter une sonde à un moniteur existant. Notez que si vous ajoutez des sondes à un moniteur, votre structure de facturation sera mise à jour pour indiquer qu'une nouvelle sonde a été ajoutée.

Ajout d'une sonde à un moniteur à l'aide de la console

Pour ajouter une sonde à un moniteur à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Dans la section Moniteurs réseau, effectuez l'une des actions suivantes :
   - Choisissez le lien Nom du moniteur auquel vous souhaitez ajouter une sonde. Choisissez l'onglet Détails du moniteur, puis dans la section Sondes, sélectionnez Ajouter une sonde.
   - Cochez la case du moniteur, sélectionnez Actions, puis choisissez Ajouter une sonde.
3. Sur la page Ajouter une sonde, procédez de la façon suivante :
   1. Sous Source du réseau AWS, choisissez un sous-réseau à ajouter au moniteur.

      > **Note**
      >
      > Vous ne pouvez ajouter qu'une seule sonde à la fois et jusqu'à quatre sondes par moniteur.

   2. Saisissez l'adresse IP de destination du réseau sur site. Les protocoles IPv4 et IPv6 sont tous deux pris en charge.
   3. Choisissez Advanced Settings (Paramètres avancés).
   4. Choisissez le Protocole réseau pour la destination. Celui-ci peut être ICMP ou TCP.
   5. Si le Protocole est TCP, saisissez les informations suivantes. Sinon, passez à l'étape suivante :
      - Saisissez le Port que votre réseau utilise pour se connecter. Le port doit être un nombre compris entre 1 et 65 535.
      - Saisissez la Taille de paquet. Il s'agit de la taille, en octets, de chaque paquet transmis par la sonde entre la source et la destination. La taille de paquet doit être un nombre compris entre 56 et 8 500.

4. (Facultatif) Dans la section Balises, ajoutez des paires Clé et Valeur pour mieux identifier cette ressource et vous permettre de rechercher ou de filtrer des informations spécifiques.

   1. Sélectionnez Ajouter une nouvelle balise.
   2. Saisissez le nom de la Clé et la Valeur associée.
   3. Choisissez Ajouter une nouvelle balise pour ajouter une nouvelle balise.

      Vous pouvez ajouter plusieurs balises en choisissant Ajouter une nouveau balise, ou vous pouvez supprimer n'importe quelle balise en choisissant Supprimer.
5. Choisissez Ajouter une sonde.

   Lorsque la sonde est en activation, l'état indique En attente. Plusieurs minutes peuvent être nécessaires pour que la sonde devienne active.

Ajout d'une sonde à l'aide de la ligne de commande ou d'une API

Ajoutez une sonde à l'aide de la ligne de commande ou d'une API. Vous ne pouvez utiliser cette commande que pour une seule sonde à la fois.

Pour ajouter une sonde à l'aide de la ligne de commande ou d'une API

1. Utilisez la commande list-monitors pour obtenir la liste de vos moniteurs si vous ne connaissez pas le nom du moniteur. Notez le nom du moniteur auquel vous souhaitez ajouter une sonde.
2. Utilisez la commande create-probe pour ajouter une sonde au moniteur.

Modification d'une sonde

Vous pouvez modifier toutes les informations relatives à une sonde en cours, que cette sonde soit activée ou désactivée. Vous pouvez modifier une sonde à l'aide de la console Amazon CloudWatch , ou à l'aide de la ligne de commande ou de l'API.

Modification d'une sonde à l'aide de la console

Pour modifier une sonde à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.

   Cliquez sur le lien Nom pour ouvrir le tableau de bord du moniteur.

2. Choisissez l’onglet Détails du moniteur.
3. Dans la section Sondes, sélectionnez le lien de la sonde que vous souhaitez modifier.
4. Sur la page du tableau de bord de la sonde, choisissez Modifier ou Actions, puis Modifier.
5. Sur la page Modifier la sonde, saisissez la nouvelle adresse IP de la sonde de destination. Les protocoles IPv4 et IPv6 sont tous deux pris en charge.
6. Choisissez Advanced Settings (Paramètres avancés).
7. Choisissez le Protocole réseau. Celui-ci peut être ICMP ou TCP.
8. Si le Protocole est TCP, saisissez les informations suivantes. Sinon, passez à l’étape suivante :
   - Saisissez le Port que votre réseau utilise pour se connecter. Le port doit être un nombre compris entre 1 et 65 535.
   - Saisissez la Taille de paquet. Il s’agit de la taille, en octets, de chaque paquet transmis par la sonde entre la source et la destination. La taille de paquet doit être un nombre compris entre 56 et 8 500.
9. (Facultatif) Dans la section Balises, ajoutez des paires Clé et Valeur pour mieux identifier cette ressource et vous permettre de rechercher ou de filtrer des informations spécifiques.
   1. Sélectionnez Ajouter une nouvelle balise.
   2. Saisissez le nom de la Clé et la Valeur associée.
   3. Choisissez Ajouter une nouvelle balise pour ajouter une nouvelle balise.

      Vous pouvez ajouter plusieurs balises en choisissant Ajouter une nouveau balise, ou vous pouvez supprimer n’importe quelle balise en choisissant Supprimer.
10. Sélectionnez Enregistrer les modifications.

Modification d’une sonde à l’aide de la ligne de commande ou de l’API

Utilisez la ligne de commande pour modifier une sonde de moniteur. Vous ne pouvez utiliser cette commande que pour une seule sonde.

Pour modifier une sonde aide de la ligne de commande ou de l’API

1. Utilisez la commande list-monitors pour obtenir la liste de vos moniteurs si vous ne connaissez pas le nom du moniteur. Notez le nom du moniteur dont vous souhaitez modifier l’état de sonde.

2. Utilisez a commande get-monitor avec le nom du moniteur indiqué à l'étape précédente. Cela renvoie une liste de toutes les sondes associées à ce moniteur. Notez l'ID de sonde que vous souhaitez modifier.
3. Utilisez la commande update-probe pour modifier les informations de la sonde.

Suppression d'une sonde

Vous pouvez supprimer une sonde plutôt que de la désactiver si vous savez que vous n'en aurez plus besoin à l'avenir. Vous ne pouvez pas récupérer une sonde supprimée et vous devez la recréer. La facturation de cette sonde s'arrête lorsque celle-ci est supprimée. Vous pouvez supprimer une sonde à l'aide de la console Amazon CloudWatch , ou à l'aide de la ligne de commande ou de l'API.

Suppression d'une sonde à l'aide de la console

Pour supprimer une sonde à l'aide de la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Dans la section Moniteurs réseau, cliquez sur le lien Nom pour ouvrir le tableau de bord du moniteur.
3. Choisissez l'onglet Détails du moniteur.
4. Cochez la case du moniteur, sélectionnez Actions, puis sélectionnez Supprimer.
5. Dans la boîte de dialogue Supprimer une sonde, choisissez Supprimer pour confirmer votre souhait de supprimer la sonde.
6. Choisissez Supprimer pour confirmer que vous voulez supprimer la sonde.

   L'état de la sonde dans la section Sondes indique Suppression. Une fois supprimée, la sonde est retirée de la section Sondes.

Suppression d'une sonde à l'aide de la ligne de commande ou de l'API

Supprimez une sonde à l'aide de la ligne de commande ou de l'API. Vous ne pouvez utiliser cette commande que pour une seule sonde.

Pour activer ou désactiver une sonde à l'aide de la ligne de commande ou de l'API

1. Utilisez la commande list-monitors pour obtenir la liste de vos moniteurs si vous ne connaissez pas le nom du moniteur. Notez le nom du moniteur avec la sonde que vous souhaitez supprimer

2. Utilisez a commande get-monitor avec le nom du moniteur indiqué à l'étape précédente. Cela renvoie une liste de toutes les sondes associées à ce moniteur. Notez l'ID de toute sonde que vous souhaitez supprimer.
3. Utilisez la commande delete-probe.

Balisage ou annulation de balisage des ressources à l'aide de la ligne de commande ou de l'API

Vous pouvez utiliser la ligne de commande ou la CLI pour ajouter ou mettre à jour des balises de ressources.

Pour mettre à jour les balises réseau à l'aide de la ligne de commande ou de l'API

- Pour répertorier les balises de ressources, utilisez list-tags-for-resources.
- Pour baliser une ressource, utilisez la commande tag-resource.
- Pour annuler le balisage d'une ressource, utilisez la commande untag-resource.

Tableaux de bord du Moniteur réseau

Vous pouvez utiliser le tableau de bord Amazon CloudWatch Network Monitor pour consulter l'état AWS du réseau, évaluer le temps d'aller-retour et les pertes de paquets. Vous pouvez consulter ces mesures à la fois pour les moniteurs et pour les sondes individuelles.

Tableaux de bord du Moniteur réseau

- Tableau de bord de moniteur
- Tableau de bord de sonde

Alarmes de sonde

Vous pouvez créer des CloudWatch alarmes Amazon en fonction des métriques Amazon CloudWatch Network Monitor, comme vous le pouvez pour les autres CloudWatch métriques Amazon. Toute alarme que vous créez apparaît dans la colonne État de la sonde de la section Détails du moniteur du tableau de bord du moniteur réseau lorsque l'alarme est déclenchée. Le statut sera OK ou En alarme. Si aucun état n'est affiché pour une sonde, aucune alarme n'a été créée pour cette sonde.

Par exemple, vous pouvez créer une alarme basée sur la métrique `PacketLoss` du Moniteur réseau et la configurer pour envoyer une notification lorsque la métrique est supérieure à une valeur que vous avez choisie. Vous configurez les alarmes pour les métriques Network Monitor en suivant les mêmes directives que pour les autres CloudWatch métriques.

Les mesures suivantes sont disponibles sous `AWS/NetworkMonitor` lors de la création d'une CloudWatch alarme pour Network Monitor.

- HealthIndicator
- PacketLoss
- RTT (Round-trip time)

Pour les étapes de création d’une alarme du Moniteur réseau, veuillez consulter la rubrique the section called “Créez une alerte basée sur un seuil statique”.

Fixer un délai pour les métriques

Les métriques et les événements des deux tableaux de bord utilisent une durée par défaut de deux heures, calculée à partir de l’heure actuelle. Vous pouvez modifier la valeur par défaut pour utiliser l’une des valeurs prédéfinies suivantes :

- 1h : une heure
- 2h : deux heures
- 1d : un jour
- 1w : une semaine

Vous pouvez également définir une durée personnalisée. Choisissez Personnalisé, choisissez un temps absolu ou relatif, puis définissez le délai à une durée de votre choix. Le temps relatif ne prend en charge que 15 jours à compter de la date d'aujourd'hui, par CloudWatch défaut.

En outre, vous pouvez choisir l’heure affichée dans les graphiques en fonction du fuseau horaire UTC ou d’un fuseau horaire local.

Tableau de bord de moniteur

Vous pouvez utiliser le tableau de bord Amazon CloudWatch Network Monitor pour consulter l'état AWS du réseau, évaluer le temps d'aller-retour et les pertes de paquets. Le Moniteur réseau possède des tableaux de bord pour les moniteurs et les sondes.

Pour accéder au tableau de bord d’un moniteur

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Dans la section Moniteurs réseau, cliquez sur le lien Nom pour ouvrir le tableau de bord du moniteur.

Présentation

La page Présentation affiche les informations suivantes pour votre moniteur :

- AWS État du AWS réseau — L'état du réseau affiche uniquement l'état général du AWS réseau. L’état sera soit Sain, soit Dégradé. Un état sain indique que le Moniteur réseau n'a observé aucun problème avec le AWS réseau. L'état Dégradé indique que le Moniteur réseau a détecté un problème avec le AWS réseau. La barre d’état de cette section indique l’état du réseau sur une durée par défaut d’une heure. Survolez n’importe quel point de la barre d’état pour afficher des informations supplémentaires.
- Résumé du trafic de sonde : affiche l'état actuel du trafic entre les AWS sous-réseaux sources du moniteur et les adresses IP de destination. Le Résumé du trafic des sondes affiche les informations suivantes :
  - Sondes en alarme : ce chiffre indique combien de vos sondes sont dans un état dégradé. Une alarme est déclenchée lorsqu’une métrique que vous avez configurée comme alarme est déclenchée. Pour plus d'informations sur les alarmes métriques du Network Monitor, consultezthe section called “Alarmes de sonde”.
  - Perte de paquets : nombre de paquets perdus du sous-réseau source vers l’adresse IP de destination. Ceci est représenté sous forme de pourcentage du nombre total de paquets envoyés.
  - Temps de propagation aller et retour : temps nécessaire, en millisecondes, à un paquet provenant du sous-réseau source pour atteindre l’adresse IP de destination, puis revenir.

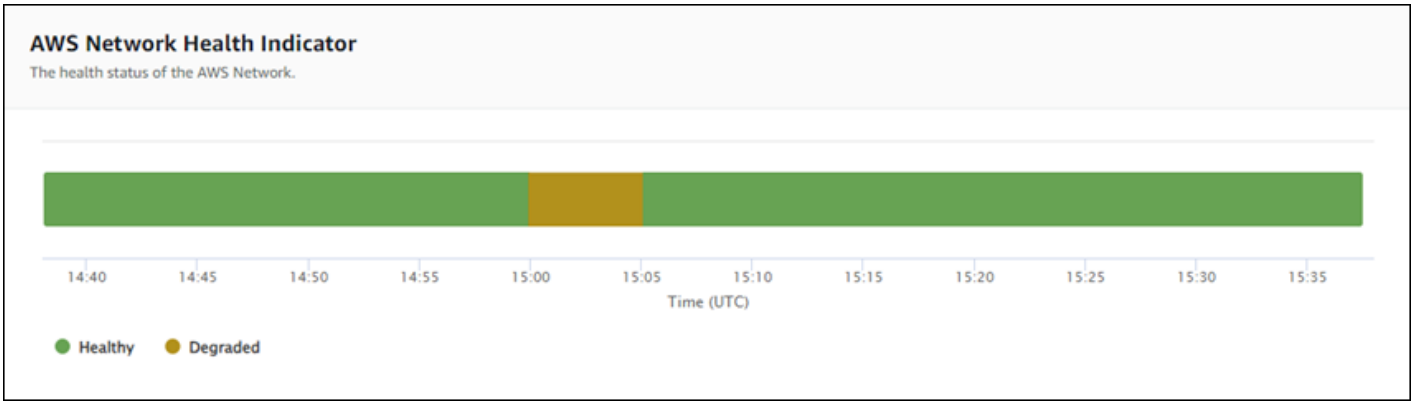
Les données sont représentées par un graphique interactif qui vous permet de voir les détails.

Par défaut, les données sont affichées pendant une période de deux heures, calculée à partir de la date et de l’heure actuelles. Vous pouvez cependant modifier la plage en fonction de vos besoins. Pour plus d’informations, consultez the section called “Fixer un délai pour les métriques”.

Métriques de suivi

Le tableau de bord du Moniteur réseau affiche une représentation graphique de vos moniteurs et sondes. Les graphiques suivants sont disponibles :

- AWS Indicateur de santé du réseau : il représente l'état du AWS réseau sur une période donnée. L'état sera soit Sain, soit Dégradé. Dans l'exemple suivant, vous verrez que de 15 h 00 UTC à 15 h 05 UTC, le AWS réseau était dans un état dégradé. Après 15 h 05, le réseau est revenu à un état sain. Vous pouvez survoler n'importe quelle section du graphique pour afficher des détails supplémentaires.



> **Note**
>
> L'indicateur de santé du réseau n'indique pas l'état de santé de la sonde, mais uniquement celui du AWS réseau.

- Perte de paquets : ce graphique affiche une ligne unique indiquant le pourcentage de perte de paquets pour chaque sonde du moniteur. La légende au bas de la page affiche chacune des sondes du moniteur, codées par couleur pour garantir leur unicité. Dans ce graphique, le survol d'une sonde permet d'afficher le sous-réseau source, l'adresse IP de destination et le pourcentage de perte de paquets. Dans l'exemple suivant, une alarme de perte de paquets a été configurée pour une sonde depuis un sous-réseau vers l'adresse IP 127.0.0.1. L'alarme a été déclenchée lorsque le seuil de perte de paquets a été dépassé pour la sonde. Le survol du graphique indique la source et la destination de la sonde, et indique qu'il y a eu une perte de paquets de 30,97 % pour cette sonde le 21 novembre à 02:41:30.

- Temps de propagation aller et retour : ce graphique affiche une ligne pour chaque sonde, indiquant le temps de propagation aller et retour pour chaque sonde. La légende au bas de la page affiche chacune des sondes du moniteur, codées par couleur pour garantir leur unicité. Dans ce graphique, le survol d'une sonde permet d'afficher le sous-réseau source, l'adresse IP de destination et le temps de propagation aller et retour. L'exemple suivant montre que le mardi 21 novembre à 21 h 45 min 30 s, le temps de propagation aller et retour d'une sonde entre un sous-réseau et l'adresse IP 127.0.0.1 était de 0,075 seconde.

Détails du moniteur

La page Détails du moniteur affiche les informations relatives à votre moniteur, y compris les sondes. Sur cette page, vous pouvez gérer les balises ou ajouter une sonde. Cette page est divisée en trois sections :

- Détails du moniteur : cette section fournit des informations sur votre moniteur. Les informations de cette section ne peuvent pas être modifiées. Vous pouvez toutefois choisir le lien Nom du rôle pour afficher les détails du rôle lié au service du Moniteur réseau.
- Sondes : cette section affiche la liste de toutes les sondes associées au moniteur. Choisissez un lien VPC ou ID de sous-réseau pour ouvrir les détails du VPC ou du sous-réseau dans la console Amazon VPC. Vous pouvez également modifier une sonde, notamment en l'activant ou en la désactivant. Pour plus d'informations, consultez the section called "Utilisation de moniteurs et de sondes".

  La section Probes affiche des informations sur chaque sonde configurée pour ce moniteur, notamment l'ID de sonde, l'ID VPC, l'ID de sous-réseau, l'adresse IP, le protocole et indique si l'état de la sonde est actif ou inactif. Si vous avez configuré une alarme pour une sonde, l'état actuel de cette alarme s'affiche. OK indique qu'aucun événement de mesure n'a déclenché d'alarme ; En alarme indique qu'une métrique que vous avez configurée CloudWatch a déclenché une alarme. Si aucun état n'est affiché pour une sonde, aucune CloudWatch alarme n'a été configurée. Pour plus d'informations sur les types d'alarmes de sonde Network Monitor que vous pouvez créer, consultezthe section called "Alarmes de sonde".
- Balises : affiche les balises actuelles d'un moniteur. Vous pouvez ajouter ou supprimer des balises en choisissant Gérer les balises. Cela ouvre la page Modifier la sonde. Pour en savoir plus sur la modification des balises, veuillez consulter la rubrique the section called "Modifier un moniteur".

Tableau de bord de sonde

Vous pouvez utiliser le tableau de bord Amazon CloudWatch Network Monitor pour consulter l'état AWS du réseau, ainsi que des informations sur le temps d'aller-retour et la perte de paquets pour des sondes spécifiques. Il existe deux tableaux de bord de sonde, Présentation et Détails de la sonde.

Vous pouvez créer des CloudWatch alarmes pour définir des seuils métriques de perte de paquets et de temps d'aller-retour. Lorsqu'un seuil est atteint pour une métrique, une CloudWatch alarme vous en informe. Pour plus d'informations sur la création d'alarmes de sonde, veuillez consulter la rubrique the section called "Alarmes de sonde".

Pour accéder à un tableau de bord de sonde

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/, puis sous Surveillance du réseau, sélectionnez Moniteur réseau.
2. Dans la section Moniteurs réseau, cliquez sur le lien Nom pour ouvrir le tableau de bord du moniteur.
3. Cliquez sur le lien ID pour afficher le tableau de bord de cette sonde.

Présentation

La page Présentation affiche les informations suivantes pour votre sonde :

- AWS Détails de l'indicateur de santé du réseau : il fournit uniquement l'état de santé global du AWS réseau. L'état sera soit Sain, soit Dégradé. L'état Dégradé indique qu'il y a un problème avec le AWS réseau, mais n'indique pas s'il y a un problème avec votre sonde.
- Perte de paquets : nombre de paquets perdus depuis le sous-réseau source vers l'adresse IP de destination pour cette sonde.
- Temps de propagation aller et retour : temps nécessaire, en millisecondes, à un paquet provenant du sous-réseau source pour atteindre l'adresse IP de destination, puis revenir.

Détails de la sonde

La page Détails de la sonde affiche les détails d'une sonde. Sur cette page, vous pouvez modifier la sonde. Pour plus d'informations, consultez the section called "Utilisation de moniteurs et de sondes".

- Détails de la sonde : cette page fournit des informations générales sur la sonde. Les informations de cette section ne peuvent pas être modifiées.
- Source et destination de la sonde : cette section affiche les détails de la sonde. Choisissez un lien VPC ou ID de sous-réseau pour ouvrir les détails du VPC ou du sous-réseau dans la console Amazon VPC. Vous pouvez également modifier une sonde, notamment en l'activant ou en la désactivant.
- Balises : affiche les balises actuelles d'un moniteur. Vous pouvez ajouter ou supprimer des balises en choisissant Gérer les balises. Cela ouvre la page Modifier la sonde. Pour en savoir plus sur la modification des balises, veuillez consulter la rubrique the section called "Modification d'une sonde".

Quotas du Moniteur réseau

Les quotas du Moniteur réseau sont les suivants :

Quota	Par défaut	Ajustable
Nombre maximum de moniteurs par compte Région AWS	100	Oui
Nombre maximal de sondes par moniteur	24	Oui
Nombre maximal de sondes par sous-réseau et par moniteur	4	Oui

Sécurité et protection des données dans le Moniteur réseau

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez de centres de données et d'architectures 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 cela 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 Amazon CloudWatch Network Monitor, consultez la section 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 de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de CloudWatch Network Monitor. Les rubriques suivantes expliquent comment configurer CloudWatch Network Monitor 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 les ressources de votre moniteur CloudWatch réseau.

Rubriques

Protection des données dans Amazon CloudWatch Network Monitor

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans Amazon CloudWatch Network Monitor. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d’informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l’authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- 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 140-2 lors de l'accès AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

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 Name (Nom). Cela inclut lorsque vous travaillez avec CloudWatch Network Monitor ou un autre outil Services AWS à l'aide de la console, de l'API ou AWS des SDK. AWS CLI 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.

Sécurité de l'infrastructure dans Amazon CloudWatch Network Monitor

En tant que service géré, Amazon CloudWatch Network Monitor est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à CloudWatch Network Monitor via le réseau. Les clients doivent supporter le protocole TLS (Sécurité de la couche transport) 1.0 ou une version ultérieure. Nous recommandons TLS 1.2 ou version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Gestion des identités et des accès pour Amazon CloudWatch Network Monitor

AWS Identity and Access Management (IAM) est un AWS service qui permet à un administrateur de contrôler en toute sécurité l'accès aux ressources. AWS Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources CloudWatch du Network Monitor. IAM est un AWS service que vous pouvez utiliser sans frais supplémentaires. Vous pouvez utiliser les fonctions d'IAM pour permettre à d'autres utilisateurs, services et applications d'utiliser vos ressources AWS pleinement ou de façon limitée, sans partager vos autorisations.

Par défaut, les utilisateurs IAM ne sont pas autorisés à créer, afficher ou modifier les ressources AWS . Pour permettre à un utilisateur IAM d'accéder à des ressources, telles qu'un réseau mondial, et d'effectuer des tâches, vous devez :

- Créer une politique IAM qui accorde à l’utilisateur IAM l’autorisation d’utiliser les actions d’API et les ressources spécifiques dont il a besoin
- Attacher la politique à l’utilisateur IAM ou au groupe auquel cet utilisateur IAM appartient.

Quand vous attachez une stratégie à un utilisateur ou à un groupe d'utilisateurs, elle accorde ou refuse aux utilisateurs l'autorisation d'exécuter les tâches spécifiées sur les ressources mentionnées.

Clés de condition

L’élément `Condition` (ou le bloc Condition) vous permet de spécifier des conditions lorsqu’une instruction est appliquée. L’élément Condition est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, comme égal ou inférieur, pour faire correspondre la condition de la politique aux valeurs de la demande. Pour plus d’informations, veuillez consulter la rubrique Éléments de politique JSON IAM : Opérateurs de condition dans le Guide de l’utilisateur AWS Identity and Access Management.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l’aide d’une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l’instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM.

Vous pouvez associer des balises aux ressources CloudWatch du Network Monitor ou transmettre des balises dans une demande au Cloud WAN. Pour contrôler l'accès basé sur des balises, vous devez fournir les informations des balises dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`. Veuillez consulter la rubrique Éléments de politique JSON IAM : Condition dans le Guide de l'utilisateur AWS Identity and Access Management pour plus d'informations.

Pour voir toutes les clés de condition AWS globales, voir les clés contextuelles de condition AWS globales dans le guide de l'utilisateur d'AWS Identity and Access Management.

Étiqueter les ressources du réseau principal

Une balise est une étiquette de métadonnées que vous attribuez ou que vous AWS attribuez à une AWS ressource. Chaque balise se compose d'une clé et d'une valeur. Vous définissez la clé et la valeur des balises que vous affectez. Par exemple, vous pouvez définir la clé `purpose` et la valeur `test` pour une ressource. Les balises vous permettent d'effectuer les actions suivantes :

- Identifiez et organisez vos AWS ressources. De nombreux AWS services prennent en charge le balisage. Vous pouvez donc attribuer le même tag aux ressources de différents services pour indiquer que les ressources sont liées.
- Contrôlez l'accès à vos AWS ressources. Pour plus d'informations, consultez la section Contrôle de l'accès aux AWS ressources à l'aide de balises dans le Guide de AWS l'utilisateur de la gestion des identités et des accès.

Comment Amazon CloudWatch Network Monitor fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à CloudWatch Network Monitor, découvrez quelles fonctionnalités IAM peuvent être utilisées avec CloudWatch Network Monitor.

Fonctionnalités IAM que vous pouvez utiliser avec Amazon CloudWatch Network Monitor

Fonction IAM	CloudWatch Support pour les moniteurs réseau
Politiques basées sur l'identité	Oui

Fonction IAM	CloudWatch Support pour les moniteurs réseau
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition d’une politique	Oui
ACL	Non
ABAC (identifications dans les politiques)	Partielle
Informations d’identification temporaires	Oui
Autorisations de principal	Oui
Fonctions du service	Non
Rôles liés à un service	Oui

Pour obtenir une vue d'ensemble de la façon dont CloudWatch Network Monitor et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour Amazon Network Monitor CloudWatch

Prend en charge les politiques basées sur l’identité	Oui

Les politiques basées sur l’identité sont des documents de politique d’autorisations JSON que vous pouvez attacher à une identité telle qu’un utilisateur, un groupe d’utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l’identité, consultez Création de politiques IAM dans le Guide de l’utilisateur IAM.

Avec les politiques IAM basées sur l’identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou

refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour Network Monitor CloudWatch

Pour consulter des exemples de politiques basées sur l'identité de CloudWatch Network Monitor, consultez. Exemples de politiques basées sur l'identité pour Amazon CloudWatch

Politiques basées sur les ressources dans Network Monitor CloudWatch

Prend en charge les politiques basées sur les ressources	Non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Différence entre les rôles IAM et les politiques basées sur une ressource dans le Guide de l'utilisateur IAM.

Actions politiques pour CloudWatch Network Monitor

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des actions de CloudWatch Network Monitor, consultez la section Actions définies par Amazon CloudWatch Network Monitor dans le Service Authorization Reference.

Les actions de stratégie dans CloudWatch Network Monitor utilisent le préfixe suivant avant l'action :

```
networkmonitor
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "networkmonitor:action1",
      "networkmonitor:action2"
         ]
```

Pour consulter des exemples de politiques basées sur l'identité de CloudWatch Network Monitor, consultez. Exemples de politiques basées sur l'identité pour Amazon CloudWatch

Ressources relatives aux politiques pour CloudWatch Network Monitor

Prend en charge les ressources de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de ressources CloudWatch Network Monitor et leurs ARN, consultez la section Ressources définies par Amazon CloudWatch Network Monitor dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon CloudWatch Network Monitor.

Clés de conditions de politique pour CloudWatch Network Monitor

Prend en charge les clés de condition de politique spécifiques au service	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR`

opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de condition du moniteur CloudWatch réseau, consultez la section Clés de condition pour Amazon CloudWatch Network Monitor dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon CloudWatch Network Monitor.

ACL dans le moniteur CloudWatch réseau

Prend en charge les listes ACL	Non

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec moniteur CloudWatch réseau

Prise en charge d'ABAC (identifications dans les politiques)	Partielle

Le contrôle d'accès basé sur les attributs (ABAC) est une politique d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L’ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l’accès basé sur des étiquettes, vous devez fournir les informations d’étiquette dans l’élément de condition d’une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d’informations sur l’ABAC, consultez Qu’est-ce que le contrôle d’accès basé sur les attributs (ABAC) ? dans le Guide de l’utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l’ABAC, consultez Utilisation du contrôle d’accès par attributs (ABAC) dans le Guide de l’utilisateur IAM.

Utilisation d'informations d'identification temporaires avec CloudWatch Network Monitor

Prend en charge les informations d’identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d’identification temporaires lorsque vous vous connectez à la console en tant qu’utilisateur, puis changez de rôle. Pour plus d’informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l’utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires

au lieu d'utiliser des clés d'accès à long terme. Pour plus d’informations, consultez Informations d’identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour CloudWatch Network Monitor

Prend en charge les sessions d'accès direct (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d’autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d’accès.

Rôles de service pour CloudWatch Network Monitor

Prend en charge les fonctions de service	Non

Une fonction du service est un rôle IAM qu’un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d’IAM. Pour plus d’informations, consultez Création d’un rôle pour la délégation d’autorisations à un Service AWS dans le Guide de l’utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber les fonctionnalités CloudWatch du Moniteur réseau. Modifiez les rôles de service uniquement lorsque CloudWatch Network Monitor fournit des instructions à cet effet.

Utilisation d'un rôle lié à un service pour Network Monitor CloudWatch

Prend en charge les rôles liés à un service.	Oui

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Exemples de politiques basées sur l'identité pour Network Monitor CloudWatch

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier les ressources CloudWatch du moniteur réseau. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par CloudWatch Network Monitor, y compris le format des ARN pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour Amazon CloudWatch Network Monitor dans la référence d'autorisation de service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console CloudWatch Network Monitor
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Résolution des problèmes CloudWatch d'identité et d'accès au moniteur réseau

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources CloudWatch Network Monitor dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue.

Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console CloudWatch Network Monitor

Pour accéder à la console Amazon CloudWatch Network Monitor, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher des informations détaillées sur les ressources du moniteur CloudWatch réseau de votre Compte AWS. Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la console CloudWatch Network Monitor, associez également le CloudWatch Network Monitor *ConsoleAccess* ou la politique *ReadOnly* AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
```

```
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Résolution des problèmes CloudWatch d'identité et d'accès au moniteur réseau

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec CloudWatch Network Monitor et IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans le Moniteur CloudWatch réseau
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder aux ressources de mon moniteur CloudWatch réseau

Je ne suis pas autorisé à effectuer une action dans le Moniteur CloudWatch réseau

Si vous recevez une erreur qui indique que vous n’êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d’effectuer l’action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `networkmonitor:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 networkmonitor:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `networkmonitor:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle à CloudWatch Network Monitor.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans le Moniteur CloudWatch réseau. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite autoriser des personnes extérieures à moi Compte AWS à accéder aux ressources de mon moniteur CloudWatch réseau

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si CloudWatch Network Monitor prend en charge ces fonctionnalités, reportez-vous àComment Amazon CloudWatch travaille avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

AWS politiques gérées pour CloudWatch Network Monitor

Pour ajouter des autorisations aux utilisateurs, aux groupes et aux rôles, il est plus facile d'utiliser des politiques AWS gérées que de les rédiger vous-même. Il faut du temps et de l'expertise pour créer des politiques gérées par le client IAM qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques AWS gérées. Ces politiques couvrent les cas d'utilisation courants et sont disponibles dans votre AWS compte. Pour plus d'informations sur les politiques AWS gérées, voir les politiques AWS gérées dans le guide de l'utilisateur IAM.

AWS les services maintiennent et mettent à jour les politiques AWS gérées. Vous ne pouvez pas modifier les autorisations dans les politiques AWS gérées. Les services ajoutent occasionnellement des autorisations à une politique gérée par AWS pour prendre en charge de nouvelles fonctionnalités. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont très susceptibles de mettre à jour une politique gérée par AWS quand une nouvelle fonctionnalité est lancée ou quand de nouvelles opérations sont disponibles. Les services ne suppriment pas les autorisations d'une politique AWS gérée. Les mises à jour des politiques n'endommageront donc pas vos autorisations existantes.

En outre, AWS prend en charge les politiques gérées pour les fonctions professionnelles qui couvrent plusieurs services. Par exemple, la politique `ReadOnlyAccess` AWS gérée fournit un accès en lecture seule à tous les AWS services et ressources. Lorsqu'un service lance une nouvelle fonctionnalité, il AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page politiques gérées par AWS pour les fonctions de tâche dans le Guide de l'utilisateur IAM.

AWS politique gérée : CloudWatchNetworkMonitorServiceRolePolicy

`CloudWatchNetworkMonitorServiceRolePolicy`Il est associé à un rôle lié au service qui permet au service d'effectuer des actions en votre nom et d'accéder aux ressources associées à CloudWatch Network Monitor. Vous ne pouvez pas associer cette politique à vos identités IAM. Pour plus d'informations, consultez the section called “Rôles liés à un service”.

CloudWatch Mises à jour des politiques AWS gérées relatives à la surveillance du réseau

Consultez les détails des mises à jour des politiques AWS gérées pour la surveillance CloudWatch du réseau depuis que ce service a commencé à suivre ces modifications en novembre 2023.

Modification	Description	Date
CloudWatchNetworkMonitorServiceRolePolicy: Nouvelle politique.	Nouvelle politique ajoutée à CloudWatch Network Monitor.	27 novembre 2023
the section called “AWSServiceRoleForNetworkMonitor ” : nouveau rôle.	Nouveau rôle ajouté au Moniteur CloudWatch réseau.	27 novembre 2023

Autorisations IAM pour CloudWatch Network Monitor

Pour utiliser Amazon CloudWatch Network Monitor, les utilisateurs doivent disposer des autorisations appropriées.

Pour plus d'informations sur la sécurité sur Amazon CloudWatch, consultezGestion des identités et des accès pour Amazon CloudWatch.

Autorisations requises pour afficher un moniteur

Pour afficher un moniteur pour Amazon CloudWatch Network Monitor dans le AWS Management Console, vous devez être connecté en tant qu'utilisateur ou en tant que rôle disposant des autorisations suivantes :

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "networkmonitor:Get*",
                "networkmonitor:List*"
                ],
            "Resource": "*"
        }
    ]
}
```

Autorisations requises pour créer un moniteur

Pour créer un moniteur dans Amazon CloudWatch Network Monitor, les utilisateurs doivent être autorisés à créer un rôle lié à un service associé à Network Monitor. Pour en savoir plus sur le rôle lié au service, veuillez consulter la rubrique Utilisation d'un rôle lié à un service pour Network Monitor CloudWatch .

Pour créer un moniteur pour Amazon CloudWatch Network Monitor dans le AWS Management Console, vous devez être connecté en tant qu'utilisateur ou en tant que rôle disposant des autorisations incluses dans la politique suivante.

> **Note**
>
> Si vous créez une politique d'autorisations basée sur l'identité qui est plus restrictive, les utilisateurs tributaires de cette politique ne pourront pas créer de moniteur.

```
{
"Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "networkmonitor:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "networkmonitor.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:GetRole",
                "iam:PutRolePolicy"
            ],
            "Resource": "arn:aws:iam::*:role/aws-service-role/
networkmonitor.amazonaws.com/AWSServiceRoleForNetworkMonitor"
        },
        {
            "Action": [
                "ec2:CreateSecurityGroup",
                "ec2:CreateNetworkInterface",
                "ec2:CreateTags"
```

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

Utilisation d'un rôle lié à un service pour Network Monitor CloudWatch

Amazon CloudWatch Network Monitor utilise le rôle lié au service suivant pour les autorisations dont il a besoin pour appeler d'autres AWS services en votre nom :

- `AWSServiceRoleForNetworkMonitor`

`AWSServiceRoleForNetworkMonitor`

CloudWatch La surveillance du réseau utilise le rôle lié au service nommé `AWSServiceRoleForNetworkMonitor` pour mettre à jour et gérer les moniteurs CloudWatch réseau.

Le rôle lié à un service `AWSServiceRoleForNetworkMonitor` approuve le fait que le service suivant endosse le rôle :

- `networkmonitor.amazonaws.com`

La politique `CloudWatchNetworkMonitorServiceRolePolicy` est attachée au rôle lié au service et autorise le service à accéder aux ressources VPC et EC2 de votre compte, ainsi qu'à gérer les moniteurs réseau créés.

Groupes d'autorisations

La politique est regroupée dans les ensembles d'autorisations suivants :

- **`cloudwatch`**- Cela permet au directeur du service de publier les métriques de surveillance du réseau sur les CloudWatch ressources.
- **`ec2`** : permet au principal du service de décrire les VPC et les sous-réseaux de votre compte afin de créer ou de mettre à jour des moniteurs et des sondes. Cela permet également au principal du service de créer, de modifier et de supprimer des groupes de sécurité, des interfaces réseau et leurs autorisations associées afin de configurer le moniteur ou la sonde afin d'envoyer le trafic de surveillance à vos points de terminaison.

Pour de plus amples informations sur la politique, veuillez consulter la rubrique the section called “AWS politiques gérées”.

L’exemple suivant montre la politique `CloudWatchNetworkMonitorServiceRolePolicy` :

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "PublishCw",
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "cloudwatch:namespace": "AWS/NetworkMonitor"
    }
   }
  },
  {
   "Sid": "DescribeAny",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribeNetworkInterfaceAttribute",
    "ec2:DescribeVpcs",
    "ec2:DescribeNetworkInterfacePermissions",
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups"
   ],
   "Resource": "*"
  },
  {
   "Sid": "DeleteModifyEc2Resources",
   "Effect": "Allow",
   "Action": [
    "ec2:AuthorizeSecurityGroupEgress",
    "ec2:CreateNetworkInterfacePermission",
    "ec2:DeleteNetworkInterfacePermission",
    "ec2:RevokeSecurityGroupEgress",
    "ec2:ModifyNetworkInterfaceAttribute",
    "ec2:DeleteNetworkInterface",
    "ec2:DeleteSecurityGroup"
   ],
```

```
    "Resource": [
     "arn:aws:ec2:*:*:network-interface/*",
     "arn:aws:ec2:*:*:security-group/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/ManagedByCloudWatchNetworkMonitor": "true"
     }
    }
   }
  ]
}
```

Création du rôle lié à un service

`AWSServiceRoleForNetworkMonitor`

Vous n'avez pas besoin de créer manuellement le rôle lié à un service `AWSServiceRoleForNetworkMonitor`.

- CloudWatch Network Monitor crée le `AWSServiceRoleForNetworkMonitor` rôle lorsque vous créez votre premier moniteur réseau. Ce rôle s'appliquera à tous les moniteurs que vous créerez ultérieurement.

Pour qu'un rôle lié à un service puisse être créé en votre nom, vous devez avoir les autorisations requises. Pour de plus amples informations, veuillez consulter Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM

Modification du rôle lié à un service

Vous pouvez modifier la description de `AWSServiceRoleForNetworkMonitor` à l'aide d'IAM. Pour de plus amples informations, veuillez consulter Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Suppression du rôle lié à un service

Si vous n'avez plus besoin d'utiliser le Moniteur CloudWatch réseau, nous vous recommandons de supprimer le `AWSServiceRoleForNetworkMonitor` rôle.

Vous ne pouvez supprimer ces rôles liés au service qu'après avoir supprimé votre moniteur réseau. Pour plus d'informations sur la suppression de votre moniteur réseau, veuillez consulter la rubrique Supprimer un moniteur réseau.

Vous pouvez utiliser la console IAM, l'IAM CLI ou l'IAM API pour supprimer les rôles liés aux services. Pour de plus amples informations, veuillez consulter Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Une fois que vous aurez supprimé, `AWSServiceRoleForNetworkMonitor` CloudWatch Network Monitor créera à nouveau le rôle lorsque vous créerez un nouveau moniteur.

Régions prises en charge pour le CloudWatch rôle lié au service Network Monitor

CloudWatch Network Monitor prend en charge le rôle lié au service partout Régions AWS où le service est disponible. Pour plus d'informations, veuillez consulter la rubrique Points de terminaison AWS dans le Références générales AWS.

Supprimer le rôle lié à un service

Si vous n'avez plus besoin d'utiliser le Moniteur CloudWatch réseau, nous vous recommandons de supprimer le `AWSServiceRoleForNetworkMonitor` rôle.

Vous ne pouvez supprimer ces rôles liés au service qu'après avoir supprimé votre moniteur réseau. Pour plus d'informations sur la suppression de votre moniteur réseau, veuillez consulter la rubrique Supprimer un moniteur réseau.

Vous pouvez utiliser la console IAM, l'IAM CLI ou l'IAM API pour supprimer les rôles liés aux services. Pour de plus amples informations, veuillez consulter Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Une fois que vous aurez supprimé, `AWSServiceRoleForNetworkMonitor` CloudWatch Network Monitor créera à nouveau le rôle lorsque vous créerez un nouveau moniteur.

Tarification

Avec Amazon CloudWatch Network Monitor, il n'y a aucun coût initial ni aucun engagement à long terme. La tarification du Moniteur réseau comprend les deux éléments suivants :

- une redevance horaire par ressource surveillée, et
- CloudWatch frais de métriques.

Lorsque vous créez un moniteur réseau, vous lui associez des ressources à surveiller. Pour Network Monitor, il s'agira de sous-réseaux de votre Amazon Virtual Private Cloud (VPC). Chaque ressource surveillée vous permet de créer jusqu'à quatre sondes à partir de chaque sous-réseau de vos VPC

vers quatre destinations. Pour mieux contrôler votre facture, vous pouvez ajuster la couverture de votre sous-réseau et la couverture IP sur site en réduisant le nombre de ressources surveillées.

Pour plus d'informations sur les tarifs, consultez la page de CloudWatch tarification d'Amazon.

Surveillance de l’infrastructure

Les rubriques de cette section décrivent les CloudWatch fonctionnalités qui peuvent vous aider à obtenir une visibilité opérationnelle sur vos AWS ressources.

Rubriques

Container Insights

Utilisez CloudWatch Container Insights pour collecter, agréger et résumer les métriques et les journaux de vos applications conteneurisées et de vos microservices. Container Insights est disponible pour les plateformes Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) et Kubernetes sur Amazon EC2. Container Insights permet de collecter des métriques à partir AWS Fargate de clusters déployés sur Amazon ECS et Amazon EKS.

CloudWatch collecte automatiquement des métriques pour de nombreuses ressources, telles que le processeur, la mémoire, le disque et le réseau. Conteneur Insights fournit également des informations de diagnostic (par exemple sur les échecs de redémarrage des conteneurs) pour vous aider à isoler les problèmes et à les résoudre rapidement. Vous pouvez également définir des CloudWatch alarmes sur les métriques collectées par Container Insights.

Container Insights collecte des données en tant qu'événements du journal des performances utilisant le format de métrique intégrée. Ces évènements du journal des performances sont des entrées qui utilisent un schéma JSON structuré qui permet aux données à haute cardinalité d'être intégrées et stockées à grande échelle. À partir de ces données, CloudWatch crée des métriques agrégées au niveau du cluster, du nœud, du pod, de la tâche et du service sous forme de CloudWatch métriques. Les métriques collectées par Container Insights sont disponibles dans des tableaux de bord CloudWatch automatiques et peuvent également être consultées dans la section Metrics de la CloudWatch console. Les métriques ne sont pas visibles tant que les tâches du conteneur ne sont pas en cours d'exécution depuis un certain temps.

Lorsque vous déployez Container Insights, il crée automatiquement un groupe de journaux pour les événements du journal des performances. Il n'est pas nécessaire de créer ce groupe de journaux vous-même.

Pour vous aider à gérer vos coûts, Container Insights CloudWatch ne crée pas automatiquement tous les indicateurs possibles à partir des données du journal. Cependant, vous pouvez consulter des mesures supplémentaires et des niveaux de granularité supplémentaires en utilisant CloudWatch Logs Insights pour analyser les événements bruts du journal des performances.

Dans la version originale de Container Insights, les métriques collectées et les logs ingérés sont facturés en tant que métriques personnalisées. Grâce à Container Insights avec observabilité améliorée pour Amazon EKS, les métriques et les journaux de Container Insights sont facturés par observation au lieu d'être facturés par métrique stockée ou par journal ingéré. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Dans Amazon EKS et Kubernetes, Container Insights utilise une version conteneurisée de l' CloudWatch agent pour découvrir tous les conteneurs en cours d'exécution dans un cluster. Il collecte ensuite les données de performance à chaque couche de la pile de performances.

Container Insights prend en charge le AWS KMS key chiffrement grâce aux journaux et aux métriques qu'il collecte. Pour activer ce chiffrement, vous devez activer manuellement le AWS KMS chiffrement pour le groupe de journaux qui reçoit les données Container Insights. Container Insight chiffre alors ces données à l'aide de la clé KMS fournie. Seules les clés symétriques sont prises en charge. N'utilisez pas de clés KMS asymétriques pour chiffrer vos groupes de journaux.

Pour plus d'informations, voir Chiffrer les données des journaux dans les CloudWatch journaux à l'aide AWS KMS de.

Container Insights avec observabilité améliorée pour Amazon EKS

Le 6 novembre 2023, une nouvelle version de Container Insights a été publiée. Cette version prend en charge une observabilité améliorée pour les clusters Amazon EKS s'exécutant sur Amazon EC2 et peut collecter des métriques plus détaillées à partir de ces clusters. Après l'installation, il collecte automatiquement des données télémétriques détaillées sur l'infrastructure et des journaux de conteneurs pour vos clusters Amazon EKS. Vous pouvez ensuite utiliser des tableaux de bord élaborés et immédiatement exploitables pour approfondir la télémétrie des applications et des infrastructures.

Container Insights avec observabilité améliorée pour Amazon EKS collecte des métriques de santé, de performance et d'état granulaires jusqu'au niveau du conteneur, ainsi que des métriques du plan

de contrôle. Pour plus d'informations sur les métriques et dimensions supplémentaires collectées, veuillez consulter Métriques Container Insights pour Amazon EKS et Kubernetes.

Si vous avez installé Container Insights en utilisant l' CloudWatch agent sur un cluster Amazon EKS sur Amazon EC2 après le 6 novembre 2023, vous disposez de Container Insights avec une observabilité améliorée pour Amazon EKS. Sinon, vous pouvez mettre à niveau un cluster Amazon EKS vers cette nouvelle version en suivant les instructions figurant dans Mise à niveau vers Container Insights avec observabilité améliorée pour Amazon EKS.

Container Insights favorise l' CloudWatch observabilité entre comptes. Vous utilisez un seul compte de surveillance pour surveiller et dépanner vos applications qui couvrent plusieurs AWS comptes au sein d'une même région. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

Container Insights avec une observabilité améliorée pour Amazon EKS prend également en charge les nœuds de travail Windows.

Container Insights avec observabilité améliorée pour Amazon EKS n'est pas pris en charge sur Fargate.

> **Note**
>
> Pour savoir si vos clusters peuvent être mis à niveau vers Container Insights avec observabilité améliorée pour Amazon EKS, accédez à la console Container Insights. Pour ce faire, choisissez Insights, Container Insights dans le volet de navigation de la CloudWatch console. Dans la console Container Insights, une bannière vous indique si des clusters Amazon EKS peuvent être mis à niveau, ainsi que des liens vers la page de mise à niveau.

Plateformes prises en charge

Container Insights est disponible pour les plateformes Amazon Elastic Container Service, Amazon Elastic Kubernetes Service et Kubernetes sur les instances Amazon EC2.

- Pour Amazon ECS, Container Insights collecte des métriques au niveau du cluster, des tâches et des services sur les instances Linux et Windows Server. Il ne peut collecter des métriques au niveau de l'instance que sur les instances Linux.

Pour Amazon ECS, les métriques réseau sont disponibles uniquement pour les conteneurs en mode réseau `bridge` et en mode réseau. `awsvpc`. Ils ne sont pas disponibles pour les conteneurs en mode réseau `host`.

- Pour Amazon Elastic Kubernetes Service et les plateformes Kubernetes sur les instances Amazon EC2, Container Insights est pris en charge uniquement sur les instances Linux.

CloudWatch image du conteneur de l'agent

Amazon fournit une image de conteneur d' CloudWatch agent sur Amazon Elastic Container Registry. Pour plus d'informations, consultez le référentiel cloudwatch-agent sur Amazon ECR.

Régions prises en charge

Container Insights pour Amazon ECS est pris en charge dans les régions suivantes :

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Californie du Nord)
- USA Ouest (Oregon)
- Afrique (Le Cap)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Hyderabad)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Osaka)
- Asia Pacific (Seoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Tokyo)
- Asie-Pacifique (Sydney)
- Canada Ouest (Calgary)
- Canada (Centre)

- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Paris)
- Europe (Espagne)
- Europe (Stockholm)
- Europe (Zurich)
- Moyen-Orient (Bahreïn)
- Moyen-Orient (EAU)
- Amérique du Sud (São Paulo)
- AWS GovCloud (USA Est)
- AWS GovCloud (US-Ouest)
- Chine (Beijing)
- Chine (Ningxia)

Régions prises en charge pour Amazon EKS et Kubernetes

Container Insights pour Amazon EKS et Kubernetes est pris en charge dans les régions suivantes :

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Californie du Nord)
- US West (Oregon)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)

- Canada (Centre)
- Chine (Beijing)
- China (Ningxia)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Paris)
- Europe (Stockholm)
- Moyen-Orient (Bahreïn)
- Amérique du Sud (Sao Paulo)
- AWS GovCloud (USA Est)
- AWS GovCloud (US-Ouest)

Configuration de Container Insights

Le processus d'installation de Container Insights est différent pour Amazon ECS, Amazon EKS et Kubernetes.

Rubriques

- Configuration de Container Insights sur Amazon ECS
- Configuration de Container Insights sur Amazon EKS et Kubernetes

Configuration de Container Insights sur Amazon ECS

Vous pouvez utiliser l'une des options suivantes ou les deux pour activer Container Insights sur les clusters Amazon ECS :

- Utilisez le AWS Management Console ou AWS CLI pour commencer à collecter des métriques au niveau du cluster, au niveau des tâches et au niveau du service.
- Déployez l' CloudWatch agent en tant que service daemon pour commencer à collecter des métriques au niveau de l'instance sur les clusters hébergés sur des instances Amazon EC2.

Rubriques

- Configuration de Container Insights sur Amazon ECS pour les métriques de niveau de cluster et de niveau de service
- Configuration de Container Insights sur Amazon ECS à l'aide de AWS Distro pour OpenTelemetry
- Déploiement de l' CloudWatch agent pour collecter des métriques au niveau de l'instance EC2 sur Amazon ECS
- Déploiement de la AWS distribution pour collecter des métriques OpenTelemetry au niveau de l'instance EC2 sur des clusters Amazon ECS
- Configurer FireLens pour envoyer des journaux à CloudWatch Logs

Configuration de Container Insights sur Amazon ECS pour les métriques de niveau de cluster et de niveau de service

Vous pouvez activer Container Insights sur les clusters Amazon ECS nouveaux et existants. Container Insights collecte des métriques au niveau du cluster, des tâches et des services. Vous pouvez activer Container Insights à l'aide de la console Amazon ECS ou du AWS CLI.

Si vous utilisez Amazon ECS sur une instance Amazon EC2 pour collecter les métriques réseau et stockage de Container Insights, vous devez lancer cette instance à l'aide d'une AMI qui inclut l'agent Amazon ECS version 1.29. Pour plus d'informations sur la mise à jour de la version de votre agent, consultez Mise à jour de l'agent du conteneur Amazon ECS

Vous pouvez utiliser le AWS CLI pour définir une autorisation au niveau du compte afin d'activer Container Insights pour tout nouveau cluster Amazon ECS créé dans votre compte. Pour ce faire, entrez la commande suivante.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

> **Note**
>
> Si la AWS KMS clé gérée par le client que vous utilisez pour vos métriques Amazon ECS Container Insights n'est pas déjà configurée pour fonctionner CloudWatch, vous devez mettre à jour la politique relative aux clés pour autoriser les journaux chiffrés dans CloudWatch les journaux. Vous devez également associer votre propre AWS KMS clé au groupe de journaux situé en dessous`/aws/ecs/containerinsights/`*`ClusterName`*`/performance`. Pour plus d'informations, voir Chiffrer les données des journaux dans les CloudWatch journaux à l'aide AWS Key Management Service de.

Configuration de Container Insights sur les clusters Amazon ECS existants

Pour activer Container Insights sur un cluster Amazon ECS existant, saisissez la commande suivante. Vous devez exécuter la version 1.16.200 ou ultérieure AWS CLI pour que la commande suivante fonctionne.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Configuration de Container Insights sur les nouveaux clusters Amazon ECS

Il existe deux façons d'activer Container Insights sur de nouveaux clusters Amazon ECS. Vous pouvez configurer Amazon ECS de sorte que tous les nouveaux clusters soient activés par défaut pour Container Insights. Sinon, vous pouvez activer un nouveau cluster lorsque vous le créez.

Utilisation de l’ AWS Management Console

Vous pouvez activer Container Insights sur tous les nouveaux clusters par défaut ou sur un seul cluster lors de sa création.

Pour activer Container Insights sur tous les nouveaux clusters par défaut

1. Ouvrez la console à partir de l'adresse https://console.aws.amazon.com/ecs/v2.
2. Dans la page de navigation, choisissez Account Settings (Paramètres du compte).
3. Choisissez Mettre à jour.
4. Pour utiliser CloudWatch Container Insights par défaut pour les clusters, sous CloudWatchContainer Insights, sélectionnez ou décochez CloudWatch Container Insights.
5. Sélectionnez Enregistrer les modifications.

Si vous n'avez pas utilisé la procédure précédente pour activer Container Insights sur tous les nouveaux clusters par défaut, vous pouvez utiliser la procédure suivante pour créer un cluster pour lequel Container Insights est activé.

Pour créer un cluster avec Container Insights activé

1. Ouvrez la console à partir de l'adresse https://console.aws.amazon.com/ecs/v2.
2. Dans le panneau de navigation, choisissez Clusters.
3. Sur la page Clusters, choisissez Create Cluster (Créer un cluster).

4. Sous Cluster configuration (Configuration de cluster), pour Cluster name (Nom du cluster), saisissez un nom unique.

   Le nom peut contenir jusqu'à 255 lettres (minuscules et majuscules), des chiffres et des traits d'union.

5. Pour activer Container Insights, développez Surveillance, puis activez Utiliser Container Insights.

Vous pouvez désormais créer des définitions de tâche, exécuter des tâches et lancer des services dans le cluster. Pour plus d'informations, consultez les ressources suivantes :

- Création d'une définition de tâche
- Exécution de tâches
- Création d'un service

Configuration de Container Insights sur les nouveaux clusters Amazon ECS à l'aide du AWS CLI

Pour activer Container Insights sur tous les nouveaux clusters par défaut, entrez la commande suivante.

```
aws ecs put-account-setting --name "containerInsights" --value "enabled"
```

Si vous n'avez pas utilisé la commande précédente pour activer Container Insights sur tous les nouveaux clusters par défaut, entrez la commande suivante pour créer un cluster avec Container Insights activé. Vous devez exécuter la version 1.16.200 ou une version ultérieure de l' AWS CLI pour que la commande suivante fonctionne.

```
aws ecs create-cluster --cluster-name myCICluster --settings
 "name=containerInsights,value=enabled"
```

Désactivation de Container Insights sur des clusters Amazon ECS

Pour désactiver Container Insights sur un cluster Amazon ECS existant, saisissez la commande suivante.

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=disabled
```

Configuration de Container Insights sur Amazon ECS à l'aide de AWS Distro pour OpenTelemetry

Utilisez cette section si vous souhaitez utiliser AWS Distro pour OpenTelemetry configurer CloudWatch Container Insights sur un cluster Amazon ECS. Pour plus d'informations sur AWS Distro for Open Telemetry, voir AWS Distro for. OpenTelemetry

Cette procédure suppose que vous avez déjà un cluster exécutant Amazon ECS. Pour plus d'informations sur l'utilisation de AWS Distro pour la télémétrie ouverte avec Amazon ECS et sur la configuration d'un cluster Amazon ECS à cette fin, consultez Configuration de AWS Distro pour Collector OpenTelemetry dans Amazon Elastic Container Service.

Étape 1 : Création d'un rôle de tâche

La première étape consiste à créer un rôle de tâche dans le cluster que le AWS OpenTelemetry Collector utilisera.

Pour créer un rôle de tâche pour AWS Distro for OpenTelemetry

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Politiques, puis Créer une politique.
3. Choisissez l'onglet JSON et copiez la politique suivante :

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

4. Choisissez Examiner une politique.

5. Pour Name (Nom), saisissez **`AWSDistroOpenTelemetryPolicy`**, puis choisissez Create policy (Créer une politique).
6. Dans le panneau de navigation, choisissez Roles (Rôles), puis Create role (Créer un rôle).
7. Dans la liste des services, choisissez Elastic Container Service.
8. Plus bas sur la page, choisissez Elastic Container Service Task (Tâche Elastic Container Service), puis choisissez Next: Permissions (Suivant : Autorisations).
9. Dans la liste des politiques, recherchez AWSDistroOpenTelemetryPolicy.
10. Cochez la case à côté de AWSDistroOpenTelemetryPolicy.
11. Sélectionnez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
12. Pour Role name (Nom du rôle), saisissez **`AWSOpenTelemetryTaskRole`**, puis choisissez Create role (Créer un rôle).

Étape 2 : Créer un rôle d'exécution de tâche

L'étape suivante consiste à créer un rôle d'exécution de tâche pour le AWS OpenTelemetry collecteur.

Pour créer un rôle d'exécution de tâches pour AWS Distro for OpenTelemetry

1. Ouvrez la console IAM à l’adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles), puis Create role (Créer un rôle).
3. Dans la liste des services, choisissez Elastic Container Service.
4. Plus bas sur la page, choisissez Elastic Container Service Task (Tâche Elastic Container Service), puis choisissez Next: Permissions (Suivant : Autorisations).
5. Dans la liste des politiques, recherchez AmazonECS, TaskExecutionRolePolicy puis cochez la case à côté d'AmazonECS. TaskExecutionRolePolicy
6. Dans la liste des politiques, recherchez CloudWatchLogsFullAccesspuis cochez la case à côté de CloudWatchLogsFullAccess.
7. Dans la liste des politiques, recherchez AmazonSSM, ReadOnlyAccess puis cochez la case à côté d'AmazonSSM. ReadOnlyAccess
8. Sélectionnez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
9. Pour Role name (Nom du rôle), saisissez **`AWSOpenTelemetryTaskExecutionRole`**, puis choisissez Create role (Créer un rôle).

Étape 3 : Créer une définition de tâche

L'étape suivante consiste à créer une définition de tâche.

Pour créer une définition de tâche pour AWS Distro for OpenTelemetry

1. Ouvrez la console à partir de l'adresse https://console.aws.amazon.com/ecs/v2.
2. Dans le panneau de navigation, choisissez Task definitions (Définition des tâches)
3. Choisissez Create new task definition (Créer une nouvelle définition de tâche), puis Create new task definition (Créer une nouvelle définition de tâche).
4. Pour Task definition family (Famille de définition de tâche), spécifiez un nom unique pour la définition de tâche.
5. Configurez vos conteneurs, puis choisissez Suivant.
6. Sous Métriques et journalisation, sélectionnez Utiliser la collecte de métriques.
7. Choisissez Suivant.
8. Choisissez Créer.

Pour plus d'informations sur l'utilisation du AWS OpenTelemetry collecteur avec Amazon ECS, consultez Configuration de AWS Distro pour OpenTelemetry Collector dans Amazon Elastic Container Service.

Étape 4 : Exécuter la tâche

La dernière étape consiste à exécuter la tâche que vous avez créée.

Pour exécuter la tâche pour AWS Distro for OpenTelemetry

1. Ouvrez la console à partir de l'adresse https://console.aws.amazon.com/ecs/v2.
2. Dans le panneau de navigation de gauche, choisissez Task Definitions (Définitions de tâche) puis sélectionnez la tâche que vous venez de créer.
3. Choisissez Actions, Déployer, Exécuter la tâche.
4. Choisissez Deploy (Déploiement), Run task (Exécution de tâche).
5. Dans la section Options de calcul, dans Cluster existant, sélectionnez le cluster.
6. Choisissez Créer.
7. Ensuite, vous pouvez vérifier les nouvelles mesures dans la CloudWatch console.

8. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
9. Dans le panneau de navigation de gauche, choisissez Metrics (Métriques).

   Vous devriez voir un espace de ContainerInsights noms ECS/. Choisissez cet espace de noms, vous devriez voir huit métriques.

Déploiement de l' CloudWatch agent pour collecter des métriques au niveau de l'instance EC2 sur Amazon ECS

Pour déployer l' CloudWatch agent afin de collecter des métriques au niveau de l'instance à partir de clusters Amazon ECS hébergés sur une instance EC2, utilisez une configuration de démarrage rapide avec une configuration par défaut ou installez l'agent manuellement pour pouvoir le personnaliser.

Les deux méthodes nécessitent que vous disposiez déjà d'au moins un cluster Amazon ECS déployé avec un type de lancement EC2 et que le conteneur de l' CloudWatch agent ait accès au service de métadonnées d'instance Amazon EC2 (IMDS). Pour plus d'informations sur IMDS, consultez Métadonnées d'instance et données utilisateur.

Ces méthodes supposent également que vous avez AWS CLI installé le. En outre, pour exécuter les commandes décrites dans les procédures suivantes, vous devez être connecté à un compte ou à un rôle soumis aux politiques IAM FullAccess et FullAccessAmazonECS_.

Rubriques

- Configuration rapide à l'aide de AWS CloudFormation
- Configuration manuelle et personnalisée

Configuration rapide à l'aide de AWS CloudFormation

Pour utiliser la configuration rapide, entrez la commande suivante à utiliser AWS CloudFormation pour installer l'agent. Remplacez *`cluster-name`* et *`cluster-region`* par le nom et la région de votre cluster Amazon ECS.

Cette commande crée les rôles IAM CWagentecs et CWagentecs TaskRole. ExecutionRole Si ces rôles existent déjà dans votre compte, utilisez `ParameterKey=CreateIAMRoles,ParameterValue=False` plutôt que `ParameterKey=CreateIAMRoles,ParameterValue=True` lorsque vous entrez la commande. Sinon, la commande échouera.

```
ClusterName=cluster-name
Region=cluster-region
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

(Alternative) Utilisation de vos propres rôles IAM

Si vous souhaitez utiliser votre propre rôle de tâche ECS personnalisé et votre rôle d'exécution de tâche ECS au lieu des ExecutionRole rôles CWagentecs TaskRole et CWagentecs, assurez-vous d'abord que le rôle à utiliser en tant que rôle de tâche ECS est attaché. CloudWatchAgentServerPolicy Assurez-vous également que le rôle à utiliser comme rôle d'exécution de tâches ECS est associé à la fois aux politiques AmazonECS CloudWatchAgentServerPolicyet aux TaskExecutionRolePolicy politiques AmazonECS. Entrez ensuite la commande suivante. Dans la commande, remplacez *task-role-arn*par l'ARN de votre rôle de tâche ECS personnalisé et remplacez *execution-role-arn*par l'ARN de votre rôle d'exécution de tâche ECS personnalisé.

```
ClusterName=cluster-name
Region=cluster-region
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cloudformation-quickstart/cwagent-ecs-instance-metric-cfn.json
aws cloudformation create-stack --stack-name CWAgentECS-${ClusterName}-${Region} \
    --template-body file://cwagent-ecs-instance-metric-cfn.json \
    --parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
                 ParameterKey=TaskRoleArn,ParameterValue=${TaskRoleArn} \
                 ParameterKey=ExecutionRoleArn,ParameterValue=${ExecutionRoleArn} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${Region}
```

Dépannage de la configuration rapide

Pour vérifier l'état de la AWS CloudFormation pile, entrez la commande suivante.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stacks --stack-name CWAgentECS-$ClusterName-$Region --
region $Region
```

Si le `StackStatus` ne correspond pas à `CREATE_COMPLETE` ou `CREATE_IN_PROGRESS`, vérifiez les événements de la pile pour trouver l'erreur. Entrez la commande suivante.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name CWAgentECS-$ClusterName-$Region
 --region $Region
```

Pour vérifier l'état du service de démon `cwagent`, saisissez la commande suivante. Dans la sortie, vous devriez voir que `runningCount` est égal à `desiredCount` dans la section `deployment`. Si elle n'est pas égale, vérifiez la section `failures` dans la sortie.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services cwagent-daemon-service --cluster $ClusterName --
region $Region
```

Vous pouvez également utiliser la console CloudWatch Logs pour consulter le journal de l'agent. Recherchez le groupe de ecs-cwagent-daemon-service journaux /ecs/.

Supprimer la AWS CloudFormation pile pour l' CloudWatch agent

Si vous devez supprimer la AWS CloudFormation pile, entrez la commande suivante.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation delete-stack --stack-name CWAgentECS-${ClusterName}-${Region} --
region ${Region}
```

Configuration manuelle et personnalisée

Suivez les étapes décrites dans cette section pour déployer manuellement l' CloudWatch agent afin de collecter des métriques au niveau de l'instance à partir de vos clusters Amazon ECS hébergés sur des instances EC2.

Rôles et politiques IAM nécessaires

Deux rôles IAM sont requis. Vous devez les créer s'ils n'existent pas déjà. Pour plus d'informations sur ces rôles, consultez Rôles IAM des tâches et Rôle d'exécution des tâches Amazon ECS.

- Rôle de tâche ECS, utilisé par l' CloudWatch agent pour publier des métriques. Si ce rôle existe déjà, vous devez vous assurer que la politique `CloudWatchAgentServerPolicy` est attachée.
- Rôle d'exécution de tâche ECS, utilisé par l'agent Amazon ECS pour lancer l' CloudWatch agent. Si ce rôle existe déjà, vous devez vous assurer que les politiques `AmazonECSTaskExecutionRolePolicy` et `CloudWatchAgentServerPolicy` sont attachées.

Si vous ne disposez pas déjà de ces rôles, vous pouvez utiliser les commandes suivantes pour les créer et joindre les politiques nécessaires. Cette première commande crée le rôle de tâches ECS.

```
aws iam create-role --role-name CWAgentECSTaskRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Après avoir saisi la commande précédente, notez que la valeur `Arn` de la sortie de commande est « TaskRoleArn ». Vous devrez l'utiliser plus tard lorsque vous utiliserez la définition de tâche. Ensuite, entrez la commande suivante pour joindre les politiques nécessaires.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSTaskRole
```

Cette commande suivante crée le rôle d'exécution de tâche ECS.

```
aws iam create-role --role-name CWAgentECSExecutionRole \
    --assume-role-policy-document "{\"Version\": \"2012-10-17\",\"Statement\": [{\"Sid
\": \"\",\"Effect\": \"Allow\",\"Principal\": {\"Service\": \"ecs-tasks.amazonaws.com
\"},\"Action\": \"sts:AssumeRole\"}]}"
```

Après avoir saisi la commande précédente, notez que la valeur `Arn` de la sortie de commande est « ExecutionRoleArn ». Vous devrez l'utiliser plus tard lorsque vous utiliserez la définition de tâche. Ensuite, entrez les commandes suivantes pour joindre les politiques nécessaires.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
CloudWatchAgentServerPolicy \
    --role-name CWAgentECSExecutionRole

aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy \
    --role-name CWAgentECSExecutionRole
```

Créer la définition de tâche et lancer le service de démon

Créez une définition de tâche et utilisez-la pour lancer l' CloudWatch agent en tant que service daemon. Pour créer la définition de tâche, entrez la commande suivante. Dans les premières lignes, remplacez les espaces réservés par les valeurs réelles de votre déploiement. *logs-region* est la région où se trouve CloudWatch Logs, et *cluster-region est la région* où se trouve votre cluster. *task-role-arn*est l'Arn du rôle de tâche ECS que vous utilisez et *execution-role-arn*l'Arn du rôle d'exécution de tâche ECS.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-
ecs-instance-metric/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-json
```

Ensuite, exécutez la commande suivante pour lancer le service de démon. Remplacez *cluster-name* et *cluster-region* par le nom et la région de votre cluster Amazon ECS.

> **Important**
>
> Supprimez toutes les stratégies relatives aux fournisseurs de capacité avant d'exécuter cette commande. Sinon, la commande ne fonctionnera pas.

```
ClusterName=cluster-name
Region=cluster-region
```

```
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

Si vous voyez le message d'erreur `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, cela signifie que vous avez déjà créé un service de démon nommé `cwagent-daemon-service`. Vous devez d'abord supprimer ce service, en utilisant la commande suivante comme exemple.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

(En option) Configuration avancée

Vous pouvez éventuellement utiliser SSM pour spécifier d'autres options de configuration pour l' CloudWatchagent dans vos clusters Amazon ECS hébergés sur des instances EC2. Les options sont les suivantes :

- `metrics_collection_interval`— Fréquence en secondes à laquelle l' CloudWatch agent collecte des métriques. La valeur par défaut est 60. La plage est comprise entre 1 et 172 000.
- `endpoint_override` – (En option) Spécifie un point de terminaison différent vers lequel envoyer les journaux. Vous pouvez effectuer cette opération si vous publiez à partir d'un cluster dans un VPC et souhaitez que les données des journaux soient transmises à un point de terminaison d'un VPC.

  La valeur de `endpoint_override` doit être une chaîne qui est une URL.
- `force_flush_interval` – Spécifie en secondes la durée maximale pendant laquelle les journaux demeurent dans la mémoire tampon avant d'être envoyés au serveur. Quelle que soit la configuration de ce champ, si la taille des journaux dans la mémoire tampon atteint 1 Mo, les journaux sont immédiatement envoyés au serveur. La valeur par défaut est de 5 secondes.

- `region` – Par défaut, l'agent publie des métriques dans la même région que celle où se trouve l'instance de conteneur Amazon ECS. Pour remplacer cela, vous pouvez spécifier une autre région ici. Par exemple, `"region" : "us-east-1"`

Voici un exemple de configuration personnalisée :

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "ecs": {
                "metrics_collection_interval": 30
            }
        },
        "force_flush_interval": 5
    }
}
```

Pour personnaliser la configuration de votre CloudWatch agent dans vos conteneurs Amazon ECS

1. Assurez-vous que la ReadOnlyAccess politique AmazonSSM est attachée à votre rôle d'exécution de tâches Amazon ECS. Vous pouvez entrer la commande suivante pour ce faire. Cet exemple suppose que votre rôle d'exécution de tâches Amazon ECS est ExecutionRole CWagentecs. Si vous utilisez un rôle différent, remplacez ce nom dans la commande suivante.

   ```
   aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/
   AmazonSSMReadOnlyAccess \
           --role-name CWAgentECSExecutionRole
   ```

2. Créez le fichier de configuration personnalisé similaire à l'exemple précédent. Renommez le fichier `/tmp/ecs-cwagent-daemon-config.json`.
3. Exécutez la commande suivante pour placer cette configuration dans le Parameter Store. Remplacez *`cluster-region`* par la région de votre cluster Amazon ECS. Pour exécuter cette commande, vous devez être connecté à un utilisateur ou à un rôle soumis à la politique AmazonSSM FullAccess.

   ```
   Region=cluster-region
   aws ssm put-parameter \
   ```

```
    --name "ecs-cwagent-daemon-service" \
    --type "String" \
    --value "`cat /tmp/ecs-cwagent-daemon-config.json`" \
    --region $Region
```

4. Téléchargez le fichier de définition de tâche dans un fichier local, tel que `/tmp/cwagent-ecs-instance-metric.json`

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/daemon-service/cwagent-ecs-instance-metric/cwagent-ecs-instance-metric.json -o /tmp/cwagent-ecs-instance-metric.json
```

5. Modifiez le fichier de définition de tâche. Supprimez la section suivante :

```
"environment": [
                {
                    "name": "USE_DEFAULT_CONFIG",
                    "value": "True"
                }
            ],
```

Remplacer cette section par ce qui suit :

```
"secrets": [
                {
                    "name": "CW_CONFIG_CONTENT",
                    "valueFrom": "ecs-cwagent-daemon-service"
                }
            ],
```

6. Redémarrez l'agent en tant que service de démon en procédant comme suit :

   a. Exécutez la commande suivante.

```
TaskRoleArn=task-role-arn
ExecutionRoleArn=execution-role-arn
AWSLogsRegion=logs-region
Region=cluster-region
cat /tmp/cwagent-ecs-instance-metric.json \
    | sed "s|{{task-role-arn}}|${TaskRoleArn}|;s|{{execution-role-arn}}|
${ExecutionRoleArn}|;s|{{awslogs-region}}|${AWSLogsRegion}|" \
```

```
    | xargs -0 aws ecs register-task-definition --region ${Region} --cli-input-
json
```

b. Exécutez la commande suivante pour lancer le service de démon. Remplacez *cluster-name* et *cluster-region* par le nom et la région de votre cluster Amazon ECS.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs create-service \
    --cluster ${ClusterName} \
    --service-name cwagent-daemon-service \
    --task-definition ecs-cwagent-daemon-service \
    --scheduling-strategy DAEMON \
    --region ${Region}
```

Si vous voyez le message d'erreur `An error occurred (InvalidParameterException) when calling the CreateService operation: Creation of service was not idempotent`, cela signifie que vous avez déjà créé un service de démon nommé `cwagent-daemon-service`. Vous devez d'abord supprimer ce service, en utilisant la commande suivante comme exemple.

```
ClusterName=cluster-name
Region=Region
aws ecs delete-service \
    --cluster ${ClusterName} \
    --service cwagent-daemon-service \
    --region ${Region} \
    --force
```

Déploiement de la AWS distribution pour collecter des métriques OpenTelemetry au niveau de l'instance EC2 sur des clusters Amazon ECS

Suivez les étapes décrites dans cette section pour utiliser AWS Distro afin de collecter des OpenTelemetry métriques au niveau de l'instance EC2 sur un cluster Amazon ECS. Pour plus d'informations sur la AWS distribution pour OpenTelemetry, consultez la section AWS Distribution pour. OpenTelemetry

Cette procédure suppose que vous avez déjà un cluster exécutant Amazon ECS. Ce cluster doit être déployé avec le type de lancement EC2. Pour plus d'informations sur l'utilisation de AWS Distro pour la télémétrie ouverte avec Amazon ECS et sur la configuration d'un cluster Amazon ECS à

cette fin, consultez Configuration de AWS Distro pour Collector OpenTelemetry dans Amazon Elastic Container Service pour les métriques au niveau des instances ECS EC2.

Rubriques

- Configuration rapide à l'aide de AWS CloudFormation
- Configuration manuelle et personnalisée

Configuration rapide à l'aide de AWS CloudFormation

Téléchargez le fichier AWS CloudFormation modèle pour installer AWS Distro for OpenTelemetry Collector pour Amazon ECS sur EC2. Exécutez la commande curl suivante.

```
curl -O https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/ecs/aws-otel-ec2-instance-metrics-daemon-deployment-cfn.yaml
```

Après avoir téléchargé le fichier modèle, ouvrez-le et remplacez *PATH_TO_ CloudFormation _TEMPLATE* par le chemin dans lequel vous avez enregistré le fichier modèle. Exportez ensuite les paramètres suivants et exécutez la AWS CloudFormation commande, comme indiqué dans la commande suivante.

- Cluster_Name– Nom du cluster Amazon ECS
- AWS_Region– Région où les données seront envoyées
- PATH_TO_ CloudFormation _TEMPLATE — Le chemin où vous avez enregistré le fichier modèle. AWS CloudFormation
- commande — Pour permettre au OpenTelemetry collecteur AWS Distro for de collecter les métriques au niveau de l'instance pour Amazon ECS sur Amazon EC2, vous devez spécifier ce paramètre. `--config=/etc/ecs/otel-instance-metrics-config.yaml`

```
ClusterName=Cluster_Name
Region=AWS_Region
command=--config=/etc/ecs/otel-instance-metrics-config.yaml
aws cloudformation create-stack --stack-name AOCECS-${ClusterName}-${Region} \
--template-body file://PATH_TO_CloudFormation_TEMPLATE \
--parameters ParameterKey=ClusterName,ParameterValue=${ClusterName} \
ParameterKey=CreateIAMRoles,ParameterValue=True \
ParameterKey=command,ParameterValue=${command} \
--capabilities CAPABILITY_NAMED_IAM \
```

```
--region ${Region}
```

Après avoir exécuté cette commande, utilisez la console Amazon ECS pour voir si la tâche est en cours d'exécution.

Dépannage de la configuration rapide

Pour vérifier l'état de la AWS CloudFormation pile, entrez la commande suivante.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack --stack-name AOCECS-$ClusterName-$Region --region
 $Region
```

Si la valeur de `StackStatus` ne correspond pas à `CREATE_COMPLETE` ou `CREATE_IN_PROGRESS`, vérifiez les événements de la pile pour trouver l'erreur. Entrez la commande suivante.

```
ClusterName=cluster-name
Region=cluster-region
aws cloudformation describe-stack-events --stack-name AOCECS-$ClusterName-$Region --
region $Region
```

Pour vérifier l'état du service de démon `AOCECS`, saisissez la commande suivante. Dans la sortie, vous devriez voir que `runningCount` est égal au `desiredCount` dans la section « deployment » (déploiement). Si ce n'est pas égal, vérifiez la section « failures » (échecs) dans la sortie.

```
ClusterName=cluster-name
Region=cluster-region
aws ecs describe-services --services AOCECS-daemon-service --cluster $ClusterName --
region $Region
```

Vous pouvez également utiliser la console CloudWatch Logs pour consulter le journal de l'agent. Recherchez le groupe de journaux /aws/ecs/containerinsights/ {} /performance. ClusterName

Configuration manuelle et personnalisée

Suivez les étapes décrites dans cette section pour déployer manuellement le AWS Distro afin de collecter des métriques OpenTelemetry au niveau de l'instance à partir de vos clusters Amazon ECS hébergés sur des instances Amazon EC2.

Étape 1 : Rôles et politiques nécessaires

Deux rôles IAM sont requis. Vous devez les créer s'ils n'existent pas déjà. Pour plus d'informations sur les rôles, consultez Créer une politique IAM et Créer un rôle IAM.

Étape 2 : Créer la définition de tâche

Créez une définition de tâche et utilisez-la pour lancer la AWS distribution en OpenTelemetry tant que service daemon.

Pour utiliser le modèle de définition de tâche afin de créer la définition de tâche, suivez les instructions de la section Créer une définition de tâche ECS EC2 pour une instance EC2 avec AWS OTel Collector.

Pour utiliser la console Amazon ECS afin de créer la définition de tâche, suivez les instructions de la section Installer le collecteur AWS OTel en créant une définition de tâche via AWS la console pour les métriques d'instance Amazon ECS EC2.

Étape 3 : Lancer le service de démon

Pour lancer la AWS distribution en OpenTelemetry tant que service daemon, suivez les instructions de la section Exécutez votre tâche sur l'Amazon Elastic Container Service (Amazon ECS) à l'aide du service daemon.

(En option) Configuration avancée

Vous pouvez éventuellement utiliser SSM pour spécifier d'autres options de configuration pour la AWS distribution OpenTelemetry dans vos clusters Amazon ECS hébergés sur des instances Amazon EC2. Pour plus d'informations sur la création d'un fichier de configuration, voir OpenTelemetry Configuration personnalisée. Pour plus d'informations sur les options que vous pouvez utiliser dans le fichier de configuration, consultez Récepteur Container Insights AWS.

Configurer FireLens pour envoyer des journaux à CloudWatch Logs

FireLens pour Amazon ECS vous permet d'utiliser les paramètres de définition des tâches pour acheminer les journaux vers Amazon Logs à CloudWatch des fins de stockage et d'analyse des journaux. FireLens fonctionne avec Fluent Bit et Fluentd. Nous fournissons une image AWS pour Fluent Bit, ou vous pouvez utiliser votre propre image Fluent Bit ou Fluentd. La création de définitions de tâches Amazon ECS avec une FireLens configuration est prise en charge à l'aide AWS des kits SDK AWS CLI, et AWS Management Console. Pour plus d'informations sur CloudWatch les journaux, voir Qu'est-ce que CloudWatch les journaux ? .

Certaines considérations clés doivent être prises en compte lors de FireLens l'utilisation d'Amazon ECS. Pour plus d'informations, consultez Éléments.

Pour trouver les images AWS pour Fluent Bit, voir Utilisation de l'image AWS pour Fluent Bit.

Pour créer une définition de tâche utilisant une FireLens configuration, voir Création d'une définition de tâche utilisant une FireLens configuration.

Exemple

L'exemple de définition de tâche suivant montre comment spécifier une configuration de journal qui transfère les journaux à un groupe de CloudWatch journaux de journaux. Pour plus d'informations, consultez Qu'est-ce qu'Amazon CloudWatch Logs ? dans le guide de l'utilisateur d'Amazon CloudWatch Logs.

Dans les options de configuration du journal, spécifiez le nom du groupe de journaux et la région dans laquelle il existe. Pour que Fluent Bit crée le groupe de journaux en votre nom, spécifiez `"auto_create_group":"true"`. Vous pouvez également spécifier l'ID de tâche comme préfixe de flux de journaux, qui facilite le filtrage. Pour plus d'informations, voir Plug-in Fluent Bit pour CloudWatch les journaux.

```
{
 "family": "firelens-example-cloudwatch",
 "taskRoleArn": "arn:aws:iam::123456789012:role/ecs_task_iam_role",
 "containerDefinitions": [
  {
   "essential": true,
   "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-bit:latest",
   "name": "log_router",
   "firelensConfiguration": {
    "type": "fluentbit"
   },
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-group": "firelens-container",
     "awslogs-region": "us-west-2",
     "awslogs-create-group": "true",
     "awslogs-stream-prefix": "firelens"
    }
   },
   "memoryReservation": 50
```

```
   },
   {
    "essential": true,
    "image": "nginx",
    "name": "app",
    "logConfiguration": {
     "logDriver":"awsfirelens",
     "options": {
     "Name": "cloudwatch",
     "region": "us-west-2",
     "log_key": "log",
                                  "log_group_name": "/aws/ecs/containerinsights/
$(ecs_cluster)/application",
     "auto_create_group": "true",
     "log_stream_name": "$(ecs_task_id)"
    }
   },
   "memoryReservation": 100
  }
 ]
}
```

Configuration de Container Insights sur Amazon EKS et Kubernetes

Container Insights est pris en charge sur les versions 1.23 et ultérieures d'Amazon EKS. La méthode d'installation rapide n'est prise en charge que sur les versions 1.24 et ultérieures.

Le processus général de configuration de Container Insights sur Amazon EKS ou Kubernetes est le suivant :

1. Vérifiez que vous disposez des prérequis nécessaires.
2. Configurez le module complémentaire Amazon CloudWatch Observability EKS, l' CloudWatch agent ou AWS Distro pour OpenTelemetry votre cluster auquel envoyer des métriques. CloudWatch

> **Note**
>
> Pour utiliser Container Insights avec une observabilité améliorée pour Amazon EKS, vous devez utiliser le module complémentaire Amazon CloudWatch Observability EKS ou l' CloudWatch agent. Pour plus d'informations sur cette version de Container Insights, veuillez consulter Container Insights avec observabilité améliorée pour Amazon EKS.

Pour utiliser Container Insights avec Fargate, vous devez AWS utiliser Distro pour. OpenTelemetry Container Insights avec observabilité améliorée pour Amazon EKS n'est pas pris en charge sur Fargate.

> **Note**
>
> Container Insights prend désormais en charge les nœuds de travail Windows dans un cluster Amazon EKS. Container Insights avec une observabilité améliorée pour Amazon EKS est également pris en charge sous Windows. Pour plus d'informations sur l'activation de Container Insights sous Windows, consultezUtilisation de l' CloudWatch agent avec l'observabilité améliorée de Container Insights activée.

Configurez Fluent Bit ou Fluentd pour envoyer les journaux à CloudWatch Logs. (Ceci est activé par défaut si vous installez le module complémentaire Amazon CloudWatch Observability EKS.)

Vous pouvez effectuer ces étapes en une seule fois dans le cadre de la configuration de démarrage rapide si vous utilisez l' CloudWatch agent, ou les effectuer séparément.

3. (En option) Configurez la journalisation de plan de contrôle Amazon EKS.
4. (Facultatif) Configurez l' CloudWatch agent en tant que point de terminaison StatsD sur le cluster auquel envoyer les métriques StatsD. CloudWatch
5. (En option) Activez les journaux d'accès App Mesh Envoy.

Dans la version originale de Container Insights, les métriques collectées et les logs ingérés sont facturés en tant que métriques personnalisées. Grâce à Container Insights avec observabilité améliorée pour Amazon EKS, les métriques et les journaux de Container Insights sont facturés par observation au lieu d'être facturés par métrique stockée ou par journal ingéré. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Rubriques

- Vérifiez les conditions préalables
- Utilisation de l' CloudWatch agent avec l'observabilité améliorée de Container Insights activée
- Utiliser AWS Distro pour OpenTelemetry
- Envoyer des journaux à CloudWatch Logs

- Mise à jour ou suppression de Container Insights sur Amazon EKS et Kubernetes

Vérifiez les conditions préalables

Avant d'installer Container Insights sur Amazon EKS ou Kubernetes, vérifiez les points suivants. Ces prérequis s'appliquent que vous utilisiez l' CloudWatch agent ou AWS Distro pour OpenTelemetry configurer Container Insights sur des clusters Amazon EKS.

- Vous disposez d'un cluster Amazon EKS ou Kubernetes fonctionnel avec des nœuds attachés dans l'une des régions prenant en charge Container Insights pour Amazon EKS et Kubernetes. Pour obtenir la liste des régions prises en charge, consultez Container Insights.
- `kubectl` est installé et en cours d'exécution. Pour plus d'informations, consultez Installation de `kubectl` dans le Guide de l'utilisateur Amazon EKS.
- Si vous utilisez Kubernetes en cours d'exécution au AWS lieu d'Amazon EKS, les conditions préalables suivantes sont également nécessaires :
  - Assurez-vous que votre cluster Kubernetes a activé le contrôle d'accès basé sur les rôles (RBAC). Pour plus d'informations, consultez Using RBAC Authorization (Utilisation des autorisations de contrôle d'accès basé sur les rôle) dans la documentation Reference de Kubernetes.
  - Votre Kubelet a activé le mode d'autorisation Webhook. Pour plus d'informations, consultez Kubelet authentication/authorization (Authentification/autorisation Kubelet) dans la documentation Reference de Kubernetes.

Vous devez également accorder des autorisations IAM pour permettre à vos nœuds de travail Amazon EKS d'envoyer des métriques et des journaux à CloudWatch. Il existe deux façons de procéder :

- Attachez une politique au rôle IAM de vos composants master. Cela fonctionne pour les clusters Amazon EKS et autres clusters Kubernetes.
- Utilisez un rôle IAM pour les comptes de service pour le cluster et attachez la politique à ce rôle. Cela ne fonctionne que pour les clusters Amazon EKS.

La première option accorde des autorisations CloudWatch pour l'ensemble du nœud, tandis que l'utilisation d'un rôle IAM pour le compte de service donne CloudWatch accès uniquement aux pods daemonset appropriés.

Attacher une politique au rôle IAM de vos composants master

Procédez comme suit pour attacher la politique au rôle IAM de vos composants master. Cela fonctionne à la fois pour les clusters Amazon EKS et les clusters Kubernetes en dehors d'Amazon EKS.

Pour ajouter la politique nécessaire au rôle IAM pour vos composants master

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Sélectionnez l'une des instances de composant master, puis choisissez le rôle IAM dans la description.
3. Sur la page du rôle IAM, choisissez Attach policies (Attacher des politiques).
4. Dans la liste des politiques, cochez la case située à côté de CloudWatchAgentServerPolicy. Si nécessaire, utilisez la zone de recherche pour trouver cette politique.
5. Choisissez Attach Policies (Attacher des politiques).

Si vous exécutez un cluster Kubernetes à l'extérieur d'Amazon EKS, il est possible que vous n'ayez pas encore de rôle IAM attaché à vos composants master. Si tel est le cas, vous devez d'abord attacher le rôle IAM à l'instance, puis ajouter la politique comme expliqué dans les étapes précédentes. Pour plus d'informations sur l'attachement d'un rôle à une instance, consultez Attachement d'un rôle IAM à une instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

Si vous exécutez un cluster Kubernetes en dehors d'Amazon EKS et que vous souhaitez collecter les ID de volume EBS dans les métriques, vous devez ajouter une autre politique au rôle IAM attaché à l'instance. Ajoutez les éléments suivants en tant que politique en ligne. Pour plus d'informations, consultez Ajout et suppression d'autorisations basées sur l'identité IAM dans le Guide de l'utilisateur IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
```

```
        }
    ]
}
```

Utilisation d'un rôle de compte de service IAM

Cette méthode fonctionne uniquement sur les clusters Amazon EKS.

Pour autoriser l' CloudWatch utilisation d'un rôle de compte de service IAM

1. Si vous ne l'avez pas déjà fait, activez les rôles IAM pour les comptes de service sur votre cluster. Pour plus d'informations, consultez Activation des rôles IAM pour les comptes de service sur votre cluster.
2. Si vous ne l'avez pas encore fait, configurez le compte de service pour utiliser un rôle IAM. Pour plus d'informations, consultez Configuration d'un compte de service Kubernetes pour assumer un rôle IAM.

   Lorsque vous créez le rôle, associez la stratégie CloudWatchAgentServerPolicyIAM au rôle en plus de la stratégie que vous créez pour le rôle. En outre, le compte de service Kubernetes associé associé à ce rôle doit être créé dans l'espace de `amazon-cloudwatch` noms, où les daemonsets CloudWatch et Fluent Bit seront déployés dans les prochaines étapes
3. Si vous ne l'avez pas déjà fait, associez le rôle IAM à un compte de service de votre cluster. Pour plus d'informations, consultez Configuration d'un compte de service Kubernetes pour assumer un rôle IAM.

Utilisation de l' CloudWatch agent avec l'observabilité améliorée de Container Insights activée

Suivez les instructions de l'une des sections suivantes pour configurer Container Insights sur un cluster Amazon EKS ou Kubernetes à l'aide de l'agent. CloudWatch Les instructions de démarrage rapide ne sont prises en charge que sur les versions 1.24 et ultérieures d'Amazon EKS.

> **Note**
>
> Vous pouvez installer Container Insights en suivant les instructions de l'une des sections suivantes. Vous n'avez pas besoin de suivre les trois séries d'instructions.

Rubriques

- Installez le module complémentaire Amazon CloudWatch Observability EKS
- Configuration Quick Start pour Container Insights sur Amazon EKS et Kubernetes
- Configurer l' CloudWatch agent pour collecter les métriques du cluster

Installez le module complémentaire Amazon CloudWatch Observability EKS

Vous pouvez utiliser l'add-on Amazon EKS afin d'installer Container Insights avec observabilité améliorée pour Amazon EKS. Le module complémentaire installe l' CloudWatch agent pour envoyer les métriques d'infrastructure depuis le cluster, installe Fluent Bit pour envoyer les journaux des conteneurs et permet également d' CloudWatch Application Signalsenvoyer la télémétrie des performances des applications.

Lorsque vous utilisez le module complémentaire Amazon EKS version 1.5.0 ou ultérieure, Container Insights est activé sur les nœuds de travail Linux et Windows du cluster. Actuellement, Application Signals n'est pas pris en charge sous Windows dans Amazon EKS.

L'add-on Amazon EKS n'est pas pris en charge pour les clusters exécutant Kubernetes au lieu d'Amazon EKS.

Pour plus d'informations sur le module complémentaire Amazon CloudWatch Observability EKS, consultezInstallez l' CloudWatch agent à l'aide du module complémentaire Amazon CloudWatch Observability EKS.

Pour installer le module complémentaire Amazon CloudWatch Observability EKS

1. Configurez d'abord les autorisations nécessaires en associant la politique CloudWatchAgentServerPolicyIAM à vos nœuds de travail. Pour ce faire, entrez la commande suivante. *my-worker-node-role*Remplacez-le par le rôle IAM utilisé par vos nœuds de travail Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

2. Saisissez la commande suivante pour installer l'add-on :

```
aws eks create-addon --cluster-name my-cluster-name --addon-name amazon-cloudwatch-
observability
```

Configuration Quick Start pour Container Insights sur Amazon EKS et Kubernetes

> **Important**
>
> Si vous installez Container Insights sur un cluster Amazon EKS, nous vous recommandons d'utiliser le module complémentaire Amazon CloudWatch Observability EKS pour l'installation, au lieu de suivre les instructions de cette section. En outre, pour récupérer des réseaux informatiques accélérés, vous devez utiliser le module complémentaire Amazon CloudWatch Observability EKS. Pour en savoir plus et des instructions, consultez Installez le module complémentaire Amazon CloudWatch Observability EKS.

Pour terminer la configuration de Container Insights, vous pouvez suivre les instructions de démarrage rapide de cette section. Si vous effectuez une installation dans un cluster Amazon EKS et que vous suivez les instructions de cette section le 6 novembre 2023 ou après cette date, vous installez Container Insights avec observabilité améliorée pour Amazon EKS dans le cluster.

> **Important**
>
> Avant d'exécuter la procédure indiquée dans cette section, vous devez satisfaire aux prérequis, y compris les autorisations IAM. Pour plus d'informations, consultez . Vérifiez les conditions préalables.

Sinon, vous pouvez suivre les instructions des deux sections suivantes, Configurer l' CloudWatch agent pour collecter les métriques du cluster et Envoyer des journaux à CloudWatch Logs. Ces sections fournissent des informations de configuration supplémentaires sur le fonctionnement de l' CloudWatch agent avec Amazon EKS et Kubernetes, mais vous demandent d'effectuer d'autres étapes d'installation.

Dans la version originale de Container Insights, les métriques collectées et les logs ingérés sont facturés en tant que métriques personnalisées. Grâce à Container Insights avec observabilité améliorée pour Amazon EKS, les métriques et les journaux de Container Insights sont facturés par observation au lieu d'être facturés par métrique stockée ou par journal ingéré. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

> **Note**
>
> Amazon a maintenant lancé Fluent Bit comme solution de journal par défaut pour Container Insights avec des gains de performances importants. Nous vous recommandons d'utiliser Fluent Bit au lieu de Fluentd.

Démarrage rapide avec l'opérateur de l' CloudWatch agent et Fluent Bit

Il existe deux configurations pour Fluent Bit : une version optimisée et une version qui offre une expérience plus similaire à Fluentd. La configuration Quick Start utilise la version optimisée. Pour plus d'informations sur la configuration compatible Fluentd, consultez Configurer Fluent Bit comme un DaemonSet pour envoyer des CloudWatch journaux à Logs.

L'opérateur de l' CloudWatch agent est un conteneur supplémentaire qui est installé sur un cluster Amazon EKS. Il est calqué sur l' OpenTelemetry opérateur pour Kubernetes. L'opérateur gère le cycle de vie des ressources Kubernetes dans un cluster. Il installe l' CloudWatch agent, le DCGM Exporter (NVIDIA) et le AWS Neuron Monitor sur un cluster Amazon EKS et les gère. Fluent Bit et l' CloudWatch agent pour Windows sont installés directement sur un cluster Amazon EKS sans que l'opérateur ne les gère.

Pour une solution d'autorité de certification plus sécurisée et riche en fonctionnalités, l'opérateur de l' CloudWatch agent a besoin de cert-manager, une solution largement adoptée pour la gestion des certificats TLS dans Kubernetes. L'utilisation de cert-manager simplifie le processus d'obtention, de renouvellement, de gestion et d'utilisation de ces certificats. Il garantit la validité et la mise à jour des certificats et tente de les renouveler à une heure définie avant leur expiration. cert-manager facilite également l'émission de certificats provenant de diverses sources prises en charge, notamment AWS Certificate Manager Private Certificate Authority.

Pour déployer Container Insights à l'aide du démarrage rapide

1. Installez cert-manager s'il n'est pas déjà installé dans le cluster. Pour plus d'informations, consultez la section Installation de cert-manager.
2. Installez les définitions de ressources personnalisées (CRD) en saisissant la commande suivante.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl
 apply --server-side -f -
```

3. Installez l'opérateur en saisissant la commande suivante. *my-cluster-name*Remplacez-le par le nom de votre cluster Amazon EKS ou Kubernetes, puis par le nom *my-cluster-region*de la région dans laquelle les journaux sont publiés. Nous vous recommandons d'utiliser la même région que celle dans laquelle votre cluster est déployé afin de réduire les coûts de transfert de données AWS sortants.

   ```
   ClusterName=my-cluster-name
   RegionName=my-cluster-region
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
   {{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
    apply -f -
   ```

   Par exemple, pour déployer Container Insights sur le cluster nommé `MyCluster` et publier les journaux et les métriques dans USA Ouest (Oregon), entrez la commande suivante.

   ```
   ClusterName='MyCluster'
   RegionName='us-west-2'
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
   {{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
    apply -f -
   ```

Migration depuis Container Insights

Si Container Insights est déjà configuré dans un cluster Amazon EKS et que vous souhaitez migrer vers Container Insights avec une meilleure observabilité pour Amazon EKS, voir Mise à niveau vers Container Insights avec observabilité améliorée pour Amazon EKS

Suppression de Container Insights

Si vous souhaitez supprimer Container Insights après avoir utilisé la configuration de démarrage rapide, entrez les commandes suivantes.

```
ClusterName=my-cluster-name
```

```
RegionName=my-cluster-region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Configurer l' CloudWatch agent pour collecter les métriques du cluster

> **Important**
>
> Si vous installez Container Insights sur un cluster Amazon EKS, nous vous recommandons d'utiliser le module complémentaire Amazon CloudWatch Observability EKS pour l'installation, au lieu de suivre les instructions de cette section. Pour en savoir plus et des instructions, consultez Installez le module complémentaire Amazon CloudWatch Observability EKS.

Pour configurer Container Insights afin de collecter des métriques, vous pouvez suivre les étapes de Configuration Quick Start pour Container Insights sur Amazon EKS et Kubernetes ou les étapes de cette section. Dans les étapes suivantes, vous configurez l' CloudWatch agent pour qu'il puisse collecter des métriques à partir de vos clusters.

Si vous effectuez une installation dans un cluster Amazon EKS et que vous suivez les instructions de cette section le 6 novembre 2023 ou après cette date, vous installez Container Insights avec observabilité améliorée pour Amazon EKS dans le cluster.

Étape 1 : créer un espace de noms pour CloudWatch

Procédez comme suit pour créer l'espace de noms Kubernetes demandé. `amazon-cloudwatch` CloudWatch Vous pouvez ignorer cette étape si vous avez déjà créé cet espace de noms.

Pour créer un espace de noms pour CloudWatch

- Entrez la commande suivante.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Étape 2 : Créer un compte de service dans le cluster

Procédez comme suit pour créer un compte de service pour l' CloudWatch agent, si vous n'en avez pas déjà un.

Pour créer un compte de service pour l' CloudWatch agent

- Entrez la commande suivante.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/cwagent/cwagent-serviceaccount.yaml
```

Si vous n'avez pas suivi les étapes précédentes, mais que vous possédez déjà un compte de service pour l' CloudWatch agent que vous souhaitez utiliser, vous devez vous assurer qu'il respecte les règles suivantes. En outre, lors des autres étapes d'installation de Container Insights, vous devez utiliser le nom de ce compte de service au lieu de `cloudwatch-agent`.

```
rules:
  - apiGroups: [""]
    resources: ["pods", "nodes", "endpoints"]
    verbs: ["list", "watch"]
  - apiGroups: [ "" ]
    resources: [ "services" ]
    verbs: [ "list", "watch" ]
  - apiGroups: ["apps"]
    resources: ["replicasets", "daemonsets", "deployments", "statefulsets"]
    verbs: ["list", "watch"]
  - apiGroups: ["batch"]
    resources: ["jobs"]
    verbs: ["list", "watch"]
  - apiGroups: [""]
    resources: ["nodes/proxy"]
    verbs: ["get"]
  - apiGroups: [""]
```

```
    resources: ["nodes/stats", "configmaps", "events"]
    verbs: ["create", "get"]
  - apiGroups: [""]
    resources: ["configmaps"]
    resourceNames: ["cwagent-clusterleader"]
    verbs: ["get","update"]
  - nonResourceURLs: ["/metrics"]
    verbs: ["get", "list", "watch"]
```

Étape 3 : créer un ConfigMap pour l' CloudWatchagent

Suivez les étapes ci-dessous pour créer un ConfigMap pour l' CloudWatch agent.

Pour créer un ConfigMap pour l' CloudWatch agent

1. Téléchargez le ConfigMap fichier YAML sur votre hôte `kubectl` client en exécutant la commande suivante :

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-configmap.yaml
   ```

2. Modifiez le fichier YAML téléchargé comme suit :

   - cluster_name – Dans la section `kubernetes`, remplacez `{{cluster_name}}` par le nom de votre cluster. Supprimez les caractères `{{}}`. Sinon, si vous utilisez un cluster Amazon EKS, vous pouvez supprimer le champ `"cluster_name"` et sa valeur. Dans ce cas, l' CloudWatch agent détecte le nom du cluster dans les balises Amazon EC2.

3. (Facultatif) Apportez d'autres modifications au ConfigMap en fonction de vos besoins de surveillance, comme suit :

   - metrics_collection_interval – Dans la section `kubernetes`, vous pouvez spécifier à quelle fréquence l'agent collecte les métriques. Le durée par défaut est 60 secondes. Dans kubelet, l'intervalle de collecte par défaut (via cadvisor) est de 15 secondes, vous ne devez donc pas définir cette valeur sur moins de 15 secondes.
   - endpoint_override — Dans la `logs` section, vous pouvez spécifier le point de terminaison CloudWatch Logs si vous souhaitez remplacer le point de terminaison par défaut. Vous pouvez effectuer cette opération si vous publiez à partir d'un cluster dans un VPC et souhaitez que les données soient transmises à un point de terminaison d'un VPC.

- force_flush_interval — Dans `logs` cette section, vous pouvez spécifier l'intervalle de traitement par lots des événements du journal avant leur publication dans Logs. CloudWatch Le durée par défaut est 5 secondes.
- region – Par défaut, l'agent a publié les métriques dans la région où se trouve le composant master. Pour contourner ce problème; vous pouvez ajouter un champ `region` dans la section `agent` : par exemple, `"region":"us-west-2"`.
- section statsd — Si vous souhaitez que l'agent CloudWatch Logs s'exécute également en tant qu'écouteur StatsD dans chaque nœud de travail de votre cluster, vous pouvez ajouter `statsd` une section à `metrics` la section, comme dans l'exemple suivant. Pour plus d'informations sur les autres options StatsD pour cette section, consultez Récupération de métriques personnalisées avec StatsD .

```
"metrics": {
  "metrics_collected": {
    "statsd": {
      "service_address":":8125"
    }
  }
}
```

Voici un exemple complet de la section de code JSON.

```
{
    "agent": {
        "region": "us-east-1"
    },
    "logs": {
        "metrics_collected": {
            "kubernetes": {
                "cluster_name": "MyCluster",
                "metrics_collection_interval": 60
            }
        },
        "force_flush_interval": 5,
        "endpoint_override": "logs.us-east-1.amazonaws.com"
    },
    "metrics": {
        "metrics_collected": {
            "statsd": {
                "service_address": ":8125"
```

```
            }
        }
    }
}
```

4. Créez le ConfigMap dans le cluster en exécutant la commande suivante.

```
kubectl apply -f cwagent-configmap.yaml
```

Étape 4 : Déployer l' CloudWatch agent en tant que DaemonSet

Pour terminer l'installation de l' CloudWatch agent et commencer à collecter les métriques du conteneur, procédez comme suit.

Pour déployer l' CloudWatch agent en tant que DaemonSet

1. • Si vous ne souhaitez pas utiliser StatsD sur le cluster, entrez la commande suivante.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
deployment-mode/daemonset/container-insights-monitoring/cwagent/cwagent-
daemonset.yaml
```

   • Si vous souhaitez utiliser StatsD, procédez comme suit :

      a. Téléchargez le DaemonSet fichier YAML sur votre hôte `kubectl` client en exécutant la commande suivante.

```
curl -O  https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-
mode/daemonset/container-insights-monitoring/cwagent/cwagent-daemonset.yaml
```

      b. Supprimez la mise en commentaire de la section `port` dans le fichier `cwagent-daemonset.yaml` comme dans l'exemple suivant :

```
ports:
  - containerPort: 8125
    hostPort: 8125
    protocol: UDP
```

      c. Déployez l' CloudWatch agent dans votre cluster en exécutant la commande suivante.

```
kubectl apply -f cwagent-daemonset.yaml
```

d. Déployez l' CloudWatch agent sur les nœuds Windows de votre cluster en exécutant la commande suivante. L'écouteur StatsD n'est pas pris en charge par l' CloudWatch agent sous Windows.

```
kubectl apply -f cwagent-daemonset-windows.yaml
```

2. Assurez-vous que l'agent est déployé en exécutant la commande suivante.

```
kubectl get pods -n amazon-cloudwatch
```

Une fois terminé, l' CloudWatch agent crée un groupe de journaux nommé `/aws/containerinsights/`*`Cluster_Name`*`/performance` et envoie les événements du journal des performances à ce groupe de journaux. Si vous avez également configuré l'agent en tant que port d'écoute StatsD, l'agent écoute également les métriques StatsD sur le port 8125 avec l'adresse IP du nœud dans lequel le pod d'application est programmé.

Résolution des problèmes

Si l'agent ne se déploie pas correctement, essayez ce qui suit :

- Exécutez la commande suivante pour obtenir la liste des pods.

```
kubectl get pods -n amazon-cloudwatch
```

- Exécutez la commande suivante et vérifiez les événements au bas de la sortie.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Exécutez la commande suivante pour vérifier les journaux.

```
kubectl logs pod-name  -n amazon-cloudwatch
```

Utiliser AWS Distro pour OpenTelemetry

Vous pouvez configurer Container Insights pour collecter des métriques à partir de clusters Amazon EKS à l'aide de AWS Distro for OpenTelemetry Collector. Pour plus d'informations sur la AWS distribution pour OpenTelemetry, consultez la section AWS Distribution pour. OpenTelemetry

> **Important**
>
> Si vous effectuez l'installation à l'aide de AWS Distro pour OpenTelemetry, vous installez Container Insights mais vous ne bénéficiez pas de Container Insights avec une observabilité améliorée pour Amazon EKS. Vous ne collecterez pas les métriques détaillées prises en charge dans Container Insights avec observabilité améliorée pour Amazon EKS.

La façon dont vous configurez Container Insights dépend de si le cluster est hébergé sur des instances Amazon EC2 ou sur AWS Fargate (Fargate).

Clusters Amazon EKS hébergés sur Amazon EC2

Si vous ne l'avez pas déjà fait, veillez à vous assurer que vous avez satisfait aux prérequis, y compris les rôles IAM nécessaires. Pour plus d'informations, consultez Vérifiez les conditions préalables.

Amazon fournit les Charts de Helm que vous pouvez utiliser pour configurer la surveillance d'Amazon Elastic Kubernetes Service sur Amazon EC2. Cette surveillance utilise le collecteur AWS Distro for OpenTelemetry (ADOT) pour les métriques et Fluent Bit pour les journaux. Le graphique Helm est donc utile pour les clients qui utilisent Amazon EKS sur Amazon EC2 et qui souhaitent collecter des métriques et des journaux à envoyer à CloudWatch Container Insights. Pour plus d'informations sur ce graphique Helm, consultez le graphique ADOT Helm pour EKS sur les métriques EC2 et les journaux sur Amazon CloudWatch Container Insights.

Sinon, vous pouvez utiliser les instructions du reste de cette section.

Déployez d'abord le AWS Distro for OpenTelemetry Collector sous forme de DaemonSet fichier en saisissant la commande suivante.

```
curl https://raw.githubusercontent.com/aws-observability/aws-otel-collector/main/
deployment-template/eks/otel-container-insights-infra.yaml |
kubectl apply -f -
```

Pour vérifier que le collecteur est en cours d'exécution, saisissez la commande suivante.

```
kubectl get pods -l name=aws-otel-eks-ci -n aws-otel-eks
```

Si la sortie de cette commande inclut plusieurs pods à l'état `Running`, le collecteur est en cours d'exécution et collecte des métriques à partir du cluster. Le collecteur crée un groupe de journaux nommé `aws/containerinsights/`*`cluster-name`*`/performance` et lui envoie les événements de journaux des performances.

Pour plus d'informations sur la façon d'afficher vos métriques Container Insights dans CloudWatch, consultezAffichage des métriques dans Container Insights.

AWS a également fourni de la documentation sur GitHub ce scénario. Si vous souhaitez personnaliser les métriques et les journaux publiés par Container Insights, consultez https://aws-otel.github.io/docs/getting-started/container-insights/eks-infra.

Clusters Amazon EKS hébergés sur Fargate

Pour savoir comment configurer et déployer un collecteur ADOT afin de collecter les métriques du système à partir des charges de travail déployées sur un cluster Amazon EKS sur Fargate et de les envoyer à Container Insights, consultez CloudWatch Container Insights EKS Fargate in the Distro pour obtenir de la documentation. AWS OpenTelemetry

Envoyer des journaux à CloudWatch Logs

Pour envoyer les journaux de vos conteneurs vers Amazon CloudWatch Logs, vous pouvez utiliser Fluent Bit ou Fluentd. Pour plus d'informations, consultez Fluent Bit et Fluentd.

Si vous n'utilisez pas déjà Fluentd, nous vous recommandons d'utiliser Fluent Bit pour les raisons suivantes :

- Fluent Bit nécessite moins de ressources et utilise plus efficacement les ressources pour l'utilisation de la mémoire et du processeur que Fluentd. Pour obtenir une comparaison plus détaillée, consultez Comparaison des performances de Fluent Bit et de Fluentd.
- L'image Fluent Bit est développée et maintenue par AWS. Cela permet d'adopter AWS les nouvelles fonctionnalités d'image Fluent Bit et de résoudre les problèmes beaucoup plus rapidement.

Rubriques

- Comparaison des performances de Fluent Bit et de Fluentd
- Configurer Fluent Bit comme un DaemonSet pour envoyer des CloudWatch journaux à Logs

- (Facultatif) Configurez Fluentd en tant que DaemonSet pour envoyer des journaux à Logs CloudWatch
- (En option) Configurez la journalisation de plan de contrôle Amazon EKS
- (En option) Activez les journaux d'accès App Mesh Envoy.
- (En option) Activez la fonction Use_Kubelet pour les grands clusters

Comparaison des performances de Fluent Bit et de Fluentd

Les tableaux suivants montrent l'avantage en matière de performances que Fluent Bit a par rapport à Fluentd en termes d'utilisation de la mémoire et du processeur. Les numéros suivants sont indiqués à titre de référence et peuvent changer en fonction de l'environnement.

Journaux par seconde	Utilisation du processeur Fluentd	Utilisation du processeur Fluent Bit avec configuration compatible Fluentd	Utilisation du processeur Fluent Bit avec configuration optimisée
100	0,35 vCPU	0,02 vCPU	0,02 vCPU
1 000	0,32 vCPU	0,14 vCPU	0,11 vCPU
5 000	0,85 vCPU	0,48 vCPU	0,30 vCPU
10 000	0,94 vCPU	0,60 vCPU	0,39 vCPU

Journaux par seconde	Utilisation de la mémoire Fluentd	Utilisation de la mémoire Fluent Bit avec configuration compatible Fluentd	Utilisation de la mémoire Fluent Bit avec configuration optimisée
100	153 Mo	46 Mo	37 Mo
1 000	270 Mo	45 Mo	40 Mo
5 000	320 Mo	55 Mo	45 Mo
10 000	375 Mo	92 Mo	75 Mo

Configurer Fluent Bit comme un DaemonSet pour envoyer des CloudWatch journaux à Logs

Les sections suivantes vous aident à déployer Fluent Bit pour envoyer des journaux depuis des conteneurs vers CloudWatch des journaux.

Rubriques

Différences si vous utilisez déjà Fluentd

Si vous utilisez déjà Fluentd pour envoyer des logs depuis des conteneurs vers Logs, lisez cette section pour voir les différences entre Fluentd et Fluentd CloudWatch Bit. Si vous n'utilisez pas déjà Fluentd avec Container Insights, vous pouvez passer à Configuration de Fluent Bit.

Nous proposons deux configurations par défaut pour Fluent Bit :

- Configuration optimisée pour Fluent Bit – Configuration alignée sur les bonnes pratiques de Fluent Bit.
- Configuration compatible Fluentd – Configuration alignée autant que possible sur le comportement Fluentd.

La liste suivante explique en détail les différences entre Fluentd et chaque configuration Fluent Bit.

- Différences dans les noms de flux de journaux – Si vous utilisez la configuration optimisée Fluent Bit, les noms de flux de journaux seront différents.

  Sous `/aws/containerinsights/Cluster_Name/application`

  - La configuration optimisée Fluent Bit envoie les journaux à *`kubernetes-nodeName`*`-application.var.log.containers.`*`kubernetes-podName_kubernetes-namespace_kubernetes-container-name-kubernetes-containerID`*
  - Fluentd envoie les journaux à *`kubernetes-podName_kubernetes-namespace_kubernetes-containerName_kubernetes-containerID`*

Sous `/aws/containerinsights/Cluster_Name/host`

- La configuration optimisée Fluent Bit envoie les journaux à *`kubernetes-nodeName`*`.`*`host-log-file`*
- Fluentd envoie les journaux à *`host-log-file`*`-`*`Kubernetes-NodePrivateIp`*

Sous `/aws/containerinsights/Cluster_Name/dataplane`

- La configuration optimisée Fluent Bit envoie les journaux à *`kubernetes-nodeName`*`.`*`dataplaneServiceLog`*
- Fluentd envoie les journaux à *`dataplaneServiceLog`*`-`*`Kubernetes-nodeName`*

- Les fichiers journaux `kube-proxy` et `aws-node` écrits par Container Insights se trouvent dans des emplacements différents. Dans la configuration Fluentd, ils sont dans `/aws/containerinsights/Cluster_Name/application`. Dans la configuration optimisée Fluent Bit, ils sont dans `/aws/containerinsights/Cluster_Name/dataplane`.
- La plupart des métadonnées, comme `pod_name` et `namespace_name`, sont les mêmes dans Fluent Bit et Fluentd, mais ce qui suit est différent.
  - La configuration optimisée Fluent Bit utilise `docker_id` et Fluentd utilise `Docker.container_id`.
  - Les deux configurations Fluent Bit n'utilisent pas les métadonnées suivantes. Elles ne sont présentes que dans Fluentd : `container_image_id`, `master_url`, `namespace_id`, et `namespace_labels`.

Configuration de Fluent Bit

Pour configurer Fluent Bit afin de collecter les journaux de vos conteneurs, vous pouvez suivre les étapes de Configuration Quick Start pour Container Insights sur Amazon EKS et Kubernetes ou celles de cette section.

Avec l'une ou l'autre méthode, le rôle IAM qui est attaché au nœuds du cluster doit disposer d'autorisations suffisantes. Pour plus d'informations sur les autorisations requises pour exécuter un cluster Amazon EKS, consultez Politiques, rôles et autorisations IAM Amazon EKS dans le Guide de l'utilisateur Amazon EKS.

Dans les étapes suivantes, vous allez configurer Fluent Bit en tant que DaemonSet pour envoyer des journaux à Logs. CloudWatch Une fois que vous avez terminé cette étape, Fluent Bit crée les groupes de journaux suivants, s'ils n'existent pas déjà.

> **Important**
>
> Si FluentD est déjà configuré dans Container Insights et que le FluentD ne fonctionne pas comme prévu (cela peut se produire si vous utilisez le moteur d'exécution), vous devez `containerd` le désinstaller avant d'installer DaemonSet FluentD pour empêcher FluentD de traiter les messages du journal d'erreurs FluentD. Sinon, vous devez désinstaller Fluentd immédiatement après avoir installé Fluent Bit. La désinstallation de Fluentd après l'installation de Fluent Bit garantit la continuité de la journalisation pendant ce processus de migration. Un seul Fluent Bit ou FluentD est nécessaire pour envoyer des journaux à Logs. CloudWatch

Nom du groupe de journaux	Source des journaux
`/aws/containerinsights/` *`Cluster_Name`* `/application`	Tous les fichiers journaux situés dans `/var/log/containers`
`/aws/containerinsights/` *`Cluster_Name`* `/host`	Journaux provenant de `/var/log/dmesg` , `/var/log/secure` et `/var/log/messages`
`/aws/containerinsights/` *`Cluster_Name`* `/dataplane`	Les journaux dans `/var/log/journal` pour `kubelet.service` , `kubeproxy.service` et `docker.service` .

Pour installer Fluent Bit pour envoyer des journaux depuis des conteneurs vers des CloudWatch journaux

1. Si vous n'avez pas encore d'espace de noms appelé `amazon-cloudwatch`, créez-en un en saisissant la commande suivante :

   ```
   kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
   ```

2. Exécutez la commande suivante pour créer un ConfigMap nom `cluster-info` avec le nom du cluster et la région à laquelle envoyer les journaux. Remplacez *`cluster-name`* et *`cluster-region`* par le nom et la région de votre cluster.

```
ClusterName=cluster-name
RegionName=cluster-region
FluentBitHttpPort='2020'
FluentBitReadFromHead='Off'
[[ ${FluentBitReadFromHead} = 'On' ]] && FluentBitReadFromTail='Off'||
 FluentBitReadFromTail='On'
[[ -z ${FluentBitHttpPort} ]] && FluentBitHttpServer='Off' ||
 FluentBitHttpServer='On'
kubectl create configmap fluent-bit-cluster-info \
--from-literal=cluster.name=${ClusterName} \
--from-literal=http.server=${FluentBitHttpServer} \
--from-literal=http.port=${FluentBitHttpPort} \
--from-literal=read.head=${FluentBitReadFromHead} \
--from-literal=read.tail=${FluentBitReadFromTail} \
--from-literal=logs.region=${RegionName} -n amazon-cloudwatch
```

Dans cette commande, le `FluentBitHttpServer` permettant de surveiller les métriques de plugin est activé par défaut. Pour le désactiver, remplacez la troisième ligne de la commande par `FluentBitHttpPort=''` (chaîne vide) dans la commande.

Également par défaut, Fluent Bit lit les fichiers journaux de processus, et ne capture que les nouveaux journaux après son déploiement. Si vous souhaitez faire le contraire, définissez `FluentBitReadFromHead='On'` et il va collecter tous les journaux dans le système de fichiers.

3. Téléchargez et déployez le daemonset Fluent Bit sur le cluster en exécutant l'une des commandes suivantes.

   - Si vous souhaitez une configuration optimisée de Fluent Bit pour les ordinateurs Linux, exécutez cette commande.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-
     bit.yaml
     ```

   - Si vous souhaitez une configuration optimisée de Fluent Bit pour les ordinateurs Windows, exécutez cette commande.

     ```
     kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-
     cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/
     ```

```
deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-windows.yaml
```

- Si vous utilisez des ordinateurs Linux et que vous souhaitez une configuration Fluent Bit plus similaire à Fluentd, exécutez cette commande.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/fluent-bit/fluent-bit-compatible.yaml
```

> **Important**
>
> La configuration du daemonset Fluent Bit définit par défaut le niveau de journalisation sur INFO, ce qui peut entraîner des coûts d'ingestion de CloudWatch logs plus élevés. Si vous souhaitez réduire le volume et les coûts d'ingestion des journaux, vous pouvez modifier le niveau de journal sur ERROR.
> Pour plus d'informations sur la réduction du volume des journaux, consultez (En option) Réduction du volume des journaux de Fluent Bit

4. Validez le déploiement en entrant la commande suivante. Chaque nœud doit avoir un pod nommé fluent-bit-*.

```
kubectl get pods -n amazon-cloudwatch
```

Les étapes ci-dessus créent les ressources suivantes dans le cluster :

- Un compte de service nommé `Fluent-Bit` dans l'espace de noms `amazon-cloudwatch`. Ce compte de service est utilisé pour exécuter le daemonSet Fluent Bit. Pour plus d'informations, consultez Managing Service Accounts (Gestion des comptes de service) dans la documentation Reference de Kubernetes.
- Un rôle de cluster rôle nommé `Fluent-Bit-role` dans l'espace de noms `amazon-cloudwatch`. Ce rôle de cluster octroie des autorisations `get`, `list` et `watch` sur les journaux de pod au compte de service `Fluent-Bit`. Pour plus d'informations, consultez API Overview (Présentation de l'API) dans la documentation Reference de Kubernetes.task

- Un ConfigMap nommé `Fluent-Bit-config` dans l'espace de `amazon-cloudwatch` noms. Il ConfigMap contient la configuration à utiliser par Fluent Bit. Pour plus d'informations, consultez Configurer un pod pour utiliser un ConfigMap dans la documentation des tâches Kubernetes.

Si vous souhaitez vérifier votre configuration Fluent Bit, procédez comme suit.

Vérifier la configuration Fluent Bit

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. Assurez-vous que vous êtes dans la région où vous avez déployé Fluent Bit.
4. Vérifiez la liste des groupes de journaux dans la région. Vous devez voir ce qui suit :
   - `/aws/containerinsights/Cluster_Name/application`
   - `/aws/containerinsights/Cluster_Name/host`
   - `/aws/containerinsights/Cluster_Name/dataplane`
5. Accédez à l'un de ces groupes de journaux et cochez la case Last Event Time (Heure du dernier événement) pour les flux de journaux. S'il est récent par rapport au moment où vous avez déployé Fluent Bi , la configuration est vérifiée.

   Il peut y avoir un léger retard dans la création du groupe de journaux `/dataplane`. Ceci est normal, car ces groupes de journaux ne sont créés que lorsque Fluent Bit commence à envoyer des journaux pour ce groupe de journaux.

Prise en charge des journaux multilignes

Pour plus d'informations sur l'utilisation de Fluent Bit avec des journaux multilignes, consultez les sections suivantes de la documentation Fluent Bit :

- Multiline Parsing
- Multiline and Containers (v1.8)
- Multiline Core (v1.8)
- Always use multiline in the tail input

(En option) Réduction du volume des journaux de Fluent Bit

Par défaut, nous envoyons les journaux des applications Fluent Bit et les métadonnées Kubernetes à. CloudWatch Si vous souhaitez réduire le volume de données à destination CloudWatch, vous pouvez arrêter l'envoi à l'une de ces sources de données ou aux deux CloudWatch.

Pour arrêter les journaux d'application Fluent Bit, supprimez la section suivante du fichier `Fluent-Bit.yaml`.

```
[INPUT]
        Name                tail
        Tag                 application.*
        Path                /var/log/containers/fluent-bit*
        Parser              docker
        DB                  /fluent-bit/state/flb_log.db
        Mem_Buf_Limit       5MB
        Skip_Long_Lines     On
        Refresh_Interval    10
```

Pour empêcher l'ajout de métadonnées Kubernetes aux événements du journal envoyés à CloudWatch, ajoutez les filtres suivants à la `application-log.conf` section du fichier. `Fluent-Bit.yaml` Remplacez <Metadata_1>les champs similaires par les identificateurs de métadonnées réels.

```
application-log.conf: |
    [FILTER]
        Name                nest
        Match               application.*
        Operation           lift
        Nested_under        kubernetes
        Add_prefix          Kube.

    [FILTER]
        Name                modify
        Match               application.*
        Remove              Kube.<Metadata_1>
        Remove              Kube.<Metadata_2>
        Remove              Kube.<Metadata_3>

    [FILTER]
        Name                nest
        Match               application.*
```

```
        Operation           nest
        Wildcard            Kube.*
        Nested_under        kubernetes
        Remove_prefix       Kube.
```

Résolution des problèmes

Si vous ne voyez pas ces groupes de journaux et si vous les cherchez dans la bonne région, vérifiez les journaux correspondant aux pods du daemonSet Fluent Bit pour identifier l'erreur.

Exécutez la commande suivante et vérifiez que l'état est `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

Si les journaux contiennent des erreurs liées aux autorisations IAM, vérifiez le rôle IAM qui est attaché au nœuds du cluster. Pour plus d'informations sur les autorisations requises pour exécuter un cluster Amazon EKS, consultez Politiques, rôles et autorisations IAM Amazon EKS dans le Guide de l'utilisateur Amazon EKS.

Si l'état du pod est `CreateContainerConfigError`, exécutez la commande suivante pour obtenir l'erreur exacte.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Tableau de bord

Vous pouvez créer un tableau de bord pour surveiller les métriques de chaque plugin en cours d'exécution. Vous pouvez voir les données pour les octets d'entrée et de sortie, et pour les taux de traitement des enregistrements, ainsi que les erreurs de sortie et les taux de nouvelle tentative/échec. Pour consulter ces métriques, vous devez installer l' CloudWatch agent avec la collecte de métriques Prometheus pour les clusters Amazon EKS et Kubernetes. Pour plus d'informations sur la configuration du tableau de bord, consultez Installez l' CloudWatch agent avec la collecte de métriques Prometheus sur les clusters Amazon EKS et Kubernetes.

> **Note**
>
> Avant de pouvoir configurer ce tableau de bord, vous devez configurer Container Insights pour les métriques Prometheus. Pour plus d'informations, consultez Surveillance des métriques Prometheus Container Insights.

Pour créer un tableau de bord pour les métriques Prometheus de Fluent Bit

1. Créez des variables d'environnement, en remplaçant les valeurs à droite dans les lignes suivantes pour correspondre à votre déploiement.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-west-1
CLUSTER_NAME=your_kubernetes_cluster_name
```

2. Créez le tableau de bord en exécutant la commande suivante.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/fluent-bit/cw_dashboard_fluent_bit.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --
dashboard-body
```

(Facultatif) Configurez Fluentd en tant que DaemonSet pour envoyer des journaux à Logs CloudWatch

> **Warning**
>
> Le support de Container Insights pour Fluentd est désormais en mode maintenance, ce qui signifie que cela ne AWS fournira aucune autre mise à jour pour Fluentd et que nous prévoyons de le rendre obsolète dans un futur proche. En outre, la configuration actuelle de Fluentd pour Container Insights utilise une ancienne version de l'image Fluentd `fluent/fluentd-kubernetes-daemonset:v1.10.3-debian-cloudwatch-1.0` qui ne contient pas les dernières améliorations et les derniers correctifs de sécurité. Pour la dernière image Fluentd prise en charge par la communauté open source, voir. fluentd-kubernetes-daemonset
>
> Nous vous recommandons vivement de migrer pour utiliser FluentBit Container Insights dans la mesure du possible. Son utilisation en FluentBit tant que redirecteur de journal pour Container Insights permet d'obtenir des gains de performances significatifs.
>
> Pour plus d'informations, consultez Configurer Fluent Bit comme un DaemonSet pour envoyer des CloudWatch journaux à Logs et Différences si vous utilisez déjà Fluentd.

Pour configurer Fluentd afin de collecter les journaux de vos conteneurs, vous pouvez suivre les étapes de Configuration Quick Start pour Container Insights sur Amazon EKS et Kubernetes ou celles de cette section. Dans les étapes suivantes, vous configurez Fluentd DaemonSet pour envoyer des journaux à CloudWatch Logs. Une fois que vous avez terminé cette étape, Fluentd crée les groupes de journaux suivants s'ils n'existent pas déjà.

Nom du groupe de journaux	Source des journaux
`/aws/containerinsights/Cluster_Name/application`	Tous les fichiers journaux situés dans `/var/log/containers`
`/aws/containerinsights/Cluster_Name/host`	Journaux provenant de `/var/log/dmesg` , `/var/log/secure` et `/var/log/messages`
`/aws/containerinsights/Cluster_Name/dataplane`	Les journaux dans `/var/log/journal` pour `kubelet.service` , `kubeproxy.service` et `docker.service` .

Étape 1 : créer un espace de noms pour CloudWatch

Procédez comme suit pour créer l'espace de noms Kubernetes demandé. `amazon-cloudwatch` CloudWatch Vous pouvez ignorer cette étape si vous avez déjà créé cet espace de noms.

Pour créer un espace de noms pour CloudWatch

- Entrez la commande suivante.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/cloudwatch-namespace.yaml
```

Étape 2 : Installer Fluentd

Pour commencer cette procédure, téléchargez Fluentd. Lorsque vous avez terminé cette procédure, le déploiement crée les ressources suivantes sur le cluster :

- Un compte de service nommé `fluentd` dans l'espace de noms `amazon-cloudwatch`. Ce compte de service est utilisé pour exécuter le Fluentd DaemonSet. Pour plus d'informations,

consultez Managing Service Accounts (Gestion des comptes de service) dans la documentation Reference de Kubernetes.

- Un rôle de cluster rôle nommé `fluentd` dans l'espace de noms `amazon-cloudwatch`. Ce rôle de cluster octroie des autorisations `get`, `list` et `watch` sur les journaux de pod au compte de service `fluentd`. Pour plus d'informations, consultez API Overview (Présentation de l'API) dans la documentation Reference de Kubernetes.task
- Un ConfigMap nommé `fluentd-config` dans l'espace de `amazon-cloudwatch` noms. Cela ConfigMap contient la configuration à utiliser par Fluentd. Pour plus d'informations, consultez Configurer un pod pour utiliser un ConfigMap dans la documentation des tâches Kubernetes.

Pour installer Fluentd

1. Créez un ConfigMap nom `cluster-info` avec le nom du cluster et la AWS région vers laquelle les journaux seront envoyés. Exécutez la commande suivante, en mettant à jour les espaces réservés avec les noms de votre cluster et de votre région.

```
kubectl create configmap cluster-info \
--from-literal=cluster.name=cluster_name \
--from-literal=logs.region=region_name -n amazon-cloudwatch
```

2. Téléchargez et déployez le Fluentd DaemonSet sur le cluster en exécutant la commande suivante. Assurez-vous d'utiliser l'image de conteneur avec la bonne architecture. L'exemple de manifeste ne fonctionne que sur les instances x86 et saisit `CrashLoopBackOff` si vous avez des instances ARM (Advanced RISC Machine) dans votre cluster. Le Daemonset Fluentd ne possède pas d'image Docker multi-architecture officielle qui vous permet d'utiliser une balise pour plusieurs images sous-jacentes et de laisser le réseau d'exécution du conteneur extraire la bonne. L'image ARM Fluentd utilise une balise différente avec un suffixe `arm64`.

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/
daemonset/container-insights-monitoring/fluentd/fluentd.yaml
```

> **Note**
>
> En raison d'une modification récente visant à optimiser la configuration Fluentd et à minimiser l'impact des requêtes d'API de Fluentd sur les points de terminaison de l'API

Kubernetes, l'option « Watch » (surveiller) pour les filtres Kubernetes a été désactivée par défaut. Pour plus de détails, consultez fluent-plugin-kubernetes_metadata_filter.

3. Validez le déploiement en exécutant la commande suivante. Chaque nœud doit avoir un pod nommé `fluentd-cloudwatch-*`.

```
kubectl get pods -n amazon-cloudwatch
```

Étape 3 : Vérifier la configuration de Fluentd

Pour vérifier votre configuration Fluentd, procédez comme suit.

Pour vérifier la configuration Fluentd pour Container Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Groupes de journaux. Assurez-vous que vous êtes dans la région où vous avez déployé Fluentd sur vos conteneurs.

   Dans la liste des groupes de journaux associée à cette région, vous devez voir ce qui suit :

   - `/aws/containerinsights/Cluster_Name/application`
   - `/aws/containerinsights/Cluster_Name/host`
   - `/aws/containerinsights/Cluster_Name/dataplane`

   Si vous voyez ces groupes de journaux, la vérification de la configuration Fluentd est terminée.

Prise en charge des journaux multilignes

Le 19 août 2019, nous avons ajouté la prise en charge des journaux multilignes pour les journaux collectés par Fluentd.

Par défaut, le déclencheur d'entrée de journal multiligne est n'importe quel caractère sans espace. Cela signifie que toutes les lignes de journal qui commencent par un caractère sans espace sont considérées comme une nouvelle entrée de journal multiligne.

Si vos propres journaux d'application utilisent un déclencheur multiligne différent, vous pouvez les prendre en charge en apportant deux modifications dans le fichier `fluentd.yaml`.

Tout d'abord, excluez-les de la prise en charge multiligne par défaut en ajoutant les noms de chemin de vos fichiers journaux dans un champ `exclude_path` de la section `containers` de `fluentd.yaml`. Voici un exemple.

```
<source>
  @type tail
  @id in_tail_container_logs
  @label @containers
  path /var/log/containers/*.log
  exclude_path ["full_pathname_of_log_file*", "full_pathname_of_log_file2*"]
```

Ensuite, ajoutez un bloc pour vos fichiers journaux au fichier `fluentd.yaml`. L'exemple ci-dessous est utilisé pour le fichier journal de l' CloudWatch agent, qui utilise une expression régulière d'horodatage comme point de départ multiligne. Vous pouvez copier ce bloc et l'ajouter à `fluentd.yaml`. Modifiez les lignes indiquées pour refléter le nom du fichier journal de votre application et le déclencheur multiligne que vous souhaitez utiliser.

```
<source>
  @type tail
  @id in_tail_cwagent_logs
  @label @cwagentlogs
  path /var/log/containers/cloudwatch-agent*
  pos_file /var/log/cloudwatch-agent.log.pos
  tag *
  read_from_head true
 <parse>
    @type json
    time_format %Y-%m-%dT%H:%M:%S.%NZ
 </parse>
</source>
```

```
<label @cwagentlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_cwagent
  </filter>

  <filter **>
```

```
    @type record_transformer
    @id filter_cwagent_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
 </filter>

  <filter **>
    @type concat
    key log
    multiline_start_regexp /^\d{4}[-/]\d{1,2}[-/]\d{1,2}/
    separator ""
    flush_interval 5
    timeout_label @NORMAL
 </filter>

 <match **>
    @type relabel
    @label @NORMAL
 </match>
</label>
```

(En option) Réduction du volume des journaux de Fluentd

Par défaut, nous envoyons les journaux des applications Fluentd et les métadonnées Kubernetes à. CloudWatch Si vous souhaitez réduire le volume de données à destination CloudWatch, vous pouvez arrêter l'envoi à l'une de ces sources de données ou aux deux CloudWatch.

Pour arrêter les journaux d'application Fluentd, supprimez la section suivante du fichier `fluentd.yaml`.

```
<source>
  @type tail
  @id in_tail_fluentd_logs
  @label @fluentdlogs
  path /var/log/containers/fluentd*
  pos_file /var/log/fluentd.log.pos
  tag *
  read_from_head true
   <parse>
   @type json
   time_format %Y-%m-%dT%H:%M:%S.%NZ
```

```
  </parse>
</source>

<label @fluentdlogs>
  <filter **>
    @type kubernetes_metadata
    @id filter_kube_metadata_fluentd
  </filter>

  <filter **>
    @type record_transformer
    @id filter_fluentd_stream_transformer
    <record>
      stream_name ${tag_parts[3]}
    </record>
   </filter>

   <match **>
     @type relabel
     @label @NORMAL
   </match>
</label>
```

Pour empêcher l'ajout de métadonnées Kubernetes aux événements du journal envoyés à CloudWatch, ajoutez une ligne à la `record_transformer` section du fichier. `fluentd.yaml` Dans la source du journal dans laquelle vous souhaitez supprimer ces métadonnées, ajoutez la ligne suivante.

```
remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
```

Par exemple :

```
<filter **>
  @type record_transformer
  @id filter_containers_stream_transformer
  <record>
    stream_name ${tag_parts[3]}
  </record>
  remove_keys $.kubernetes.pod_id, $.kubernetes.master_url,
 $.kubernetes.container_image_id, $.kubernetes.namespace_id
```

```
</filter>
```

Résolution des problèmes

Si vous ne voyez pas ces groupes de journaux et que vous recherchez dans la bonne région, consultez les journaux des DaemonSet pods Fluentd pour rechercher l'erreur.

Exécutez la commande suivante et vérifiez que l'état est `Running`.

```
kubectl get pods -n amazon-cloudwatch
```

Dans les résultats de la commande précédente, notez le nom du pod qui commence par `fluentd-cloudwatch`. Utilisez ce nom de pod dans la commande suivante.

```
kubectl logs pod_name -n amazon-cloudwatch
```

Si les journaux contiennent des erreurs liées aux autorisations IAM, vérifiez le rôle IAM attaché au nœuds du cluster. Pour plus d'informations sur les autorisations requises pour exécuter un cluster Amazon EKS, consultez Politiques, rôles et autorisations IAM Amazon EKS dans le Guide de l'utilisateur Amazon EKS.

Si l'état du pod est `CreateContainerConfigError`, exécutez la commande suivante pour obtenir l'erreur exacte.

```
kubectl describe pod pod_name -n amazon-cloudwatch
```

Si l'état du pod est `CrashLoopBackOff`, assurez-vous que l'architecture de l'image du conteneur Fluentd est la même que le nœud lorsque vous avez installé Fluentd. Si votre cluster possède à la fois des nœuds x86 et ARM64, vous pouvez utiliser une étiquette kubernetes.io/arch pour placer les images sur le bon nœud. Pour plus d'informations, consultez kuberntes.io/arch.

(En option) Configurez la journalisation de plan de contrôle Amazon EKS

Si vous utilisez Amazon EKS, vous pouvez éventuellement activer la journalisation du plan de contrôle Amazon EKS, afin de fournir des journaux d'audit et de diagnostic directement depuis le plan de contrôle Amazon EKS vers CloudWatch Logs. Pour plus d'informations, consultez Journalisation de plan de contrôle Amazon EKS.

(En option) Activez les journaux d'accès App Mesh Envoy.

Vous pouvez configurer Container Insights Fluentd pour envoyer les journaux d'accès d'App Mesh Envoy à CloudWatch Logs. Pour plus d'informations, consultez Journalisation.

Pour que les journaux d'accès d'Envoy soient envoyés à CloudWatch Logs

1. Configurez Fluentd dans le cluster. Pour plus d'informations, consultez (Facultatif) Configurez Fluentd en tant que DaemonSet pour envoyer des journaux à Logs CloudWatch .
2. Configurez les journaux d'accès Envoy pour vos nœuds virtuels. Pour obtenir des instructions, consultez Journalisation. Assurez-vous de configurer le chemin d'accès aux journaux sur **`/dev/stdout`** dans chaque nœud virtuel.

Lorsque vous avez terminé, les journaux d'accès Envoy sont envoyés au groupe de journaux `/aws/containerinsights/`*`Cluster_Name`*`/application`.

(En option) Activez la fonction Use_Kubelet pour les grands clusters

Par défaut, la fonctionnalité Use_Kubelet est désactivée dans le plugin Kubernetes. FluentBit L'activation de cette fonction peut réduire le trafic vers le serveur API et atténuer le problème du goulot d'étranglement du serveur API. Nous vous recommandons d'activer cette fonction pour les grands clusters.

Pour activer Use_Kubelet, ajoutez d'abord les nœuds et les autorisations de nœuds et de proxy à la configuration ClusterRole.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: fluent-bit-role
rules:
  - nonResourceURLs:
      - /metrics
    verbs:
      - get
  - apiGroups: [""]
    resources:
      - namespaces
      - pods
      - pods/logs
      - nodes
```

```
      - nodes/proxy
    verbs: ["get", "list", "watch"]
```

Dans la DaemonSet configuration, cette fonctionnalité nécessite un accès au réseau hôte. La version de l'image pour `amazon/aws-for-fluent-bit` doit être 2.12.0 ou une version ultérieure, ou la version de l'image du bit fluent doit être 1.7.2 ou une version ultérieure.

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluent-bit
  namespace: amazon-cloudwatch
  labels:
    k8s-app: fluent-bit
    version: v1
    kubernetes.io/cluster-service: "true"
spec:
  selector:
    matchLabels:
      k8s-app: fluent-bit
  template:
    metadata:
      labels:
        k8s-app: fluent-bit
        version: v1
        kubernetes.io/cluster-service: "true"
    spec:
      containers:
      - name: fluent-bit
        image: amazon/aws-for-fluent-bit:2.19.0
        imagePullPolicy: Always
        env:
            - name: AWS_REGION
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: logs.region
            - name: CLUSTER_NAME
              valueFrom:
                configMapKeyRef:
                  name: fluent-bit-cluster-info
                  key: cluster.name
            - name: HTTP_SERVER
```

```
            valueFrom:
              configMapKeyRef:
                name: fluent-bit-cluster-info
                key: http.server
          - name: HTTP_PORT
            valueFrom:
              configMapKeyRef:
                name: fluent-bit-cluster-info
                key: http.port
          - name: READ_FROM_HEAD
            valueFrom:
              configMapKeyRef:
                name: fluent-bit-cluster-info
                key: read.head
          - name: READ_FROM_TAIL
            valueFrom:
              configMapKeyRef:
                name: fluent-bit-cluster-info
                key: read.tail
          - name: HOST_NAME
            valueFrom:
              fieldRef:
                fieldPath: spec.nodeName
          - name: HOSTNAME
            valueFrom:
              fieldRef:
                apiVersion: v1
                fieldPath: metadata.name
          - name: CI_VERSION
            value: "k8s/1.3.8"
        resources:
            limits:
              memory: 200Mi
            requests:
              cpu: 500m
              memory: 100Mi
        volumeMounts:
        # Please don't change below read-only permissions
        - name: fluentbitstate
          mountPath: /var/fluent-bit/state
        - name: varlog
          mountPath: /var/log
          readOnly: true
        - name: varlibdockercontainers
```

```
          mountPath: /var/lib/docker/containers
          readOnly: true
        - name: fluent-bit-config
          mountPath: /fluent-bit/etc/
        - name: runlogjournal
          mountPath: /run/log/journal
          readOnly: true
        - name: dmesg
          mountPath: /var/log/dmesg
          readOnly: true
      terminationGracePeriodSeconds: 10
      hostNetwork: true
      dnsPolicy: ClusterFirstWithHostNet
      volumes:
      - name: fluentbitstate
        hostPath:
          path: /var/fluent-bit/state
      - name: varlog
        hostPath:
          path: /var/log
      - name: varlibdockercontainers
        hostPath:
          path: /var/lib/docker/containers
      - name: fluent-bit-config
        configMap:
          name: fluent-bit-config
      - name: runlogjournal
        hostPath:
          path: /run/log/journal
      - name: dmesg
        hostPath:
          path: /var/log/dmesg
      serviceAccountName: fluent-bit
      tolerations:
      - key: node-role.kubernetes.io/master
        operator: Exists
        effect: NoSchedule
      - operator: "Exists"
        effect: "NoExecute"
      - operator: "Exists"
        effect: "NoSchedule"
```

La configuration du plugin Kubernetes doit ressembler à ce qui suit :

```
[FILTER]
        Name                kubernetes
        Match               application.*
        Kube_URL            https://kubernetes.default.svc:443
        Kube_Tag_Prefix     application.var.log.containers.
        Merge_Log           On
        Merge_Log_Key       log_processed
        K8S-Logging.Parser  On
        K8S-Logging.Exclude Off
        Labels              Off
        Annotations         Off
        Use_Kubelet         On
        Kubelet_Port        10250
        Buffer_Size         0
```

Mise à jour ou suppression de Container Insights sur Amazon EKS et Kubernetes

Suivez les étapes décrites dans ces sections pour mettre à jour l'image du conteneur de votre CloudWatch agent ou pour supprimer Container Insights d'un cluster Amazon EKS ou Kubernetes.

Rubriques

- Mise à niveau vers Container Insights avec observabilité améliorée pour Amazon EKS
- Mise à jour de l'image du conteneur de l' CloudWatch agent
- Suppression de l' CloudWatch agent et de Fluent Bit for Container Insights

Mise à niveau vers Container Insights avec observabilité améliorée pour Amazon EKS

> **Important**
>
> Si vous mettez à niveau ou installez Container Insights sur un cluster Amazon EKS, nous vous recommandons d'utiliser le module complémentaire Amazon CloudWatch Observability EKS pour l'installation, au lieu de suivre les instructions de cette section. En outre, pour récupérer des métriques de calcul accéléré, vous devez utiliser le module complémentaire Amazon CloudWatch Observability EKS. Pour en savoir plus et des instructions, consultez Installez le module complémentaire Amazon CloudWatch Observability EKS.

Container Insights avec observabilité améliorée pour Amazon EKS est la dernière version de Container Insights. Il collecte des métriques détaillées à partir de clusters exécutant Amazon EKS et

propose des tableaux de bord élaborés et immédiatement exploitables pour approfondir la télémétrie des applications et des infrastructures. Pour plus d'informations sur cette version de Container Insights, veuillez consulter Container Insights avec observabilité améliorée pour Amazon EKS.

Si vous avez installé la version originale de Container Insights dans un cluster Amazon EKS et que vous souhaitez la mettre à niveau vers la version la plus récente avec observabilité améliorée, suivez les instructions de cette section.

> **Important**
>
> Avant de terminer les étapes de cette section, vous devez avoir vérifié les prérequis, y compris le gestionnaire de certificats. Pour plus d'informations, consultez Démarrage rapide avec l'opérateur de l' CloudWatch agent et Fluent Bit.

Pour mettre à niveau un cluster Amazon EKS vers Container Insights avec observabilité améliorée pour Amazon EKS

1. Installez l'opérateur de l' CloudWatch agent en saisissant la commande suivante. *my-cluster-name*Remplacez-le par le nom de votre cluster Amazon EKS ou Kubernetes, puis par le nom *my-cluster-region*de la région dans laquelle les journaux sont publiés. Nous vous recommandons d'utiliser la même région que celle dans laquelle votre cluster est déployé afin de réduire les coûts de transfert de données AWS sortants.

   ```
   ClusterName=my-cluster-name
   RegionName=my-cluster-region
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
   container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
   {{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
    apply -f -
   ```

   Si vous remarquez une panne due à un conflit de ressources, c'est probablement parce que l' CloudWatch agent et Fluent Bit ainsi que les composants associés tels que le ServiceAccount, le ClusterRole et le sont déjà ClusterRoleBinding installés sur le cluster. Lorsque l'opérateur de l' CloudWatch agent essaie d'installer l' CloudWatch agent et ses composants associés, s'il détecte une modification du contenu, il échoue par défaut à l'installation ou à la mise à jour pour éviter de modifier l'état des ressources du cluster. Nous vous recommandons de supprimer tout CloudWatch agent existant avec une configuration de Container Insights que vous aviez précédemment installé sur le cluster, puis d'installer l'opérateur de l' CloudWatch agent.

2. (Facultatif) Pour appliquer une configuration Fluent Bit personnalisée existante, vous devez mettre à jour le fichier de configuration associé au daemonset Fluent Bit. L'opérateur de l' CloudWatch agent fournit une configuration par défaut pour Fluent Bit, et vous pouvez remplacer ou modifier la configuration par défaut selon vos besoins. Pour appliquer une configuration personnalisée, procédez comme suit.

   a. Ouvrez la configuration existante en saisissant la commande suivante.

   ```
   kubectl edit cm fluent-bit-config -n amazon-cloudwatch
   ```

   b. Apportez vos modifications dans le fichier, puis entrez `:wq` pour enregistrer le fichier et quittez le mode d'édition.

   c. Redémarrez Fluent Bit en saisissant la commande suivante.

   ```
   kubectl rollout restart fluent-bit -n amazon-cloudwatch
   ```

Mise à jour de l'image du conteneur de l' CloudWatch agent

> **Important**
>
> Si vous mettez à niveau ou installez Container Insights sur un cluster Amazon EKS, nous vous recommandons d'utiliser le module complémentaire Amazon CloudWatch Observability EKS pour l'installation, au lieu de suivre les instructions de cette section. En outre, pour récupérer des métriques de calcul accéléré, vous devez utiliser le module complémentaire Amazon CloudWatch Observability EKS ou l'opérateur de l' CloudWatch agent. Pour en savoir plus et des instructions, consultez Installez le module complémentaire Amazon CloudWatch Observability EKS.

Si vous avez besoin de mettre à jour votre image de conteneur vers la dernière version, suivez les étapes de cette section.

Pour mettre à jour votre image de conteneur

1. Vérifiez si la définition de ressource `amazoncloudwatchagent` client (CRD) existe déjà en saisissant la commande suivante.

```
kubectl get crds amazoncloudwatchagents.cloudwatch.aws.amazon.com -n amazon-
cloudwatch
```

Si cette commande renvoie une erreur indiquant que le CRD est absent, Container Insights with Enhanced Observability for Amazon EKS n'est pas configuré avec l'opérateur de l'agent sur le CloudWatch cluster. Dans ce cas, consultez Mise à niveau vers Container Insights avec observabilité améliorée pour Amazon EKS.

2. Appliquez le dernier fichier `cwagent-version.yaml` en entrant la commande suivante.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-version.yaml | kubectl apply -f -
```

Suppression de l' CloudWatch agent et de Fluent Bit for Container Insights

Si vous avez installé Container Insights en installant le module complémentaire CloudWatch Observability pour Amazon EKS, vous pouvez supprimer Container Insights et l' CloudWatch agent en saisissant la commande suivante :

> **Note**
>
> Le module complémentaire Amazon EKS prend désormais en charge Container Insights sur les nœuds de travail Windows. Si vous supprimez le module complémentaire Amazon EKS, Container Insights pour Windows est également supprimé.

```
aws eks delete-addon –cluster-name my-cluster –addon-name amazon-cloudwatch-
observability
```

Sinon, pour supprimer toutes les ressources liées à l' CloudWatch agent et à Fluent Bit, entrez la commande suivante. Dans cette commande, *My_Cluster_Name est le nom* de votre cluster Amazon EKS ou Kubernetes, et *My_Region est le nom de la région dans laquelle* les journaux sont publiés.

```
ClusterName=My_Cluster_Name
RegionName=My-Region
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/main/k8s-quickstart/cwagent-operator-rendered.yaml | sed 's/
```

```
{{cluster_name}}/'${ClusterName}'/g;s/{{region_name}}/'${RegionName}'/g' | kubectl
 delete -f -
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/main/k8s-quickstart/cwagent-custom-resource-definitions.yaml | kubectl delete
 -f -
```

Affichage des métriques dans Container Insights

Une fois que Container Insights est configuré et qu'il collecte des métriques, vous pouvez consulter ces métriques dans la CloudWatch console.

Pour que les métriques Container Insights apparaissent sur votre tableau de bord, vous devez effectuer la configuration de Container Insights. Pour plus d'informations, consultez . Configuration de Container Insights.

Cette procédure explique comment afficher les métriques générées automatiquement par Container Insights à partir des données collectées dans le journal. Le reste de cette section explique comment approfondir l'analyse de vos données et utiliser CloudWatch Logs Insights pour consulter davantage de statistiques à des niveaux de granularité plus élevés.

Pour afficher les métriques dans Container Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Insights, puis choisissez Container Insights.
3. Dans la liste déroulante sous Container Insights, sélectionnez Surveillance des performances.
4. Utilisez les listes déroulantes situées en haut pour sélectionner le type de ressource à afficher, ainsi que la ressource spécifique.

Vous pouvez définir une CloudWatch alarme pour n'importe quel indicateur collecté par Container Insights. Pour plus d’informations, consultez Utilisation des CloudWatch alarmes Amazon.

> **Note**
>
> Si vous avez déjà configuré CloudWatch Application Insights pour surveiller vos applications conteneurisées, le tableau de bord Application Insights apparaît sous le tableau de bord Container Insights. Si vous n'avez pas encore activé Application Insights, vous pouvez le faire en choisissantConfiguration automatique de l'application Insightssous la vue des performances dans le tableau de bord Container Insights.

Pour plus d'informations sur Application Insight et les applications conteneurisées, consultezActiver la surveillance des ressources Application Insights pour Amazon ECS et Amazon EKS.

Affichage des principaux contributeurs

Pour certaines vues de la surveillance des performances de Container Insights, vous pouvez également voir les principaux contributeurs par mémoire ou processeur, ou les ressources les plus récemment actives. Cette option est disponible lorsque vous sélectionnez l'un des tableaux de bord suivants dans la liste déroulante située en haut de la page :

- Services ECS
- Tâches ECS
- Espaces de noms EKS
- Services EKS
- Pods EKS

Lorsque vous affichez l'un de ces types de ressources, le bas de la page affiche un tableau trié initialement par utilisation du processeur. Vous pouvez le modifier pour trier en fonction de l'utilisation de la mémoire ou de l'activité récente. Pour en savoir plus sur l'une des lignes du tableau, vous pouvez cocher la case en regard de celle-ci, puis choisir Actions et choisir l'une des options dans le menu Actions.

Utilisation de CloudWatch Logs Insights pour consulter les données de Container Insights

Container Insights collecte des métrique en utilisant les évènements des journaux de performances utilisant le format de métrique intégrée. Les journaux sont stockés dans CloudWatch des journaux. CloudWatch génère automatiquement plusieurs métriques à partir des journaux que vous pouvez consulter dans la CloudWatch console. Vous pouvez également effectuer une analyse plus approfondie des données de performance collectées à l'aide des requêtes CloudWatch Logs Insights.

Pour plus d'informations sur CloudWatch Logs Insights, voir Analyser les données des CloudWatch journaux avec Logs Insights. Pour plus d'informations sur les champs de journaux que vous pouvez utiliser dans les requêtes, consultez Évènements du journaal de performances de Container Insights pour Amazon EKS et Kubernetes.

Pour utiliser CloudWatch Logs Insights pour interroger les données métriques de votre conteneur

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

   Près du haut de l'écran se trouve l'éditeur de requête. Lorsque vous ouvrez CloudWatch Logs Insights pour la première fois, cette zone contient une requête par défaut qui renvoie les 20 événements de journal les plus récents.
3. Dans la zone au-dessus de l'éditeur de requête, sélectionnez l'un des groupes de journaux Container Insights sur lequel faire porter la requête. Pour que les exemples de requêtes suivants fonctionnent, le nom du groupe de journaux doit se terminer par performance.

   Lorsque vous sélectionnez un groupe de CloudWatch journaux, Logs Insights détecte automatiquement les champs des données du groupe de journaux et les affiche dans la section Champs découverts dans le volet droit. Il affiche également un graphique à barres des événements de journaux dans ce groupe de journaux au fil du temps. Ce graphique à barres montre la distribution des événements dans le groupe de journaux correspondant à vos requêtes et plages de temps et pas seulement les événements affichés dans le tableau.
4. Dans l'éditeur de requête, remplacez la requête par défaut par la requête suivante, puis choisissez Run query (Exécuter la requête).

   ```
   STATS avg(node_cpu_utilization) as avg_node_cpu_utilization by NodeName
   | SORT avg_node_cpu_utilization DESC
   ```

   Cette requête affiche une liste de nœuds, triés en fonction de leur utilisation moyenne de l'UC.
5. Pour voir un autre exemple, remplacez cette requête par une autre, puis choisissez Run query (Exécuter la requête). D'autres exemples de requêtes sont répertoriés plus loin sur cette page.

   ```
   STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by
    PodName
   | SORT avg_number_of_container_restarts DESC
   ```

   Cette requête affiche une liste de vos pods, triés en fonction du nombre moyen de redémarrages de conteneurs.
6. Si vous souhaitez essayer une autre requête, vous pouvez utiliser les champs d'inclusion répertoriés dans la liste à droite de l'écran. Pour plus d'informations sur la syntaxe des requêtes, voir Syntaxe de requête de CloudWatch Logs Insights.

Pour afficher les listes de vos ressources

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Resources (Ressources).
3. Par défaut une liste de vos ressources surveillées par Container Insights et des alertes que vous avez définies sur ces ressources s'affiche. Pour afficher une carte visuelle des ressources, choisissez Vue Carte.
4. Dans la vue Carte, vous pouvez placer le pointeur sur n'importe quelle ressource pour afficher les métriques de base relatives à cette ressource. Vous pouvez choisir n'importe quelle ressource pour afficher des graphiques plus détaillés sur la ressource.

Cas d'utilisation : Affichage des métriques au niveau des tâches dans les conteneurs Amazon ECS

L'exemple suivant montre comment utiliser CloudWatch Logs Insights pour approfondir vos logs Container Insights. Pour plus d'exemples, consultez le blog Présentation d'Amazon CloudWatch Container Insights pour Amazon ECS.

Container Insights ne génère pas automatiquement de métriques au niveau de granularité Tâche. La requête suivante affiche les métriques au niveau des tâches pour l'utilisation du processeur et de la mémoire.

```
stats avg(CpuUtilized) as CPU, avg(MemoryUtilized) as Mem by TaskId, ContainerName
| sort Mem, CPU desc
```

Autres exemples de requêtes pour Container Insights

Liste de vos pods, triés par nombre moyen de redémarrages de conteneur

```
STATS avg(number_of_container_restarts) as avg_number_of_container_restarts by PodName
| SORT avg_number_of_container_restarts DESC
```

Pods demandés et pods en cours d'exécution

```
fields @timestamp, @message
| sort @timestamp desc
| filter Type="Pod"
```

```
| stats min(pod_number_of_containers) as requested,
 min(pod_number_of_running_containers) as running, ceil(avg(pod_number_of_containers-
pod_number_of_running_containers)) as pods_missing by kubernetes.pod_name
| sort pods_missing desc
```

Nombre de défaillances de nœuds de cluster

```
stats avg(cluster_failed_node_count) as CountOfNodeFailures
| filter Type="Cluster"
| sort @timestamp desc
```

Erreurs de journal d'application par nom de conteneur

```
stats count() as countoferrors by kubernetes.container_name
| filter stream="stderr"
| sort countoferrors desc
```

Métriques collectées par Container Insights

Container Insights collecte un ensemble de métriques pour Amazon ECS et AWS Fargate Amazon ECS, et un ensemble différent pour Amazon EKS, AWS Fargate Amazon EKS et Kubernetes.

Les métriques ne sont pas visibles tant que les tâches du conteneur ne sont pas en cours d'exécution depuis un certain temps.

Rubriques

- Métriques de Container Insights pour Amazon ECS
- Métriques Container Insights pour Amazon EKS et Kubernetes

Métriques de Container Insights pour Amazon ECS

Le tableau suivant répertorie les métriques et les dimensions collectées par Container Insights pour Amazon ECS. Ces métriques sont dans l'espace de noms `ECS/ContainerInsights`. Pour plus d'informations, consultez . Métriques.

Si vous ne voyez pas toutes les métriques Container Insights dans votre console, assurez-vous que vous avez terminé la configuration de Container Insights. Les métriques n'apparaissent pas tant que Container Insights n'a pas été complètement configuré. Pour plus d’informations, consultez Configuration de Container Insights.

Les métriques suivantes sont disponibles lorsque vous effectuez les étapes de la section Configuration de Container Insights sur Amazon ECS pour les métriques de niveau de cluster et de niveau de service

Nom de la métrique	Dimensions	Description
`ContainerInstanceCount`	ClusterName	Nombre d'instances EC2 exécutant l'agent Amazon ECS qui sont enregistrées auprès d'un cluster. Cette métrique n'est collectée que pour les instances de conteneur qui exécutent des tâches Amazon ECS dans le cluster. Elle n'est pas collectée pour les instances de conteneur vides ne comportant aucune tâche Amazon ECS. Unité : nombre

`CpuUtilized`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Unités UC utilisées par les tâches de la ressource spécifiée par l'ensemble de dimensions que vous utilisez.
		
		Cette métrique est collectée uniquement pour les tâches qui ont une réservation d'UC définie dans leur définition de tâche.

Nom de la métrique	Dimensions	Description
		Unité : aucune
CpuReserved	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Les unités UC réservée par les tâches de la ressource qui est spécifié par la dimension définie que vous utilisez. Cette métrique est collectée uniquement pour les tâches qui ont une réservation d'UC définie dans leur définitio n de tâche. Unité : aucune
DeploymentCount	ServiceName , ClusterName	Nombre de déploieme nts dans un Amazon ECS service. Unité : nombre
DesiredTaskCount	ServiceName , ClusterName	Nombre souhaité de tâches pour un Amazon ECS service. Unité : nombre

Nom de la métrique	Dimensions	Description
EBSFilesystemSize	VolumeName , TaskDefinitionFamily , ClusterName TaskDefinitionFamily , ClusterName ServiceName , ClusterName	La quantité totale, en gigaoctets (Go), de stockage du système de fichiers Amazon EBS allouée aux ressources spécifiées par les dimensions que vous utilisez. Cette métrique n'est disponible que pour les tâches exécutées sur l'infrastructure Amazon ECS exécutée sur Fargate à l'aide de la version de plate-forme ou sur les instances 1.4.0 Amazon EC2 utilisant la version de l'agent de conteneur ou une version ultérieure. 1.79.0 Unité : Gigaoctets (Go)

Nom de la métrique	Dimensions	Description
EBSFilesystemUtilized	VolumeName , TaskDefinitionFamily , ClusterName TaskDefinitionFamily , ClusterName ServiceName , ClusterName	La quantité totale, en gigaoctets (Go), du stockage du système de fichiers Amazon EBS utilisé par les ressources spécifiées par les dimensions que vous utilisez. Cette métrique n'est disponible que pour les tâches exécutées sur l'infrastructure Amazon ECS exécutée sur Fargate à l'aide de la version de plate-forme ou sur les instances 1.4.0 Amazon EC2 utilisant la version de l'agent de conteneur ou une version ultérieure. 1.79.0 Pour les tâches exécutées sur Fargate, Fargate réserve de l'espace sur le disque uniquement utilisé par Fargate. Il n'y a aucun coût associé à l'espace utilisé par Fargate, mais vous pourrez voir ce stockage supplémentaire à l'aide d'outils tels que. df

Nom de la métrique	Dimensions	Description
		Unité : Gigaoctets (Go)
EphemeralStorageReserved [1]	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Le nombre d'octets réservés au stockage éphémère dans la ressource qui est spécifiée par les dimensions que vous utilisez. Le stockage éphémère est utilisé pour le système de fichiers racine du conteneur et tout volume hôte de montage lié défini dans l'image du conteneur et la définition de la tâche. La quantité de stockage éphémère ne peut pas être modifiée dans une tâche en cours d'exécution. Cette métrique est uniquement disponible pour les tâches qui s'exécutent sur la plateforme Linux Fargate version 1.4.0 ou ultérieure. Unité : Gigaoctets (Go)

Nom de la métrique	Dimensions	Description

`EphemeralStorageUtilized` [1]	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Le nombre d'octets utilisés à partir du stockage éphémère dans la ressource qui est spécifiée par les dimensions que vous utilisez. Le stockage éphémère est utilisé pour le système de fichiers racine du conteneur et tout volume hôte de montage lié défini dans l'image du conteneur et la définition de la tâche. La quantité de stockage éphémère ne peut pas être modifiée dans une tâche en cours d'exécution.
		
		Cette métrique est uniquement disponible pour les tâches qui s'exécutent sur la plateforme Linux Fargate version 1.4.0 ou ultérieure.
		
		Unité : Gigaoctets (Go)

Nom de la métrique	Dimensions	Description
MemoryUtilized	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Mémoire utilisée par les tâches de la ressource spécifiée par l'ensemble de dimensions que vous utilisez. Cette métrique est collectée uniquement pour les tâches qui ont une réservation de mémoire définie dans leur définition de tâche. Unité : mégaoctets
MemoryReserved	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Mémoire réservée par les tâches de la ressource qui est spécifiée par l'ensemble de dimensions que vous utilisez. Cette métrique est collectée uniquement pour les tâches qui ont une réservation de mémoire définie dans leur définition de tâche. Unité : mégaoctets

Nom de la métrique	Dimensions	Description

`NetworkRxBytes`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Nombre d'octets reçus par la ressource qui est spécifiée par les dimensions que vous utilisez. Cette métrique est obtenue à partir du moteur d'exécution Docker.
		
		Cette métrique est disponible uniquement pour les conteneurs utilisant les modes réseau `awsvpc` ou `bridge`.
		
		Unité : octets/seconde

`NetworkTxBytes`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Nombre d'octets transmis par la ressource spécifié par les dimensions que vous utilisez. Cette métrique est obtenue à partir du moteur d'exécution Docker.
		
		Cette métrique est disponible uniquement pour les conteneurs utilisant les modes réseau `awsvpc` ou `bridge`.
		
		Unité : octets/seconde

Nom de la métrique	Dimensions	Description
PendingTaskCount	ServiceName , ClusterName	Nombre de tâches actuellement dans l'état PENDING. Unité : nombre
RunningTaskCount	ServiceName , ClusterName	Nombre de tâches actuellement dans l'état RUNNING. Unité : nombre
ServiceCount	ClusterName	Nombre de services dans le cluster. Unité : nombre
StorageReadBytes	TaskDefinitionFamily , ClusterName ServiceName , ClusterName ClusterName	Le nombre d'octets lus à partir du stockage sur l'instance dans la ressource spécifiée par les dimensions que vous utilisez. Cela n'inclut pas les octets de lecture pour vos périphériques de stockage. Cette métrique est obtenue à partir du moteur d'exécution Docker. Unité : octets

Nom de la métrique	Dimensions	Description

`StorageWriteBytes`	`TaskDefinitionFamily` , `ClusterName`
	
	`ServiceName` , `ClusterName`
	
	`ClusterName`
		Nombre d'octets écrits sur le stockage de la ressource spécifiée par les dimensions que vous utilisez. Cette métrique est obtenue à partir du moteur d'exécution Docker.
		
		Unité : octets

`TaskCount`	`ClusterName`	Nombre de tâches en cours d'exécution dans le cluster. Unité : nombre
`TaskSetCount`	`ServiceName` , `ClusterName`	Nombre d'ensembles de tâches dans le service. Unité : nombre

> **Note**
>
> Les métriques `EphemeralStorageReserved` et `EphemeralStorageUtilized` sont uniquement disponibles pour les tâches qui s'exécutent sur la plateforme Fargate Linux version 1.4.0 ou ultérieure.
> Fargate réserve de l'espace sur le disque. Il n'est utilisé que par Fargate. Vous n'êtes pas facturé pour cela. Cela n'apparaît pas dans ces statistiques. Toutefois, vous pouvez voir ce stockage supplémentaire dans d'autres outils tels que `df`.

Les métriques suivantes sont disponibles lorsque vous effectuez les étapes de la section Déploiement de l' CloudWatch agent pour collecter des métriques au niveau de l'instance EC2 sur Amazon ECS

Nom de la métrique	Dimensions	Description
instance_cpu_limit	ClusterName	Nombre maximal d'unités d'UC qui peut être attribué à une seule instance EC2 du cluster. Unité : aucune
instance_cpu_reserved_capacity	ClusterName InstanceId, ContainerInstanceId, ClusterName	Pourcentage d'UC actuellement réservé sur une seule instance EC2 dans le cluster. Unité : pourcentage
instance_cpu_usage_total	ClusterName	Nombre d'unités d'UC utilisées sur une instance EC2 unique dans le cluster. Unité : aucune
instance_cpu_utilization	ClusterName InstanceId, ContainerInstanceId, ClusterName	Pourcentage total d'unités d'UC utilisées sur une seule instance EC2 dans le cluster. Unité : pourcentage
instance_filesystem_utilization	ClusterName InstanceId, ContainerInstanceId, ClusterName	Pourcentage total de la capacité du système de fichiers utilisée sur une seule instance EC2 du cluster.

Nom de la métrique	Dimensions	Description
		Unité : pourcentage
instance_memory_limit	ClusterName	Quantité de mémoire maximale, en octets, qui peut être attribuée à une seule instance EC2 du cluster. Unité : octets
instance_memory_reserved_capacity	ClusterName InstanceId , ContainerInstanceId , ClusterName	Pourcentage de mémoire actuellement réservé sur une seule instance EC2 du cluster. Unité : pourcentage
instance_memory_utilization	ClusterName InstanceId , ContainerInstanceId , ClusterName	Pourcentage total de mémoire utilisé sur une seule instance EC2 du cluster. Unité : pourcentage
instance_memory_working_set	ClusterName	Quantité de mémoire, en octets, utilisée sur une seule instance EC2 du cluster. Unité : octets

Nom de la métrique	Dimensions	Description
`instance_network_total_bytes`	`ClusterName`	Nombre total d'octets par seconde transmis et reçus sur le réseau sur une seule instance EC2 dans le cluster. Unité : octets/seconde
`instance_number_of_running_tasks`	`ClusterName`	Nombre de tâches en cours d'exécution sur une seule instance EC2 dans le cluster. Unité : nombre

Métriques Container Insights pour Amazon EKS et Kubernetes

Les tableaux suivants répertorient les métriques et les dimensions collectées par Container Insights pour Amazon EKS et Kubernetes. Ces métriques sont dans l'espace de noms `ContainerInsights`. Pour plus d'informations, consultez . Métriques.

Si vous ne voyez pas toutes les métriques Container Insights dans votre console, assurez-vous que vous avez terminé la configuration de Container Insights. Les métriques n'apparaissent pas tant que Container Insights n'a pas été complètement configuré. Pour plus d’informations, consultez Configuration de Container Insights.

Si vous utilisez la version 1.5.0 ou ultérieure du module complémentaire Amazon EKS ou la version 1.300035.0 de l' CloudWatch agent, la plupart des métriques répertoriées dans le tableau suivant sont collectées pour les nœuds Linux et Windows. Consultez la colonne Nom de la métrique du tableau pour savoir quelles mesures ne sont pas collectées pour Windows.

Avec la version originale de Container Insights, les métriques sont facturées en tant que métriques personnalisées. Grâce à Container Insights avec observabilité améliorée pour Amazon EKS, les métriques de Container Insights sont facturées par observation au lieu d'être facturées par métrique

stockée ou par journal ingéré. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

> **Note**
>
> Sous Windows, les métriques réseau telles que `pod_network_rx_bytes` et ne `pod_network_tx_bytes` sont pas collectées pour les conteneurs de processus hôtes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
`cluster_failed_node_count`	`ClusterName`		Nombre d'échecs des nœuds de travail dans le cluster. Un nœud est considéré comme ayant échoué s'il souffre de conditions de nœud. Pour plus d'informations, consultez Conditions dans la documentation Kubernetes.
`cluster_node_count`	`ClusterName`		Nombre total de composants master dans le cluster.

`namespace_number_of_running_pods`	`Namespace` `ClusterName`
	
	`ClusterName`
			Nombre de pods exécutés par espace de nom dans la ressource spécifiée

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			par les dimensions que vous utilisez.
`node_cpu_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	Nombre maximal d'unités UC qui peut être attribué à un seul nœud du cluster.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_cpu_reserved_capacity`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Pourcentage d'unités UC qui sont réservées pour les composants de nœud, tels que Kubelet, Kube-proxy et Docker.
			
			Formule : `node_cpu_request / node_cpu_limit`
			
			Note
			`node_cpu_request` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.
`node_cpu_usage_total`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	Nombre d'unités UC en cours d'utilisation sur les nœuds du cluster.

`node_cpu_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Pourcentage total d'unités UC en cours d'utilisation sur les nœuds du cluster.
			
			Formule : `node_cpu_usage_total / node_cpu_limit`

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_filesystem_utilization`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Pourcentage total d'une capacité de système de fichiers utilisée sur les nœuds du cluster.
			
			Formule : `node_filesystem_usage / node_filesystem_capacity`
			
			Note
			`node_filesystem_usage` et `node_filesystem_capacity` ne sont pas indiqués directement sous forme de métrique, mais constituent des champs dans les événement

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			s du journal des performances. Pour plus d'informations, consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.
`node_memory_limit`	`ClusterName`	`ClusterName` , `InstanceId` , `NodeName`	Quantité de mémoire maximale, en octets, qui peut être attribuée à un seul nœud du cluster.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_filesystem_inodes`
 	
Cette métrique est disponible uniquement avec Container Insights avec une observabilité améliorée pour Amazon EKS. Il n'est pas disponible sous Windows.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Le nombre total d'inodes (utilisés et non utilisés) sur un nœud.

`node_filesystem_inodes_free`
 	
Cette métrique est disponible uniquement avec Container Insights avec une observabilité améliorée pour Amazon EKS. Il n'est pas disponible sous Windows.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Le nombre d'inodes non utilisés sur un nœud.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_memory_reserved_capacity`	`NodeName`, `ClusterName`, `InstanceId`
	
	`ClusterName`
			Pourcentage de mémoire actuellement utilisé sur les nœuds du cluster.
			
			Formule : `node_memory_request / node_memory_limit`
			
			Note
			`node_memory_request` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.
node_memory_utilization	NodeName, ClusterName , InstanceId ClusterName		Pourcentage de mémoire actuellement utilisé par le ou les nœuds. Il s'agit du pourcentage d'utilisation de la mémoire du nœud divisé par la limitation de la mémoire du nœud. Formule : node_memory_working_set / node_memory_limit .

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
`node_memory_working_set`	`ClusterName`	`ClusterName`, `InstanceId`, `NodeName`	Quantité de mémoire, en octets, utilisée dans l'ensemble de travail des nœuds du cluster.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_network_total_bytes`	`NodeName`, `ClusterName` , `InstanceId`
	
	`ClusterName`
			Nombre total d'octets transmis et reçus par seconde sur le réseau par nœud dans un cluster.
			
			Formule : `node_network_rx_bytes + node_network_tx_bytes`
			
			Note
			`node_network_rx_bytes` et `node_network_tx_bytes` ne sont pas indiqués directement sous forme de métrique, mais constituent des champs dans les événements du journal des performan

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			ces. Pour plus d'informations, consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

`node_number_of_running_containers`	`NodeName`, `ClusterName`, `InstanceId`
	
	`ClusterName`
			Nombre de conteneurs en cours d'exécution par nœud dans un cluster.

`node_number_of_running_pods`	`NodeName`, `ClusterName`, `InstanceId`
	
	`ClusterName`
			Nombre de pods en cours d'exécution par nœud dans un cluster.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_status_allocatable_pods`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Le nombre de pods pouvant être affectés à un nœud en fonction de ses ressources allouables, défini comme le reste de la capacité d'un nœud après prise en compte des réserves de démons du système et des seuils d'expulsion stricts.

`node_status_capacity_pods`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Le nombre de pods qui peuvent être affectés à un nœud en fonction de sa capacité.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_status_condition_ready`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indique si la condition d'état `Ready` du nœud est vraie.

`node_status_condition_memory_pressure`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indique si la condition d'état `MemoryPressure` du nœud est vraie.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_status_condition_pid_pressure`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indique si la condition d'état `PIDPressure` du nœud est vraie.

`node_status_condition_disk_pressure`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Indique si la condition d'état `OutOfDisk` du nœud est vraie.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_status_condition_unknown`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Indique si l'une des conditions d'état du nœud est inconnue.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_interface_network_rx_dropped`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Le nombre de paquets qui ont été reçus et ensuite abandonnés par une interface de réseau sur le nœud.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_interface_network_tx_dropped`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Le nombre de paquets qui devaient être transmis, mais qui ont été abandonnés par une interface réseau sur le nœud.

`node_diskio_io_service_bytes_total`
 	
Cette métrique est disponible uniquement avec Container Insights avec une observabilité améliorée pour Amazon EKS. Il n'est pas disponible sous Windows.
		`ClusterName`
		
		`ClusterName`, `InstanceId`, `NodeName`
			Le nombre total d'octets transférés par toutes les opérations d'E/S sur le nœud.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`node_diskio_io_serviced_total`
 	
Cette métrique est disponible uniquement avec Container Insights avec une observabilité améliorée pour Amazon EKS. Il n'est pas disponible sous Windows.
		`ClusterName`
		
		`ClusterName` , `InstanceId` , `NodeName`
			Le nombre total d'opérations d'E/S sur le nœud.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_cpu_reserved_capacity`	`PodName`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			Capacité d'UC qui est réservée par pod dans un cluster.
			
			Formule : `pod_cpu_request / node_cpu_limit`
			
			Note
			`pod_cpu_request` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs pertinents dans les

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_cpu_utilization`	`PodName`, Namespace, `ClusterName`
	
	Espace de noms, `ClusterName`
	
	Service, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Pourcentage d'unités UC utilisées par les pods.
			
			Formule : `pod_cpu_usage_total / node_cpu_limit`
			
			Note
			`pod_cpu_usage_total` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs pertinents dans les

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_cpu_utilization_over_pod_limit`	`PodName`, Namespace, `ClusterName`
	
	Espace de noms, `ClusterName`
	
	Service, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Pourcentage d'unités UC en cours d'utilisation par les pods par rapport à la limite des pods.
			
			Formule : `pod_cpu_usage_total / pod_cpu_limit`
			
			Note
			`pod_cpu_usage_total` et `pod_cpu_limit` ne sont pas indiqués directement sous forme de métrique, mais constituent des champs dans les événements du journal des performances. Pour

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			plus d'informations, consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_memory_reserved_capacity`	`PodName`, `Namespace`, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
		
		`ClusterName` , `Namespace` , `Service`
			Pourcentage de mémoire réservé aux pods.
			
			Formule : `pod_memory_request / node_memory_limit`
			
			Note
			`pod_memory_request` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs pertinent

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			s dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_memory_utilization`	`PodName`, `Namespace`, `ClusterName`
	
	Espace de noms, `ClusterName`
	
	Service, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Pourcentage de mémoire actuellement utilisé par le ou les pods.
			
			Formule : `pod_memory_working_set / node_memory_limit`
			
			Note
			`pod_memory_working_set` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_memory_utilization_over_pod_limit`	`PodName`, `Namespace`, `ClusterName`
	
	Espace de noms, `ClusterName`
	
	Service, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Pourcentage de mémoire utilisé par les pods par rapport à la limite des pods. Si l'un des conteneurs dans le pod n'a pas de limite de mémoire définie, cette métrique n'apparaît pas.
			
			Formule : `pod_memory_working_set / pod_memory_limit`
			
			Note
			`pod_memory_working_set` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			des performances. Pour plus d'informations, consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_network_rx_bytes`	`PodName`, `Namespace`, `ClusterName`
	
	Espace de noms, `ClusterName`
	
	Service, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Nombre d'octets reçus par seconde sur le réseau par le pod.
			
			Formule : `sum(pod_interface_network_rx_bytes)`
			
			Note
			`pod_interface_network_rx_bytes` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_network_tx_bytes`	`PodName`, `Namespace`, `ClusterName`
	
	Espace de noms, `ClusterName`
	
	Service, Namespace, `ClusterName`
	
	`ClusterName`
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`	Nombre d'octets transmis par seconde sur le réseau par le pod.
			
			Formule : `sum(pod_interface_network_tx_bytes)`
			
			Note
			`pod_interface_network_tx_bytes` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_cpu_request`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Les requêtes de processeur du pod.
			
			Formule : `sum(container_cpu_request)`
			
			Note
			`pod_cpu_request` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs pertinents dans les évènements

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_memory_request`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Les requêtes de mémoire du pod.
			
			Formule : `sum(container_memory_request)`
			
			Note
			`pod_memory_request` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs pertinents dans les évènements

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_cpu_limit`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			La limite de processeur définie pour les conteneurs du pod. Si l'un des conteneurs du pod n'a pas de limite de processeur définie, cette métrique n'apparaît pas.
			
			Formule : `sum(container_cpu_limit)`
			
			Note
			`pod_cpu_limit` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informa

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			tions, consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_memory_limit`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			La limite de mémoire définie pour les conteneurs du pod. Si l'un des conteneurs dans le pod n'a pas de limite de mémoire définie, cette métrique n'apparaît pas.
			
			Formule : `sum(container_memory_limit)`
			
			Note
			`pod_cpu_limit` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations,

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

`pod_status_failed`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indique que tous les conteneurs du pod sont résiliés et qu'au moins un conteneur s'est arrêté avec un statut différent de zéro ou a été résilié par le système.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_status_ready`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indique que tous les conteneurs du pod sont prêts, ayant atteint l'état `ContainerReady` .

`pod_status_running`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indique que tous les conteneurs du pod sont en cours d'exécution.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_status_scheduled`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Indique que le pod a été planifié sur un nœud.

`pod_status_unknown`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Indique que l'état du pod ne peut pas être obtenu.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_status_pending`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Indique que le pod a été accepté par le cluster, mais qu'un ou plusieurs conteneurs ne sont pas encore prêts.

`pod_status_succeeded`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Indique que tous les conteneurs du pod ont été correctement résiliés et ne seront pas redémarrés.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_number_of_containers`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Indique le nombre de conteneurs définis dans la spécification du pod.

`pod_number_of_running_containers`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Indique le nombre de conteneurs du pod qui sont actuellement dans l'état `Running`.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_container_status_terminated`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indique le nombre de conteneurs du pod qui sont dans l'état `Terminated` .

`pod_container_status_running`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indique le nombre de conteneurs du pod qui sont dans l'état `Running`.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_container_status_waiting`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Indique le nombre de conteneurs du pod qui sont dans l'état `Waiting`.

`pod_interface_network_rx_dropped`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
		
		`Namespace` , `ClusterName` , `Service`
		
		`ClusterName` , `Namespace` , `PodName`, `FullPodName`
			Le nombre de paquets qui ont été reçus et ensuite abandonnés par une interface réseau pour le pod.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`pod_interface_network_tx_dropped`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
		
		`Namespace`, `ClusterName`, `Service`
		
		`ClusterName`, `Namespace`, `PodName`, `FullPodName`
			Le nombre de paquets qui devaient être transmis, mais qui ont été abandonnés pour le pod.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`container_cpu_utilization`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Pourcentage d'unités de processeur utilisées par le conteneur.
			
			Formule : `container_cpu_usage_total / node_cpu_limit`
			
			Note
			`container_cpu_utilization` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations, consultez

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`container_cpu_utilization_over_container_limit`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Pourcentage d'unités de processeur en cours d'utilisation par le conteneur par rapport à la limite du conteneur. Si le conteneur n'a pas de limite de processeur définie, cette métrique n'apparaît pas.
			
			Formule : `container_cpu_usage_total / container_cpu_limit`
			
			Note
			`container_cpu_utilization_over_container_limit` n'est pas indiqué directement sous forme de métrique, mais constitue

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`container_memory_utilization`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Le pourcentage d'unités de mémoire en cours d'utilisation par le conteneur.
			
			Formule : `container_memory_working_set / node_memory_limit`
			
			Note
			`container_memory_utilization` n'est pas indiqué directement sous forme de métrique, mais constitue un champ dans les événements du journal des performances. Pour plus d'informations,

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`container_memory_utilization_over_container_limit`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Le pourcentage d'unités de mémoire en cours d'utilisation par le conteneur par rapport à la limite du conteneur. Si le conteneur n'a pas de limite de mémoire définie, cette métrique n'apparaît pas.
			
			Formule : `container_memory_working_set / container_memory_limit`
			
			Note
			`container_memory_utilization_over_container_limit` n'est pas indiqué directement sous forme de métrique, mais constitue

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description
			un champ dans les événements du journal des performances. Pour plus d'informations, consultez Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`container_memory_failures_total`
 	
Cette métrique est disponible uniquement avec Container Insights avec une observabilité améliorée pour Amazon EKS. Il n'est pas disponible sous Windows.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName`
		
		`PodName`, `Namespace` , `ClusterName` , `ContainerName` , `FullPodName`
			Le nombre d'échecs d'allocation de mémoire rencontrés par le conteneur.

`pod_number_of_container_restarts`	PodName, `Namespace` , `ClusterName`		Nombre total de redémarrages de conteneur dans un pod.
`service_number_of_running_pods`	Service, `Namespace` , `ClusterName` `ClusterName`		Nombre de blocs exécutant le ou les services du cluster.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`replicas_desired`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			Le nombre de pods souhaités pour une charge de travail, tel que défini dans la spécification de charge de travail.

`replicas_ready`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace` , `ClusterName`
			Le nombre de pods pour une charge de travail qui ont atteint le statut prêt.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`status_replicas_available`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
			Le nombre de pods disponibles pour une charge de travail. Un pod est disponible lorsqu'il est répond au critère `minReadySeconds` défini dans la spécification de charge de travail.

`status_replicas_unavailable`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`PodName`, `Namespace`, `ClusterName`
			Le nombre de pods indisponibles pour une charge de travail. Un pod est disponible lorsqu'il est répond au critère `minReadySeconds` défini dans la spécification de charge de travail. Les pods ne sont pas disponibles s'ils ne répondent pas à ce critère.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`apiserver_storage_objects`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `resource`
			Le nombre d'objets stockés dans etcd au moment de la dernière vérification.

`apiserver_request_total`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			Le nombre total de demandes d'API adressées au serveur d'API Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`apiserver_request_duration_seconds`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , verb`
			Latence de réponse pour les demandes d'API adressées au serveur d'API Kubernetes.

`apiserver_admission_controller_admission_duration_seconds`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , operation`
			Latence du contrôleur d'admission en secondes. Un contrôleur d'admission est un code qui intercepte les requêtes adressées au serveur d'API Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`rest_client_request_duration_seconds`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `operation`
			Latence de réponse rencontrée par les clients qui appellent le serveur d'API Kubernetes. Cette métrique est expérimentale et peut changer dans les futures versions de Kubernetes.

`rest_client_requests_total`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `code`, `method`
			Le nombre total de demandes d'API adressées au serveur d'API Kubernetes par les clients. Cette métrique est expérimentale et peut changer dans les futures versions de Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`etcd_request_duration_seconds`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , operation`
			Latence de réponse des appels d'API à Etcd. Cette métrique est expérimentale et peut changer dans les futures versions de Kubernetes.

`apiserver_storage_size_bytes`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , endpoint`
			Taille du fichier de base de données de stockage physiquement alloué en octets. Cette métrique est expérimentale et peut changer dans les futures versions de Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`apiserver_longrunning_requests`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , resource`
			Le nombre de requêtes actives de longue durée adressées au serveur d'API Kubernetes.

`apiserver_current_inflight_requests`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , request_kind`
			Le nombre de requêtes en cours de traitement par le serveur d'API Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`apiserver_admission_webhook_admission_duration_seconds`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , name`
			Latence du webhook d'admission en secondes. Les webhooks d'admission sont des rappels HTTP qui reçoivent les requêtes d'admission et en font quelque chose.

`apiserver_admission_step_admission_duration_seconds`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , operation`
			Latence des sous-étapes d'admission en secondes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`apiserver_requested_deprecated_apis`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `group`
			Nombre de requêtes adressées à des API obsolètes sur le serveur d'API Kubernetes.

`apiserver_request_total_5XX`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `code`, `verb`
			Nombre de demandes adressées au serveur d'API Kubernetes qui ont reçu un code de réponse HTTP 5xx.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`apiserver_storage_list_duration_seconds`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `resource`
			Latence de réponse lors de l'établissement de listes d'objets à partir d'Etcd. Cette métrique est expérimentale et peut changer dans les futures versions de Kubernetes.

`apiserver_current_inqueue_requests`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName`, `request_kind`
			Le nombre de requêtes mises en file d'attente par le serveur d'API Kubernetes. Cette métrique est expérimentale et peut changer dans les futures versions de Kubernetes.

Nom de la métrique	Dimensions avec n'importe quelle version de Container Insights	Dimensions supplémentaires avec Container Insights avec observabilité améliorée pour Amazon EKS	Description

`apiserver_flowcontrol_rejected_requests_total`
 	
Cette métrique n'est disponible que si vous utilisez Container Insights avec observabilité améliorée pour Amazon EKS.
		`ClusterName`
		
		`ClusterName , reason`
			Nombre de requêtes rejetées par le sous-système API Priority and Fairness. Cette métrique est expérimentale et peut changer dans les futures versions de Kubernetes.

Métriques du GPU NVIDIA

À partir de la version `1.300034.0` de l' CloudWatch agent, Container Insights, doté d'une observabilité améliorée pour Amazon EKS, collecte par défaut les métriques du GPU NVIDIA à partir des charges de travail EKS. L' CloudWatch agent doit être installé à l'aide de la version complémentaire CloudWatch Observability EKS `v1.3.0-eksbuild.1` ou d'une version ultérieure. Pour plus d'informations, consultez Installez l' CloudWatch agent à l'aide du module complémentaire Amazon CloudWatch Observability EKS. Les métriques du GPU NVIDIA collectées sont répertoriées dans le tableau de cette section.

Pour que Container Insights collecte les métriques du GPU NVIDIA, vous devez remplir les conditions préalables suivantes :

- Vous devez utiliser Container Insights avec une observabilité améliorée pour Amazon EKS, avec la version complémentaire Amazon CloudWatch Observability EKS `v1.3.0-eksbuild.1` ou une version ultérieure.

- Le plug-in de périphérique NVIDIA pour Kubernetes doit être installé dans le cluster.
- Le kit d'outils de conteneurs NVIDIA doit être installé sur les nœuds du cluster. Par exemple, les AMI accélérées optimisées pour Amazon EKS sont créées avec les composants nécessaires.

Vous pouvez choisir de ne pas collecter les métriques du GPU NVIDIA en définissant l'`accelerated_compute_metrics`option dans le fichier de configuration de l' CloudWatch agent Begin sur. `false` Pour plus d'informations et un exemple de configuration de désinscription, consultez(Facultatif) Configuration supplémentaire.

Nom de la métrique	Dimensions	Description

`container_gpu_memory_total`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		Taille totale de la mémoire tampon d'images, en octets, sur le ou les GPU alloués au conteneur.

`container_gpu_memory_used`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `GpuDevice`
		Les octets de mémoire tampon utilisés sur le ou les GPU alloués au conteneur.

Nom de la métrique	Dimensions	Description
container_gpu_memory_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Pourcentage de mémoire tampon d'images utilisé par le ou les GPU alloués au conteneur.
container_gpu_power_draw	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Consommation électrique en watts du ou des GPU alloués au conteneur.

Nom de la métrique	Dimensions	Description
container_gpu_temperature	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Température en degrés Celsius du ou des GPU affectés au conteneur.
container_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Pourcentage d'utilisation du ou des GPU alloués au conteneur.
node_gpu_memory_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Taille totale de la mémoire tampon d'images, en octets, sur le ou les GPU alloués au nœud.

Nom de la métrique	Dimensions	Description
node_gpu_memory_used	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Les octets de mémoire tampon utilisés sur le ou les GPU alloués au nœud.
node_gpu_memory_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Pourcentage de mémoire tampon d'images utilisé sur le ou les GPU alloués au nœud.
node_gpu_power_draw	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Consommation électrique en watts du ou des GPU alloués au nœud.
node_gpu_temperature	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Température en degrés Celsius du ou des GPU alloués au nœud.

Nom de la métrique	Dimensions	Description
node_gpu_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, GpuDevice	Pourcentage d'utilisation du ou des GPU alloués au nœud.
pod_gpu_memory_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Taille totale de la mémoire tampon d'images, en octets, sur le ou les GPU alloués au pod.

Nom de la métrique	Dimensions	Description
pod_gpu_memory_used	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Les octets de mémoire tampon utilisés sur le ou les GPU alloués au pod.
pod_gpu_memory_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Pourcentage de mémoire tampon d'images utilisé par le ou les GPU alloués au pod.

Nom de la métrique	Dimensions	Description
pod_gpu_power_draw	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Consommation électrique en watts du ou des GPU alloués au module.
pod_gpu_temperature	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName . GpuDevice	Température en degrés Celsius du ou des GPU affectés au pod.

Nom de la métrique	Dimensions	Description
pod_gpu_utilization	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , GpuDevice	Pourcentage d'utilisation du ou des GPU alloués au pod.

AWS Métriques neuronales pour AWS Trainium et Inferentia AWS

À partir de la version `1.300036.0` de l' CloudWatch agent, Container Insights with Enhanced Observability for Amazon EKS collecte par défaut les métriques de calcul accéléré à partir des accélérateurs AWS Trainium et AWS Inferentia. L' CloudWatch agent doit être installé à l'aide de la version complémentaire CloudWatch Observability EKS `v1.5.0-eksbuild.1` ou d'une version ultérieure. Pour plus d'informations sur le module complémentaire, consultezInstallez l' CloudWatch agent à l'aide du module complémentaire Amazon CloudWatch Observability EKS. Pour plus d'informations sur AWS Trainium, consultez AWS Trainium. Pour plus d'informations sur AWS Inferentia, voir AWS Inferentia.

Pour que Container Insights collecte des métriques AWS Neuron, vous devez remplir les conditions préalables suivantes :

- Vous devez utiliser Container Insights avec une observabilité améliorée pour Amazon EKS, avec la version complémentaire Amazon CloudWatch Observability EKS `v1.5.0-eksbuild.1` ou une version ultérieure.
- Le pilote Neuron doit être installé sur les nœuds du cluster.
- Le plug-in du périphérique Neuron doit être installé sur le cluster. Par exemple, les AMI accélérées optimisées pour Amazon EKS sont créées avec les composants nécessaires.

Les mesures collectées sont répertoriées dans le tableau de cette section. Les métriques sont collectées pour AWS Trainium, AWS Inferentia et AWS Inferentia2.

L' CloudWatch agent collecte ces métriques à partir du moniteur Neuron et effectue la corrélation des ressources Kubernetes nécessaire pour fournir des métriques au niveau du pod et du conteneur.

Nom de la métrique	Dimensions	Description

`container_neuroncore_utilization`	`ClusterName`
	
	`ClusterName`, `Namespace`, `PodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`, `NeuronDevice`, `NeuronCore`
		NeuronCore utilisation, pendant la période capturée, de l' NeuronCore allocation au conteneur.
		
		Unité : pourcentage

`container_neuroncore_memory_usage_constants`	`ClusterName`
	
	`ClusterName`, `Namespace`, `PodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`
	
	`ClusterName`, `Namespace`, `PodName`, `FullPodName`, `ContainerName`, `NeuronDevice`, `NeuronCore`
		La quantité de mémoire du périphérique utilisée pour les constantes pendant l'entraînement par le NeuronCore qui est allouée au conteneur (ou les poids lors de l'inférence).
		
		Unité : octets

`container_neuroncore_memory`	`ClusterName`	La quantité de mémoire de l'appareil utilisée pour le code exécutable des

Nom de la métrique	Dimensions	Description
_usage_model_code	ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	modèles par le NeuronCore qui est allouée au conteneur. Unité : octets
container_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour le bloc-notes partagé par les NeuronCore modèles et allouée au conteneur. Cette zone de mémoire est réservée aux modèles. Unité : octets

Nom de la métrique	Dimensions	Description
container_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La quantité de mémoire du périphéri que utilisée pour le runtime Neuron par la mémoire NeuronCore allouée au conteneur. Unité : octets
container_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La quantité de mémoire du périphéri que utilisée pour les tenseurs par la quantité NeuronCore allouée au conteneur. Unité : octets

Nom de la métrique	Dimensions	Description
container_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice , NeuronCore	La quantité totale de mémoire utilisée par la mémoire NeuronCore allouée au conteneur. Unité : octets
container_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , NeuronDevice	Le nombre d'événements ECC corrigés et non corrigés pour la SRAM sur puce et la mémoire du périphérique Neuron sur le nœud. Unité : nombre

Nom de la métrique	Dimensions	Description
pod_neuroncore_utilization	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	L' NeuronCore utilisation pendant la période capturée de l' NeuronCore allocation au pod. Unité : pourcentage
pod_neuroncore_memory_usage_constants	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour les constantes pendant l'entraînement par le NeuronCore module (ou les poids lors de l'inférence). Unité : octets

Nom de la métrique	Dimensions	Description
pod_neuroncore_memory_usage_model_code	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour le code exécutable des modèles par le NeuronCore qui est allouée au pod. Unité : octets
pod_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour le bloc-notes partagée entre les modèles par le module NeuronCore et allouée au pod. Cette zone de mémoire est réservée aux modèles. Unité : octets

Nom de la métrique	Dimensions	Description
pod_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour le runtime Neuron par la mémoire NeuronCore allouée au pod. Unité : octets
pod_neuroncore_memory_usage_tensors	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour les tenseurs par la mémoire NeuronCore allouée au pod. Unité : octets

Nom de la métrique	Dimensions	Description
pod_neuroncore_memory_usage_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice , NeuronCore	La quantité totale de mémoire utilisée par la mémoire NeuronCore allouée au pod. Unité : octets
pod_neurondevice_hw_ecc_events_total	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , NeuronDevice	Le nombre d'événements ECC corrigés et non corrigés pour la SRAM intégrée et la mémoire du périphérique Neuron allouée à un pod. Unité : octets

Nom de la métrique	Dimensions	Description
node_neuroncore_utilization	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	L' NeuronCore utilisation pendant la période capturée du NeuronCore montant alloué au nœud. Unité : pourcentage
node_neuroncore_memory_usage_constants	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantité de mémoire du périphérique utilisée pour les constantes pendant l'entraînement par le NeuronCore nœud (ou les poids lors de l'inférence). Unité : octets
node_neuroncore_memory_usage_model_code	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour le code exécutable des modèles par le NeuronCore qui est allouée au nœud. Unité : octets
node_neuroncore_memory_usage_model_shared_scratchpad	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour le bloc-notes partagé entre NeuronCore les modèles par le nœud. Il s'agit d'une zone de mémoire réservée aux modèles. Unité : octets

Nom de la métrique	Dimensions	Description
node_neuroncore_memory_usage_runtime_memory	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantité de mémoire de l'appareil utilisée pour le runtime Neuron par le périphérique NeuronCore qui est allouée au nœud. Unité : octets
node_neuroncore_memory_usage_tensors	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantité de mémoire du périphérique utilisée pour les tenseurs par le NeuronCore qui est allouée au nœud. Unité : octets
node_neuroncore_memory_usage_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceType , InstanceId , NodeName, NeuronDevice , NeuronCore	La quantité totale de mémoire utilisée par le nœud NeuronCore qui est allouée au nœud. Unité : octets
node_neuron_execution_errors_total	ClusterName ClusterName , InstanceId , NodeName	Nombre total d'erreurs d'exécution sur le nœud. Ceci est calculé par l' CloudWatch agent en agrégeant les erreurs des types suivants :generic,numerical ,transient , modelruntime, et hardware Unité : nombre

Nom de la métrique	Dimensions	Description
node_neuron_device_runtime_memory_used_bytes	ClusterName ClusterName , InstanceId , NodeName	L'utilisation totale de la mémoire du dispositif Neuron en octets sur le nœud. Unité : octets
node_neuron_execution_latency	ClusterName ClusterName , InstanceId , NodeName	En secondes, latence d'une exécution sur le nœud, telle que mesurée par le temps d'exécution du Neuron. Unité : secondes
node_neuron_device_hw_ecc_events_total	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , NodeName, NeuronDevice	Le nombre d'événements ECC corrigés et non corrigés pour la SRAM sur puce et la mémoire du périphérique Neuron sur le nœud. Unité : nombre

AWS Métriques de l'Elastic Fabric Adapter (EFA)

À partir de la version `1.300037.0` de l' CloudWatch agent, Container Insights, doté d'une observabilité améliorée pour Amazon EKS, collecte les métriques AWS Elastic Fabric Adapter (EFA) à partir de clusters Amazon EKS sur des instances Linux. L' CloudWatch agent doit être installé à l'aide de la version complémentaire CloudWatch Observability EKS `v1.5.2-eksbuild.1` ou d'une version ultérieure. Pour plus d'informations sur le module complémentaire, consultezInstallez l' CloudWatch agent à l'aide du module complémentaire Amazon CloudWatch Observability EKS. Pour plus d'informations sur AWS Elastic Fabric Adapter, consultez Elastic Fabric Adapter.

Pour que Container Insights collecte les métriques des adaptateurs AWS Elastic Fabric, vous devez remplir les conditions préalables suivantes :

- Vous devez utiliser Container Insights avec une observabilité améliorée pour Amazon EKS, avec la version complémentaire Amazon CloudWatch Observability EKS `v1.5.2-eksbuild.1` ou une version ultérieure.
- Le plug-in de périphérique EFA doit être installé sur le cluster. Pour plus d'informations, voir aws-efa-k8 s-device-plugin sur GitHub.

Les métriques collectées sont répertoriées dans le tableau suivant.

Nom de la métrique	Dimensions	Description

`container_efa_rx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Le nombre d'octets par seconde reçus par le ou les périphériques EFA alloués au conteneur.
		
		Unité : octets/seconde

`container_efa_tx_bytes`	`ClusterName`
	
	`ClusterName` , `Namespace` , `PodName`, `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName`
	
	`ClusterName` , `Namespace` , `PodName`, `FullPodName` , `ContainerName` , `EfaDevice`
		Le nombre d'octets par seconde transmis par le ou les périphériques EFA alloués au conteneur.
		
		Unité : octets/seconde

Nom de la métrique	Dimensions	Description
container_efa_rx_dropped	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Le nombre de paquets reçus puis déposés par le ou les périphériques EFA alloués au conteneur. Unité : compte/seconde
container_efa_rdma_read_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Nombre d'octets par seconde reçus à l'aide d'opérations de lecture à accès direct à distance à la mémoire par le ou les dispositifs EFA affectés au conteneur. Unité : octets/seconde

Nom de la métrique	Dimensions	Description
container_efa_rdma_write_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Nombre d'octets par seconde transmis à l'aide d'opérations de lecture à accès direct à distance à la mémoire par le ou les dispositifs EFA affectés au conteneur. Unité : octets/seconde
container_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace , PodName, ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName ClusterName , Namespace , PodName, FullPodName , ContainerName , EfaDevice	Nombre d'octets par seconde reçus lors des opérations d'écriture à accès direct à la mémoire à distance par le ou les périphériques EFA alloués au conteneur. Unité : octets/seconde

Nom de la métrique	Dimensions	Description
pod_efa_rx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Le nombre d'octets par seconde reçus par le ou les appareils EFA alloués au pod. Unité : octets/seconde
pod_efa_tx_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Nombre d'octets par seconde transmis par le ou les périphériques EFA alloués au pod. Unité : octets/seconde

Nom de la métrique	Dimensions	Description
pod_efa_rx_dropped	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Le nombre de paquets reçus puis déposés par le ou les périphériques EFA alloués au pod. Unité : compte/seconde
pod_efa_rdma_read_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Nombre d'octets par seconde reçus à l'aide des opérations de lecture à accès direct à distance à la mémoire par le ou les périphériques EFA affectés au pod. Unité : octets/seconde

Nom de la métrique	Dimensions	Description
pod_efa_rdma_write_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Nombre d'octets par seconde transmis à l'aide d'opérations de lecture à accès direct à distance à la mémoire par le ou les périphériques EFA affectés au pod. Unité : octets/seconde
pod_efa_rdma_write_recv_bytes	ClusterName ClusterName , Namespace ClusterName , Namespace , Service ClusterName , Namespace , PodName ClusterName , Namespace , PodName, FullPodName ClusterName , Namespace , PodName, FullPodName , EfaDevice	Nombre d'octets par seconde reçus lors des opérations d'écriture à accès direct à la mémoire à distance par le ou les périphériques EFA affectés au pod. Unité : octets/seconde

Nom de la métrique	Dimensions	Description
node_efa_rx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Nombre d'octets par seconde reçus par le ou les périphériques EFA alloués au nœud. Unité : octets/seconde
node_efa_tx_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Nombre d'octets par seconde transmis par le ou les périphériques EFA alloués au nœud. Unité : octets/seconde
node_efa_rx_dropped	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Le nombre de paquets reçus puis abandonnés par le ou les périphéri ques EFA alloués au nœud. Unité : compte/seconde
node_efa_rdma_read_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Nombre d'octets par seconde reçus à l'aide d'opérations de lecture à accès direct à distance à la mémoire par le ou les dispositifs EFA alloués au nœud. Unité : octets/seconde

Nom de la métrique	Dimensions	Description
pod_efa_rdma_write_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Nombre d'octets par seconde transmis à l'aide d'opérations de lecture à accès direct à distance à la mémoire par le ou les périphériques EFA affectés au pod. Unité : octets/seconde
node_efa_rdma_write_recv_bytes	ClusterName ClusterName , InstanceId , NodeName ClusterName , InstanceId , InstanceType , NodeName, EfaDevice	Nombre d'octets par seconde reçus lors des opérations d'écriture à accès direct à la mémoire à distance par le ou les dispositifs EFA alloués au nœud. Unité : octets/seconde

Référence des journaux de performances Container Insights

Cette section inclut des informations de référence sur la façon dont Container Insights utilise les évènements de journaux de performances pour collecter les métriques. Lorsque vous déployez Container Insights, il crée automatiquement un groupe de journaux pour les événements du journal des performances. Il n'est pas nécessaire de créer ce groupe de journaux vous-même.

Rubriques

- Évènements de journaux de performances de Container Insights pour Amazon ECS
- Évènements du journaal de performances de Container Insights pour Amazon EKS et Kubernetes
- Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes

Évènements de journaux de performances de Container Insights pour Amazon ECS

Voici des exemples d'évènements de journaux de performance que Container Insights collecte depuis Amazon ECS.

Ces journaux se trouvent dans CloudWatch Logs, dans un groupe de journaux nommé `/aws/ecs/containerinsights/`*`CLUSTER_NAME`*`/performance`. Au sein de ce groupe de journaux, chaque instance de conteneur aura un flux de journaux nommé `AgentTelemetry-`*`CONTAINER_INSTANCE_ID`*.

Vous pouvez interroger ces journaux à l'aide de requêtes, par exemple `{ $.Type = "Container" }` pour afficher tous les événements des journaux de conteneurs.

Type : Conteneur


```
{
 "Version":"0",
 "Type":"Container",
 "ContainerName":"sleep",
 "TaskId":"7ac4dfba69214411b4783a3b8189c9ba",
 "TaskDefinitionFamily":"sleep360",
 "TaskDefinitionRevision":"1",
 "ContainerInstanceId":"0d7650e6dec34c1a9200f72098071e8f",
 "EC2InstanceId":"i-0c470579dbcdbd2f3",
 "ClusterName":"MyCluster",
 "Image":"busybox",
 "ContainerKnownStatus":"RUNNING",
 "Timestamp":1623963900000,
 "CpuUtilized":0.0,
 "CpuReserved":10.0,
 "MemoryUtilized":0,
 "MemoryReserved":10,
 "StorageReadBytes":0,
 "StorageWriteBytes":0,
 "NetworkRxBytes":0,
 "NetworkRxDropped":0,
 "NetworkRxErrors":0,
 "NetworkRxPackets":14,
 "NetworkTxBytes":0,
 "NetworkTxDropped":0,
 "NetworkTxErrors":0,
 "NetworkTxPackets":0
```

```
}
```

Type : Tâche

```
{
    "Version": "0",
    "Type": "Task",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "TaskDefinitionFamily": "sleep360",
    "TaskDefinitionRevision": "1",
    "ContainerInstanceId": "0d7650e6dec34c1a9200f72098071e8f",
    "EC2InstanceId": "i-0c470579dbcdbd2f3",
    "ClusterName": "MyCluster",
    "AccountID": "637146863587",
    "Region": "us-west-2",
    "AvailabilityZone": "us-west-2b",
    "KnownStatus": "RUNNING",
    "LaunchType": "EC2",
    "PullStartedAt": 1623963608201,
    "PullStoppedAt": 1623963610065,
    "CreatedAt": 1623963607094,
    "StartedAt": 1623963610382,
    "Timestamp": 1623963900000,
    "CpuUtilized": 0.0,
    "CpuReserved": 10.0,
    "MemoryUtilized": 0,
    "MemoryReserved": 10,
    "StorageReadBytes": 0,
    "StorageWriteBytes": 0,
    "NetworkRxBytes": 0,
    "NetworkRxDropped": 0,
    "NetworkRxErrors": 0,
    "NetworkRxPackets": 14,
    "NetworkTxBytes": 0,
    "NetworkTxDropped": 0,
    "NetworkTxErrors": 0,
    "NetworkTxPackets": 0,
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
```

```
                {
                    "Name": "CpuUtilized",
                    "Unit": "None"
                },
                {
                    "Name": "CpuReserved",
                    "Unit": "None"
                },
                {
                    "Name": "MemoryUtilized",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "MemoryReserved",
                    "Unit": "Megabytes"
                },
                {
                    "Name": "StorageReadBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "StorageWriteBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkRxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "NetworkTxBytes",
                    "Unit": "Bytes/Second"
                },
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
                [
```

```
                    "ClusterName",
                    "TaskDefinitionFamily"
                ]
            ]
        }
    ]
}
```

Type : Service

```
{
    "Version": "0",
    "Type": "Service",
    "ServiceName": "myCIService",
    "ClusterName": "myCICluster",
    "Timestamp": 1561586460000,
    "DesiredTaskCount": 2,
    "RunningTaskCount": 2,
    "PendingTaskCount": 0,
    "DeploymentCount": 1,
    "TaskSetCount": 0,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "DesiredTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "RunningTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "PendingTaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "DeploymentCount",
                    "Unit": "Count"
                },
                {
                    "Name": "TaskSetCount",
```

```
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Type : Volume

```
{
    "Version": "0",
    "Type": "Volume",
    "TaskDefinitionFamily": "myCITaskDef",
    "TaskId": "7ac4dfba69214411b4783a3b8189c9ba",
    "ClusterName": "myCICluster",
    "ServiceName": "myCIService",
    "VolumeId": "vol-1233436545ff708cb",
    "InstanceId": "i-0c470579dbcdbd2f3",
    "LaunchType": "EC2",
    "VolumeName": "MyVolumeName",
    "EBSFilesystemUtilized": 10,
    "EBSFilesystemSize": 20,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "EBSFilesystemSize",
                    "Unit": "Gigabytes"
                },
                {
                    "Name": "EBSFilesystemUtilzed",
                    "Unit": "Gigabytes"
                }
            ],
            "Dimensions": [
                ["ClusterName"],
```

```
                [
                    "VolumeName",
                    "TaskDefinitionFamily",
                    "ClusterName"
                ],
                [
                    "ServiceName",
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Type : Cluster

```
{
    "Version": "0",
    "Type": "Cluster",
    "ClusterName": "myCICluster",
    "Timestamp": 1561587300000,
    "TaskCount": 5,
    "ContainerInstanceCount": 5,
    "ServiceCount": 2,
    "CloudWatchMetrics": [
        {
            "Namespace": "ECS/ContainerInsights",
            "Metrics": [
                {
                    "Name": "TaskCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ContainerInstanceCount",
                    "Unit": "Count"
                },
                {
                    "Name": "ServiceCount",
                    "Unit": "Count"
                }
            ],
            "Dimensions": [
                [
```

```
                    "ClusterName"
                ]
            ]
        }
    ]
}
```

Évènements du journaal de performances de Container Insights pour Amazon EKS et Kubernetes

Voici des exemples d'évènements de journaux de performances que Container Insights collecte à partir des clusters Amazon EKS et Kubernetes.

Type : Node

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
```

```
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_containers"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "node_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "node_network_total_bytes"
        },
        {
          "Unit": "Percent",
          "Name": "node_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "node_memory_reserved_capacity"
        },
        {
          "Unit": "Count",
          "Name": "node_number_of_running_pods"
        },
        {
          "Unit": "Count",
```

```
          "Name": "node_number_of_running_containers"
        },
        {
          "Name": "node_cpu_usage_total"
        },
        {
          "Name": "node_cpu_limit"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_working_set"
        },
        {
          "Unit": "Bytes",
          "Name": "node_memory_limit"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "/proc",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567096682364",
  "Type": "Node",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_cpu_limit": 4000,
  "node_cpu_request": 1130,
  "node_cpu_reserved_capacity": 28.249999999999996,
```

```
  "node_cpu_usage_system": 33.794636630852764,
  "node_cpu_usage_total": 136.47852169244098,
  "node_cpu_usage_user": 71.67075111567326,
  "node_cpu_utilization": 3.4119630423110245,
  "node_memory_cache": 3103297536,
  "node_memory_failcnt": 0,
  "node_memory_hierarchical_pgfault": 0,
  "node_memory_hierarchical_pgmajfault": 0,
  "node_memory_limit": 16624865280,
  "node_memory_mapped_file": 406646784,
  "node_memory_max_usage": 4230746112,
  "node_memory_pgfault": 0,
  "node_memory_pgmajfault": 0,
  "node_memory_request": 1115684864,
  "node_memory_reserved_capacity": 6.7109407818311055,
  "node_memory_rss": 798146560,
  "node_memory_swap": 0,
  "node_memory_usage": 3901444096,
  "node_memory_utilization": 6.601302600149552,
  "node_memory_working_set": 1097457664,
  "node_network_rx_bytes": 35918.392817386324,
  "node_network_rx_dropped": 0,
  "node_network_rx_errors": 0,
  "node_network_rx_packets": 157.67565245448117,
  "node_network_total_bytes": 68264.20276554905,
  "node_network_tx_bytes": 32345.80994816272,
  "node_network_tx_dropped": 0,
  "node_network_tx_errors": 0,
  "node_network_tx_packets": 154.21455923431654,
  "node_number_of_running_containers": 16,
  "node_number_of_running_pods": 13
}
```


Type : NodeFS


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
```

```
          "Name": "node_filesystem_utilization"
        }
      ],
      "Dimensions": [
        [
          "NodeName",
          "InstanceId",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097939726",
  "Type": "NodeFS",
  "Version": "0",
  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_filesystem_available": 17298395136,
  "node_filesystem_capacity": 21462233088,
  "node_filesystem_inodes": 10484720,
  "node_filesystem_inodes_free": 10367158,
  "node_filesystem_usage": 4163837952,
  "node_filesystem_utilization": 19.400767547940255
}
```


Type : NodeDisk IO

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor"
  ],
  "Timestamp": "1567096928131",
  "Type": "NodeDiskIO",
  "Version": "0",
  "device": "/dev/nvme0n1",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_diskio_io_service_bytes_async": 9750.505814277016,
  "node_diskio_io_service_bytes_read": 0,
  "node_diskio_io_service_bytes_sync": 230.6174506688036,
  "node_diskio_io_service_bytes_total": 9981.123264945818,
  "node_diskio_io_service_bytes_write": 9981.123264945818,
  "node_diskio_io_serviced_async": 1.153087253344018,
  "node_diskio_io_serviced_read": 0,
  "node_diskio_io_serviced_sync": 0.03603397666700056,
  "node_diskio_io_serviced_total": 1.1891212300110185,
  "node_diskio_io_serviced_write": 1.1891212300110185
}
```


Type : NodeNet


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
```

```
  "Timestamp": "1567096928131",
  "Type": "NodeNet",
  "Version": "0",
  "interface": "eni972f6bfa9a0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal"
  },
  "node_interface_network_rx_bytes": 3163.008420864309,
  "node_interface_network_rx_dropped": 0,
  "node_interface_network_rx_errors": 0,
  "node_interface_network_rx_packets": 16.575629266820258,
  "node_interface_network_total_bytes": 3518.3935157426017,
  "node_interface_network_tx_bytes": 355.385094878293,
  "node_interface_network_tx_dropped": 0,
  "node_interface_network_tx_errors": 0,
  "node_interface_network_tx_packets": 3.9997714100370625
}
```


Type : Pod


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_utilization"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_rx_bytes"
        },
        {
          "Unit": "Bytes/Second",
          "Name": "pod_network_tx_bytes"
        },
        {
```

```
          "Unit": "Percent",
          "Name": "pod_cpu_utilization_over_pod_limit"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_utilization_over_pod_limit"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ],
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Percent",
          "Name": "pod_cpu_reserved_capacity"
        },
        {
          "Unit": "Percent",
          "Name": "pod_memory_reserved_capacity"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
```

```
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    },
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "pod_number_of_container_restarts"
        }
      ],
      "Dimensions": [
        [
          "PodName",
          "Namespace",
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "Pod",
  "Version": "0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
```

```
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_cpu_limit": 200,
  "pod_cpu_request": 200,
  "pod_cpu_reserved_capacity": 5,
  "pod_cpu_usage_system": 1.4504841104992765,
  "pod_cpu_usage_total": 5.817016867430125,
  "pod_cpu_usage_user": 1.1281543081661038,
  "pod_cpu_utilization": 0.14542542168575312,
  "pod_cpu_utilization_over_pod_limit": 2.9085084337150624,
  "pod_memory_cache": 8192,
  "pod_memory_failcnt": 0,
  "pod_memory_hierarchical_pgfault": 0,
  "pod_memory_hierarchical_pgmajfault": 0,
  "pod_memory_limit": 104857600,
  "pod_memory_mapped_file": 0,
  "pod_memory_max_usage": 25268224,
  "pod_memory_pgfault": 0,
  "pod_memory_pgmajfault": 0,
  "pod_memory_request": 104857600,
  "pod_memory_reserved_capacity": 0.6307275170893897,
  "pod_memory_rss": 22777856,
  "pod_memory_swap": 0,
  "pod_memory_usage": 25141248,
  "pod_memory_utilization": 0.10988455961791709,
  "pod_memory_utilization_over_pod_limit": 17.421875,
  "pod_memory_working_set": 18268160,
  "pod_network_rx_bytes": 9880.697124714186,
  "pod_network_rx_dropped": 0,
  "pod_network_rx_errors": 0,
  "pod_network_rx_packets": 107.80005532263283,
  "pod_network_total_bytes": 10158.829201483635,
  "pod_network_tx_bytes": 278.13207676944796,
```

```
  "pod_network_tx_dropped": 0,
  "pod_network_tx_errors": 0,
  "pod_network_tx_packets": 1.146027574644318,
  "pod_number_of_container_restarts": 0,
  "pod_number_of_containers": 1,
  "pod_number_of_running_containers": 1,
  "pod_status": "Running"
}
```

Type : PodNet

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097351092",
  "Type": "PodNet",
  "Version": "0",
  "interface": "eth0",
  "kubernetes": {
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
```

```
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "pod_interface_network_rx_bytes": 9880.697124714186,
  "pod_interface_network_rx_dropped": 0,
  "pod_interface_network_rx_errors": 0,
  "pod_interface_network_rx_packets": 107.80005532263283,
  "pod_interface_network_total_bytes": 10158.829201483635,
  "pod_interface_network_tx_bytes": 278.13207676944796,
  "pod_interface_network_tx_dropped": 0,
  "pod_interface_network_tx_errors": 0,
  "pod_interface_network_tx_packets": 1.146027574644318
}
```

Type : Conteneur

```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-NodeGroup-
sample",
  "ClusterName": "myCICluster",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "pod",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "Container",
  "Version": "0",
  "container_cpu_limit": 200,
  "container_cpu_request": 200,
  "container_cpu_usage_system": 1.87958283771964,
  "container_cpu_usage_total": 6.159993652997942,
  "container_cpu_usage_user": 1.6707403001952357,
  "container_cpu_utilization": 0.15399984132494854,
  "container_memory_cache": 8192,
  "container_memory_failcnt": 0,
  "container_memory_hierarchical_pgfault": 0,
```

```
  "container_memory_hierarchical_pgmajfault": 0,
  "container_memory_limit": 104857600,
  "container_memory_mapped_file": 0,
  "container_memory_max_usage": 24580096,
  "container_memory_pgfault": 0,
  "container_memory_pgmajfault": 0,
  "container_memory_request": 104857600,
  "container_memory_rss": 22736896,
  "container_memory_swap": 0,
  "container_memory_usage": 24453120,
  "container_memory_utilization": 0.10574541028701798,
  "container_memory_working_set": 17580032,
  "container_status": "Running",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  },
  "number_of_container_restarts": 0
}
```


Type : ContainerFS


```
{
  "AutoScalingGroupName": "eksctl-myCICluster-nodegroup-standard-workers-
NodeGroup-1174PV2WHZAYU",
```

```
  "ClusterName": "myCICluster",
  "EBSVolumeId": "aws://us-west-2b/vol-0a53108976d4a2fda",
  "InstanceId": "i-1234567890123456",
  "InstanceType": "t3.xlarge",
  "Namespace": "amazon-cloudwatch",
  "NodeName": "ip-192-0-2-0.us-west-2.compute.internal",
  "PodName": "cloudwatch-agent-statsd",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "cadvisor",
    "calculated"
  ],
  "Timestamp": "1567097399912",
  "Type": "ContainerFS",
  "Version": "0",

  "device": "/dev/nvme0n1p1",
  "fstype": "vfs",
  "kubernetes": {
    "container_name": "cloudwatch-agent",
    "docker": {
      "container_id":
 "8967b6b37da239dfad197c9fdea3e5dfd35a8a759ec86e2e4c3f7b401e232706"
    },
    "host": "ip-192-168-75-26.us-west-2.compute.internal",
    "labels": {
      "app": "cloudwatch-agent-statsd",
      "pod-template-hash": "df44f855f"
    },
    "namespace_name": "amazon-cloudwatch",
    "pod_id": "2f4ff5ac-c813-11e9-a31d-06e9dde32928",
    "pod_name": "cloudwatch-agent-statsd-df44f855f-ts4q2",
    "pod_owners": [
      {
        "owner_kind": "Deployment",
        "owner_name": "cloudwatch-agent-statsd"
      }
    ],
    "service_name": "cloudwatch-agent-statsd"
  }
}
```


Type : Cluster

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "cluster_node_count"
        },
        {
          "Unit": "Count",
          "Name": "cluster_failed_node_count"
        }
      ],
      "Dimensions": [
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "Cluster",
  "Version": "0",
  "cluster_failed_node_count": 0,
  "cluster_node_count": 3
}
```

Type : ClusterService

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "service_number_of_running_pods"
        }
      ],
```

```
      "Dimensions": [
        [
          "Service",
          "Namespace",
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Service": "cloudwatch-agent-statsd",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097534160",
  "Type": "ClusterService",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch",
    "service_name": "cloudwatch-agent-statsd"
  },
  "service_number_of_running_pods": 1
}
```

Type : ClusterNamespace

```
{
  "CloudWatchMetrics": [
    {
      "Metrics": [
        {
          "Unit": "Count",
          "Name": "namespace_number_of_running_pods"
        }
      ],
      "Dimensions": [
        [
          "Namespace",
```

```
          "ClusterName"
        ],
        [
          "ClusterName"
        ]
      ],
      "Namespace": "ContainerInsights"
    }
  ],
  "ClusterName": "myCICluster",
  "Namespace": "amazon-cloudwatch",
  "Sources": [
    "apiserver"
  ],
  "Timestamp": "1567097594160",
  "Type": "ClusterNamespace",
  "Version": "0",
  "kubernetes": {
    "namespace_name": "amazon-cloudwatch"
  },
  "namespace_number_of_running_pods": 7
}
```

Champs pertinents dans les évènements de journaux de performance pour Amazon EKS et Kubernetes

Pour Amazon EKS et Kubernetes, l' CloudWatch agent conteneurisé émet des données sous forme d'événements du journal des performances. Cela permet d' CloudWatch ingérer et de stocker des données à haute cardinalité. CloudWatch utilise les données contenues dans les événements du journal des performances pour créer CloudWatch des métriques agrégées au niveau du cluster, du nœud et du pod sans perdre de détails granulaires.

Le tableau suivant répertorie les champs de ces événements du journal de performances qui sont pertinents pour la collecte de données de métriques Container Insights. Vous pouvez utiliser CloudWatch Logs Insights pour rechercher l'un de ces champs afin de collecter des données ou d'étudier des problèmes. Pour plus d'informations, voir Analyser les données des CloudWatch journaux avec Logs Insights.

Type	Champ de journal	Source	Formule ou notes
Pod	`pod_cpu_utilization`	Calculé	Formule : `pod_cpu_usage_total / node_cpu_limit`
Pod	`pod_cpu_usage_total` `pod_cpu_usage_total` est signalé en Millicores.	cadvisor	
Pod	`pod_cpu_limit`	Calculé	Formule : `sum(container_cpu_limit)` `sum(container_cpu_limit)` inclut des pods déjà terminés. Si l'un des conteneurs dans le pod n'a pas de limite d'UC définie, ce champ n'apparaît pas dans l'événement de journal. Inclut

Type	Champ de journal	Source	Formule ou notes
			les conteneurs init.
Pod	`pod_cpu_request`	Calculé	Formule : `sum(container_cpu_request)` L'élément `container_cpu_request` n'est pas forcément défini. Seuls ceux définis sont inclus dans la somme.
Pod	`pod_cpu_utilization_over_pod_limit`	Calculé	Formule : `pod_cpu_usage_total / pod_cpu_limit`
Pod	`pod_cpu_reserved_capacity`	Calculé	Formule : `pod_cpu_request / node_cpu_limit`

Type	Champ de journal	Source	Formule ou notes
Pod	`pod_memory_utilization`	Calculé	Formule : `pod_memory_working_set / node_memory_limit` Il s'agit du pourcentage d'utilisation de la mémoire du pod par rapport à la limitation de la mémoire du nœud.
Pod	`pod_memory_working_set`	cadvisor	

Type	Champ de journal	Source	Formule ou notes
Pod	`pod_memory_limit`	Calculé	Formule : `sum(container_memory_limit)` Si l'un des conteneurs dans le pod n'a pas de limite de mémoire définie, ce champ n'apparaît pas dans l'événement de journal. Inclut les conteneurs init.

Type	Champ de journal	Source	Formule ou notes
Pod	`pod_memory_request`	Calculé	Formule : `sum(container_memory_request)` L'élément `container_memory_request` n'est pas forcément défini. Seuls ceux définis sont inclus dans la somme.

Type	Champ de journal	Source	Formule ou notes
Pod	`pod_memory_utilization_over_pod_limit`	Calculé	Formule : `pod_memory_working_set / pod_memory_limit` Si l'un des conteneurs dans le pod n'a pas de limite de mémoire définie, ce champ n'apparaît pas dans l'événement de journal. Inclut les conteneurs init.
Pod	`pod_memory_reserved_capacity`	Calculé	Formule : `pod_memory_request / node_memory_limit`

Type	Champ de journal	Source	Formule ou notes
Pod	`pod_network_tx_bytes`	Calculé	Formule : `sum(pod_interface_network_tx_bytes)` Ces données sont disponibles pour toutes les interfaces réseau (par pod). L'CloudWatch agent calcule le total et ajoute des règles d'extraction métriques.
Pod	`pod_network_rx_bytes`	Calculé	Formule : `sum(pod_interface_network_rx_bytes)`
Pod	`pod_network_total_bytes`	Calculé	Formule : `pod_network_rx_bytes + pod_network_tx_bytes`

Type	Champ de journal	Source	Formule ou notes
PodNet	`pod_interface_network_rx_bytes`	cadvisor	Ces données correspondent au nombre d'octets rx réseau par seconde d'une interface réseau de pod.
PodNet	`pod_interface_network_tx_bytes`	cadvisor	Ces données correspondent au nombre d'octets tx réseau par seconde d'une interface réseau de pod.
Conteneur	`container_cpu_usage_total`	cadvisor	
Conteneur	`container_cpu_limit`	cadvisor	Cet élément n'est pas forcément défini. S'il n'est pas défini, il ne sera pas transmis.

Type	Champ de journal	Source	Formule ou notes
Conteneur	container_cpu_request	cadvisor	Cet élément n'est pas forcément défini. S'il n'est pas défini, il ne sera pas transmis.
Conteneur	container_memory_working_set	cadvisor	
Conteneur	container_memory_limit	pod	Cet élément n'est pas forcément défini. S'il n'est pas défini, il ne sera pas transmis.
Conteneur	container_memory_request	pod	Cet élément n'est pas forcément défini. S'il n'est pas défini, il ne sera pas transmis.
Nœud	node_cpu_utilization	Calculé	Formule : node_cpu_usage_total / node_cpu_limit
Nœud	node_cpu_usage_total	cadvisor	

Type	Champ de journal	Source	Formule ou notes
Nœud	`node_cpu_limit`	/proc	
Nœud	`node_cpu_request`	Calculé	Formule : `sum(pod_cpu_request)` Pour les tâches cron, `node_cpu_request` inclut également les requêtes provenant de pods terminés. Cela peut entraîner une valeur élevée pour `node_cpu_reserved_capacity` .
Nœud	`node_cpu_reserved_capacity`	Calculé	Formule : `node_cpu_request / node_cpu_limit`

Type	Champ de journal	Source	Formule ou notes
Nœud	node_memory_utilization	Calculé	Formule : node_memory_working_set / node_memory_limit
Nœud	node_memory_working_set	cadvisor	
Nœud	node_memory_limit	/proc	
Nœud	node_memory_request	Calculé	Formule : sum(pod_memory_request)
Nœud	node_memory_reserved_capacity	Calculé	Formule : node_memory_request / node_memory_limit
Nœud	node_network_rx_bytes	Calculé	Formule : sum(node_interface_network_rx_bytes)
Nœud	node_network_tx_bytes	Calculé	Formule : sum(node_interface_network_tx_bytes)

Type	Champ de journal	Source	Formule ou notes
Nœud	`node_network_total_bytes`	Calculé	Formule : `node_network_rx_bytes + node_network_tx_bytes`
Nœud	`node_number_of_running_pods`	Liste de pods	
Nœud	`node_number_of_running_containers`	Liste de pods	
NodeNet	`node_interface_network_rx_bytes`	cadvisor	Ces données correspondent au nombre d'octets rx réseau par seconde d'une interface réseau de composant master.
NodeNet	`node_interface_network_tx_bytes`	cadvisor	Ces données correspondent au nombre d'octets tx réseau par seconde d'une interface réseau de composant master.

Type	Champ de journal	Source	Formule ou notes
NodeFS	`node_filesystem_capacity`	cadvisor	
NodeFS	`node_filesystem_usage`	cadvisor	
NodeFS	`node_filesystem_utilization`	Calculé	Formule : `node_filesystem_usage / node_filesystem_capacity` Ces données sont disponibles par nom de périphérique.
Cluster	`cluster_failed_node_count`	Serveur d'API	
Cluster	`cluster_node_count`	Serveur d'API	
Service	`service_number_of_running_pods`	Serveur d'API	
`Namespace`	`namespace_number_of_running_pods`	Serveur d'API	

Exemples de calculs de métriques

Cette section inclut des exemples qui illustrent comment certaines des valeurs figurant dans le tableau précédent sont calculées.

Supposons que vous ayez un cluster à l'état suivant.

```
Node1
   node_cpu_limit = 4
```

```
   node_cpu_usage_total = 3

   Pod1
     pod_cpu_usage_total = 2

     Container1
        container_cpu_limit = 1
        container_cpu_request = 1
        container_cpu_usage_total = 0.8

     Container2
        container_cpu_limit = null
        container_cpu_request = null
        container_cpu_usage_total = 1.2

   Pod2
     pod_cpu_usage_total = 0.4

     Container3
        container_cpu_limit = 1
        container_cpu_request = 0.5
        container_cpu_usage_total = 0.4

Node2
   node_cpu_limit = 8
   node_cpu_usage_total = 1.5

   Pod3
     pod_cpu_usage_total = 1

     Container4
        container_cpu_limit = 2
        container_cpu_request = 2
        container_cpu_usage_total = 1
```

Le tableau suivant montre la façon dont les métriques d'UC du pod sont calculées à l'aide de ces données.

Métrique	Formule	Pod1	Pod2	Pod3
`pod_cpu_utilization`	`pod_cpu_usage_total / node_cpu_limit`	2 / 4 = 50 %	0,4 / 4 = 10 %	1 / 8 = 12,5 %

Métrique	Formule	Pod1	Pod2	Pod3
`pod_cpu_utilization_over_pod_limit`	`pod_cpu_usage_total / sum(container_cpu_limit)`	Non applicable (N/A), car la limite d'UC pour `Container 2` n'est pas définie	0,4 / 1 = 40 %	1 / 2 = 50 %
`pod_cpu_reserved_capacity`	`sum(container_cpu_request) / node_cpu_limit`	(1 + 0) / 4 = 25 %	0,5 / 4 = 12,5 %	2 / 8 = 25 %

Le tableau suivant montre la façon dont les métriques d'UC du nœud sont calculées à l'aide de ces données.

Métrique	Formule	Node1	Node2
`node_cpu_utilization`	`node_cpu_usage_total / node_cpu_limit`	3 / 4 = 75 %	1,5 / 8 = 18,75 %
`node_cpu_reserved_capacity`	`sum(pod_cpu_request) / node_cpu_limit`	1,5 / 4 = 37,5 %	2 / 8 = 25 %

Surveillance des métriques Prometheus Container Insights

CloudWatch La surveillance de Container Insights pour Prometheus automatise la découverte des métriques Prometheus à partir de systèmes et de charges de travail conteneurisés. Prometheus est une boîte à outils de surveillance de systèmes et d'alerte open source. Pour plus d'informations, consultez What is Prometheus? dans la documentation Prometheus.

La découverte des métriques Prometheus est prise en charge pour les clusters Amazon Elastic Container Service, Amazon Elastic Kubernetes Service et Kubernetes s'exécutant

sur des instances Amazon EC2. Les types de métriques de compteur, de jauge et récapitulatives de Prometheus sont collectés. La prise en charge des métriques d'histogramme sera ajoutée lors d'une prochaine mise à jour.

Pour les clusters Amazon ECS et Amazon EKS, les types de lancements EC2 et Fargate sont pris en charge. Container Insights collecte automatiquement les métriques de plusieurs applications, et vous pouvez le configurer pour collecter les métriques à partir de n'importe quelle application.

Vous pouvez adopter Prometheus comme méthode open source et standard ouverte pour intégrer des métriques personnalisées. CloudWatch L' CloudWatch agent soutenu par Prometheus découvre et collecte les métriques Prometheus afin de surveiller, de dépanner et d'avertir plus rapidement en cas de dégradation des performances et de défaillances des applications. Cette méthode réduit également le nombre d'outils de surveillance nécessaires pour améliorer l'observabilité.

Container Insights Le support de Prometheus pay-per-use concerne les métriques et les journaux, y compris la collecte, le stockage et l'analyse. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

Tableaux de bord préconçus pour certaines charges de travail

La solution Container Insights Prometheus inclut des tableaux de bord préconçus pour les applications les plus populaires répertoriées dans cette section. Pour obtenir des exemples de configurations pour ces applications, consultez (En option) Configuration d'exemples d'applications Amazon ECS conteneurisées pour les test de métriques Prometheus et (En option) Configuration d'exemples d'applications Amazon EKS conteneurisées pour les test de métriques Prometheus.

Vous pouvez également configurer Container Insights pour collecter les métriques Prometheus à partir d'autres services et applications conteneurisés, en modifiant le fichier de configuration de l'agent.

Applications avec des tableaux de bord préconçus pour les clusters Amazon EKS et Kubernetes s'exécutant sur les instances Amazon EC2 :

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy

Applications avec des tableaux de bord préconçus pour les clusters Amazon ECS :

- AWS App Mesh
- Java/JMX
- NGINX
- NGINX Plus

Configuration de la collecte de métriques Prometheus sur des clusters Amazon ECS

Pour collecter les métriques Prometheus à partir de clusters Amazon ECS, vous pouvez utiliser CloudWatch l'agent comme collecteur ou utiliser AWS la distribution pour collecteur. OpenTelemetry Pour plus d'informations sur l'utilisation de AWS Distro for OpenTelemetry Collector, consultez https://aws-otel.github.io/docs/getting-started/container-insights/ecs-prometheus.

Les sections suivantes expliquent comment utiliser l' CloudWatch agent comme collecteur pour récupérer les métriques Prometheus. Vous installez l' CloudWatch agent avec Prometheus monitoring sur les clusters exécutant Amazon ECS, et vous pouvez éventuellement configurer l'agent pour récupérer des cibles supplémentaires. Ces sections fournissent également des didacticiels facultatifs pour configurer des exemples d'applications à utiliser pour les tests avec la surveillance Prometheus.

Container Insights sur Amazon ECS prend en charge les combinaisons suivantes de type de lancement et de mode réseau pour les métriques Prometheus :

Type de lancement Amazon ECS	Modes réseau pris en charge
EC2 (Linux)	pont, hôte et awsvpc
Fargate	awsvpc

Exigences de groupe de sécurité VPC

Les règles d'entrée des groupes de sécurité pour les charges de travail Prometheus doivent ouvrir les ports Prometheus à l'agent pour récupérer les métriques Prometheus par CloudWatch l'adresse IP privée.

Les règles de sortie du groupe de sécurité pour l' CloudWatch agent doivent permettre à l'agent de se connecter au CloudWatch port des charges de travail Prometheus via une adresse IP privée.

Rubriques

Installez l' CloudWatch agent avec la collecte de métriques Prometheus sur les clusters Amazon ECS

Cette section explique comment configurer l' CloudWatch agent avec la surveillance Prometheus dans un cluster exécutant Amazon ECS. Après cela, l'agent récupère et importe automatiquement les métriques pour les applications suivantes exécutées dans ce cluster.

- AWS App Mesh
- Java/JMX

Vous pouvez également configurer l'agent pour récupérer et importer les métriques à partir d'applications et sources Prometheus supplémentaires.

Configuration de rôles IAM

Vous avez besoin de deux rôles IAM pour définir la tâche de l' CloudWatch agent. Si vous indiquez **`CreateIAMRoles=True`** dans la AWS CloudFormation pile que Container Insights doit créer ces rôles pour vous, les rôles seront créés avec les autorisations appropriées. Si vous souhaitez les créer vous-même ou utiliser des rôles existants, les autorisations et rôles suivants sont requis.

- CloudWatch rôle de tâche ECS de l' CloudWatch agent : le conteneur de l'agent utilise ce rôle. Elle doit inclure la CloudWatchAgentServerPolicypolitique et une politique gérée par le client qui contient les autorisations en lecture seule suivantes :
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:ListServices`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeServices`
  - `ecs:DescribeTasks`

  - `ecs:DescribeTaskDefinition`
- CloudWatch rôle d'exécution des tâches de l'agent ECS : il s'agit du rôle dont Amazon ECS a besoin pour lancer et exécuter vos conteneurs. Assurez-vous que votre rôle d'exécution des tâches est associé à AmazonSSMTaskExecutionRolePolicy, ReadOnlyAccess AmazonECS et aux politiques. CloudWatchAgentServerPolicy Si vous souhaitez stocker des données plus sensibles que Amazon ECS peut utiliser, consultez Spécification de données sensibles.

Installez l' CloudWatch agent avec le système de surveillance Prometheus en utilisant AWS CloudFormation

Vous pouvez AWS CloudFormation installer l' CloudWatch agent avec Prometheus monitoring pour les clusters Amazon ECS. La liste suivante montre les paramètres que vous utiliserez dans le modèle AWS CloudFormation .

- ECS ClusterName — Spécifie le cluster Amazon ECS cible.
- CreateIAMRoles– Spécifiez **`True`** pour créer de nouveaux rôles pour le rôle de tâche Amazon ECS et le rôle d'exécution de tâche Amazon ECS. Spécifiez **`False`** pour réutiliser les rôles existants.
- TaskRoleName— Si vous avez spécifié **`True`** CreateIamRoles, cela indique le nom à utiliser pour le nouveau rôle de tâche Amazon ECS. Si vous avez spécifié **`False`** pour CreateIAMRoles, cela spécifie le rôle existant à utiliser en tant que rôle de tâche Amazon ECS.
- ExecutionRoleName— Si vous avez spécifié **`True`** CreateIamRoles, cela indique le nom à utiliser pour le nouveau rôle d'exécution de tâches Amazon ECS. Si vous avez spécifié **`False`** pour CreateIAMRoles, cela spécifie le rôle existant à utiliser en tant que rôle d'exécution de tâche Amazon ECS.
- ECS NetworkMode — Si vous utilisez le type de lancement EC2, spécifiez le mode réseau ici. Il doit être **`bridge`** ou **`host`**.
- ECS LaunchType — Spécifiez **`fargate`** soit**`EC2`**.
- SecurityGroupID — Si NetworkMode c'est le cas de l'ECS**`awsvpc`**, spécifiez l'ID du groupe de sécurité ici.
- SubnetId — Si NetworkMode c'est le cas de l'ECS**`awsvpc`**, spécifiez l'ID du sous-réseau ici.

Exemples de commande

Cette section inclut des exemples de AWS CloudFormation commandes pour installer Container Insights avec la surveillance Prometheus dans différents scénarios.

Création d'une AWS CloudFormation pile pour un cluster Amazon ECS en mode réseau en mode pont

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=bridge
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Création d'une AWS CloudFormation pile pour un cluster Amazon ECS en mode réseau hôte

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_NETWORK_MODE=host
export CREATE_IAM_ROLES=True
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name


curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

Création d'une AWS CloudFormation pile pour un cluster Amazon ECS en mode réseau awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=EC2
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                 ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                 ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
```

```
    --profile ${AWS_PROFILE}
```

Créer une AWS CloudFormation pile pour un cluster Fargate en mode réseau awsvpc

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export ECS_CLUSTER_NAME=your_ec2_ecs_cluster_name
export ECS_LAUNCH_TYPE=FARGATE
export CREATE_IAM_ROLES=True
export ECS_CLUSTER_SECURITY_GROUP=your_security_group_eg_sg-xxxxxxxxxx
export ECS_CLUSTER_SUBNET=your_subnet_eg_subnet-xxxxxxxxxx
export ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
export ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-
prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-${ECS_LAUNCH_TYPE}-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=${CREATE_IAM_ROLES} \
                 ParameterKey=ECSLaunchType,ParameterValue=${ECS_LAUNCH_TYPE} \
                 ParameterKey=SecurityGroupID,ParameterValue=
${ECS_CLUSTER_SECURITY_GROUP} \
                 ParameterKey=SubnetID,ParameterValue=${ECS_CLUSTER_SUBNET} \
                 ParameterKey=TaskRoleName,ParameterValue=${ECS_TASK_ROLE_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=
${ECS_EXECUTION_ROLE_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

AWS ressources créées par la AWS CloudFormation pile

Le tableau suivant répertorie les AWS ressources créées lorsque vous configurez la surveillance AWS CloudFormation de Container Insights with Prometheus sur un cluster Amazon ECS.

Type de ressource	Nom de la ressource	Commentaires
AWS::SSM::Parameter	AmazonCloudWatch-*CW -$ ECS_CLUSTER_NAME AgentConfig -$ ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE*	Il s'agit de l' CloudWatch agent avec la définition par défaut du format métrique intégré App Mesh et Java/JMX.
AWS::SSM::Parameter	AmazonCloudWatch-*Prometheus ConfigName -$ ECS_CLUSTER_NAME -$ ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE*	Il s'agit de la configuration de récupération Prometheus.
AWS::IAM::Role	$ECS_TASK_ROLE_NAME.	Le rôle de tâche Amazon ECS. Ceci est créé uniquement si vous avez spécifié **`True`** pour `CREATE_IAM_ROLES` .
AWS::IAM::Role	${ECS_EXECUTION_ROLE_NAME}	Le rôle d'exécution de tâche Amazon ECS. Ceci est créé uniquement si vous avez spécifié **`True`** pour `CREATE_IAM_ROLES` .
AWS::ECS::TaskDefinition	cwagent-prometheus-$*ECS_CLUSTER_NAME* -$*ECS_LAUNCH_TYPE* -$*ECS_NETWORK_MODE*	
AWS::ECS::Service	cwagent-prometheus-replica-service-*$ ECS_LAUNCH_TYPE -$ ECS_NETWORK_MODE*	

Supprimer la AWS CloudFormation pile de l' CloudWatch agent grâce à la surveillance Prometheus

Pour supprimer l' CloudWatch agent d'un cluster Amazon ECS, entrez ces commandes.

```
export AWS_PROFILE=your_aws_config_profile_eg_default
export AWS_DEFAULT_REGION=your_aws_region_eg_ap-southeast-1
export CLOUDFORMATION_STACK_NAME=your_cloudformation_stack_name
```

```
aws cloudformation delete-stack \
--stack-name ${CLOUDFORMATION_STACK_NAME} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Récupération de sources Prometheus supplémentaires et importation de ces métriques

L' CloudWatch agent chargé de surveiller Prometheus a besoin de deux configurations pour récupérer les métriques Prometheus. L'une concerne les configurations standard Prometheus, comme décrit dans <scrape_config> dans la documentation Prometheus. L'autre concerne la configuration de l' CloudWatch agent.

Pour les clusters Amazon ECS, les configurations sont intégrées au Parameter Store de AWS Systems Manager par les secrets de la définition de tâche Amazon ECS :

- Le secret `PROMETHEUS_CONFIG_CONTENT` concerne la configuration de récupération Prometheus.
- Le secret `CW_CONFIG_CONTENT` réside dans la configuration de CloudWatch l'agent.

Pour extraire des sources de métriques Prometheus supplémentaires et les importer, CloudWatch vous devez modifier à la fois la configuration de Prometheus Scrape et la configuration de l'agent, puis redéployer CloudWatch l'agent avec la configuration mise à jour.

Exigences de groupe de sécurité VPC

Les règles d'entrée des groupes de sécurité pour les charges de travail Prometheus doivent ouvrir les ports Prometheus à l'agent pour récupérer les métriques Prometheus par CloudWatch l'adresse IP privée.

Les règles de sortie du groupe de sécurité pour l' CloudWatch agent doivent permettre à l'agent de se connecter au CloudWatch port des charges de travail Prometheus via une adresse IP privée.

Configuration de récupération Prometheus

L' CloudWatch agent prend en charge les configurations standard de Prometheus scrape, comme indiquéhttps://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config <scrape_config>dans la documentation de Prometheus. Vous pouvez modifier cette section pour mettre à jour les configurations déjà présentes dans ce fichier et ajouter des cibles de récupération Prometheus supplémentaires. Par défaut, l'exemple de fichier de configuration contient les lignes de configuration globale suivantes :

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval– Définit la fréquence à laquelle récupérer les cibles.
- scrape_timeout– Définit le temps d'attente avant l'expiration d'une requête de récupération.

Vous pouvez également définir différentes valeurs pour ces paramètres au niveau de la tâche, afin de remplacer les configurations globales.

Tâches de récupération Prometheus

Certaines tâches de scraping par défaut sont déjà configurées dans les fichiers YAML de l' CloudWatch agent. Par exemple, dans les fichiers YAML pour Amazon ECS tels que `cwagent-ecs-prometheus-metric-for-bridge-host.yaml`, les tâches de récupération par défaut sont configurées dans la section `ecs_service_discovery`.

```
"ecs_service_discovery": {
                  "sd_frequency": "1m",
                  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
                  "docker_label": {
                  },
                  "task_definition_list": [
                    {
                      "sd_job_name": "ecs-appmesh-colors",
                      "sd_metrics_ports": "9901",
                      "sd_task_definition_arn_pattern": ".*:task-definition\/.*-
ColorTeller-(white):[0-9]+",
                      "sd_metrics_path": "/stats/prometheus"
                    },
                    {
                      "sd_job_name": "ecs-appmesh-gateway",
                      "sd_metrics_ports": "9901",
                      "sd_task_definition_arn_pattern": ".*:task-definition/.*-
ColorGateway:[0-9]+",
                      "sd_metrics_path": "/stats/prometheus"
                    }
                  ]
                }
```

Chacune de ces cibles par défaut est supprimée et les métriques sont envoyées dans le journal des événements CloudWatch à l'aide d'un format de métrique intégré. Pour plus d'informations, consultez Intégration de métriques dans les journaux.

Les évènements de journaux des clusters Amazon ECS sont stockés dans le groupe de journaux / aws/ecs/containerinsights/***cluster_name***/prometheus.

Chaque tâche de récupération est contenue dans un flux de journaux différent au sein de ce groupe de journaux.

Pour ajouter une nouvelle cible de récupération, vous ajoutez une nouvelle entrée dans la section `task_definition_list` sous la section `ecs_service_discovery` du fichier YAML et redémarrez l'agent. Pour un exemple de ce processus, consultez Didacticiel pour l'ajout d'une nouvelle cible de récupération Prometheus : métrique du serveur d'API Prometheus.

CloudWatch configuration de l'agent pour Prometheus

Le fichier de configuration de l' CloudWatch agent contient une `prometheus` section `metrics_collected` dédiée à la configuration du scraping de Prometheus. Elle inclut les options de configuration suivantes :

- cluster_name– Spécifie le nom du cluster à ajouter en tant qu'étiquette dans l'évènement du journal. Ce champ est facultatif. Si vous l'omettez, l'agent peut détecter le nom du cluster Amazon ECS.
- log_group_name– Spécifie le nom du groupe de journaux pour les métriques Prometheus récupérées. Ce champ est facultatif. *Si vous l'omettez, utilisez CloudWatch* ***/aws/ecs/containerinsights/ cluster_name /prometheus*** *pour les journaux des clusters Amazon ECS.*
- prometheus_config_path– Spécifie le chemin d'accès du fichier de configuration de récupération Prometheus. Si la valeur de ce champ commence par `env:`, le contenu du fichier de configuration de récupération Prometheus sera récupéré à partir de la variable d'environnement du conteneur. Ne modifiez pas ce champ.
- ecs_service_discovery– Il s'agit de la section qui spécifie les configurations des fonctions de découverte automatique de la cible Amazon ECS Prometheus. Deux modes sont pris en charge pour découvrir les cibles Prometheus : découverte basée sur l'étiquette docker du conteneur ou découverte basée sur l'expression régulière ARN de définition de tâche Amazon ECS. Vous pouvez utiliser les deux modes ensemble et l' CloudWatch agent dédupliquera les cibles découvertes en fonction de : {private_ip} : {port}/{metrics_path}.

La section `ecs_service_discovery` peut contenir les champs suivants :

- `sd_frequency` est la fréquence de découverte des exportateurs Prometheus. Spécifiez un nombre et un suffixe d'unité. Par exemple, `1m` pour une fois par minute ou `30s` pour une fois toutes les 30 secondes. Les suffixes d'unités valides sont `ns`, `us`, `ms`, `s`, `m` et `h`.

  Ce champ est facultatif. La valeur par défaut est de 60 secondes (1 minute).
- `sd_target_cluster` est le nom du cluster Amazon ECS cible pour la découverte automatique. Ce champ est facultatif. Le nom par défaut est le nom du cluster Amazon ECS sur lequel l' CloudWatch agent est installé.
- `sd_cluster_region` est la région du cluster Amazon ECS cible. Ce champ est facultatif. La valeur par défaut est la région du cluster Amazon ECS dans laquelle l' CloudWatchagent est installé.
- `sd_result_file` est le chemin d'accès du fichier YAML pour les résultats de la cible Prometheus. La configuration de récupération Prometheus fera référence à ce fichier.
- `docker_label` est une section facultative que vous pouvez utiliser pour spécifier la configuration de la découverte de service basée sur des étiquettes de docker. Si vous omettez cette section, la découverte basée sur les étiquettes docker n'est pas utilisée. Cette section peut contenir les champs suivants :
  - `sd_port_label` est le nom de l'étiquette docker du conteneur qui spécifie le port du conteneur pour les métriques Prometheus. La valeur par défaut est `ECS_PROMETHEUS_EXPORTER_PORT`. Si le conteneur ne possède pas cette étiquette docker, l' CloudWatch agent l'ignorera.
  - `sd_metrics_path_label` est le nom de l'étiquette docker du conteneur qui spécifie le chemin d'accès aux métriques Prometheus. La valeur par défaut est `ECS_PROMETHEUS_METRICS_PATH`. Si le conteneur n'a pas cette étiquette docker, l'agent utilise le chemin par défaut `/metrics`.
  - `sd_job_name_label` est le nom de l'étiquette docker du conteneur qui spécifie le nom de la tâche de récupération Prometheus. La valeur par défaut est `job`. Si le conteneur ne possède pas cette étiquette docker, l' CloudWatch agent utilise le nom de la tâche dans la configuration Prometheus Scrape.
- `task_definition_list` est une section facultative que vous pouvez utiliser pour spécifier la configuration de la découverte de service basée sur les définitions de tâches. Si vous omettez cette section, la découverte basée sur les définitions de tâches n'est pas utilisée. Cette section peut contenir les champs suivants :

    - `sd_task_definition_arn_pattern` est le modèle à utiliser pour spécifier les définitions de tâches Amazon ECS à découvrir. Il s'agit d'une expression régulière.
    - `sd_metrics_ports` répertorie le containerPort pour les métriques Prometheus. Séparez les containerPorts par des points-virgules.
    - `sd_container_name_pattern` spécifie les noms des conteneurs de tâches Amazon ECS. Il s'agit d'une expression régulière.
    - `sd_metrics_path` spécifie le chemin de métrique Prometheus. Si vous ne spécifiez pas ce paramètre, l'agent utilise le chemin par défaut `/metrics`
    - `sd_job_name` spécifie le nom de la tâche de récupération Prometheus. Si vous omettez ce champ, l' CloudWatch agent utilise le nom de la tâche dans la configuration Prometheus Scrape.
  - `service_name_list_for_tasks` est une section facultative que vous pouvez utiliser pour spécifier la configuration de la découverte de service basée sur les noms. Si vous omettez cette section, la découverte basée sur les noms n'est pas utilisée. Cette section peut contenir les champs suivants :
    - `sd_service_name_pattern` est le modèle à utiliser pour spécifier l'Amazon ECS service où les tâches doivent être découvertes. Il s'agit d'une expression régulière.
    - `sd_metrics_ports` répertorie le `containerPort` pour les métriques Prometheus. Séparez plusieurs `containerPorts` avec des points-virgules.
    - `sd_container_name_pattern` spécifie les noms des conteneurs de tâches Amazon ECS. Il s'agit d'une expression régulière.
    - `sd_metrics_path` spécifie le chemin d'accès aux métriques Prometheus. Si vous ne spécifiez pas ce paramètre, l'agent utilise le chemin par défaut `/metrics`.
    - `sd_job_name` spécifie le nom de la tâche de récupération Prometheus. Si vous omettez ce champ, l' CloudWatch agent utilise le nom de la tâche dans la configuration Prometheus Scrape.
- metric_declaration – Ce sont des sections qui spécifient le tableau de journaux avec le format de métrique intégré à générer. Il existe des `metric_declaration` sections pour chaque source Prometheus à partir de laquelle CloudWatch l'agent importe par défaut. Chacune de ces sections comprend les champs suivants :
  - `label_matcher` est une expression régulière qui vérifie la valeur des étiquettes répertoriées dans `source_labels`. Les métriques correspondantes sont activées pour être incluses dans le format de métrique intégré envoyé à CloudWatch.

Si plusieurs étiquettes sont spécifiées dans `source_labels`, nous vous recommandons de ne pas utiliser les caractères ^ ou $ dans l'expression régulière pour `label_matcher`.

- `source_labels` spécifie la valeur des étiquettes qui sont vérifiées par la ligne `label_matcher`.
- `label_separator` spécifie le séparateur à utiliser dans la ligne `label_matcher` si plusieurs `source_labels` sont spécifiées. La valeur par défaut est `;`. Vous pouvez voir cette valeur par défaut utilisée dans la ligne `label_matcher` dans l'exemple suivant.
- `metric_selectors`est une expression régulière qui spécifie les métriques à collecter et à envoyer CloudWatch.
- `dimensions`est la liste des étiquettes à utiliser comme CloudWatch dimensions pour chaque métrique sélectionnée.

Consultez l'exemple `metric_declaration` suivant.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system$",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

Cet exemple montre comment configurer une section de format de métrique intégrée à envoyer en tant qu'événement de journaux si les conditions suivantes sont remplies :

- La valeur de `Service` contient `node-exporter` ou `kube-dns`.
- La valeur de `Namespace` est `kube-system`.
- La métrique Prometheus `coredns_dns_request_type_count_total` contient les deux étiquettes `Namespace` et `Service`.

L'événement de journal envoyé inclut la section en surbrillance suivante :

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "Namespace":"kube-system",
   "Service":"kube-dns",
   "coredns_dns_request_type_count_total":2562,
   "eks_amazonaws_com_component":"kube-dns",
   "instance":"192.168.61.254:9153",
   "job":"kubernetes-service-endpoints",
   ...
}
```

Guide détaillé de la découverte automatique sur les clusters Amazon ECS

Prometheus fournit des dizaines de mécanismes dynamiques de découverte de service, comme décrit dans <scrape_config>. Toutefois, il n'y a pas de découverte de service intégrée pour Amazon ECS. L' CloudWatch agent ajoute ce mécanisme.

Lorsque la découverte du service Amazon ECS Prometheus est activée, CloudWatch l'agent effectue régulièrement les appels d'API suivants aux frontends Amazon ECS et Amazon EC2 pour récupérer les métadonnées des tâches ECS en cours d'exécution dans le cluster ECS cible.

```
EC2:DescribeInstances
ECS:ListTasks
ECS:ListServices
ECS:DescribeContainerInstances
ECS:DescribeServices
ECS:DescribeTasks
```

```
ECS:DescribeTaskDefinition
```

Les métadonnées sont utilisées par l' CloudWatch agent pour scanner les cibles Prometheus au sein du cluster ECS. L' CloudWatch agent prend en charge trois modes de découverte de services :

- découverte de service basée sur l'étiquette docker du conteneur ;
- découverte de service basée sur l'expression régulière ARN de la tâche ECS ;
- découverte de service basée sur l'expression régulière du nom du service ECS.

Tous les modes peuvent être utilisés ensemble. CloudWatch l'agent déduplique les cibles découvertes en fonction de :. `{private_ip}:{port}/{metrics_path}`

Toutes les cibles découvertes sont écrites dans un fichier de résultats spécifié par le champ `sd_result_file` de configuration du conteneur de l' CloudWatch agent. Voici un exemple de fichier de résultats :

```
- targets:
  - 10.6.1.95:32785
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT: "9406"
    ECS_PROMETHEUS_JOB_NAME: demo-jar-ec2-bridge-dynamic
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-jar-ec2-bridge-dynamic-port
    TaskGroup: family:demo-jar-ec2-bridge-dynamic-port
    TaskRevision: "7"
    VpcId: vpc-01234567890
    container_name: demo-jar-ec2-bridge-dynamic-port
    job: demo-jar-ec2-bridge-dynamic
- targets:
  - 10.6.3.193:9404
  labels:
    __metrics_path__: /metrics
    ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_B: "9404"
    ECS_PROMETHEUS_JOB_NAME: demo-tomcat-ec2-bridge-mapped-port
    ECS_PROMETHEUS_METRICS_PATH: /metrics
    InstanceType: t3.medium
    LaunchType: EC2
```

```
    SubnetId: subnet-123456789012
    TaskDefinitionFamily: demo-tomcat-ec2-bridge-mapped-port
    TaskGroup: family:demo-jar-tomcat-bridge-mapped-port
    TaskRevision: "12"
    VpcId: vpc-01234567890
    container_name: demo-tomcat-ec2-bridge-mapped-port
    job: demo-tomcat-ec2-bridge-mapped-port
```

Vous pouvez directement intégrer ce fichier de résultats à la découverte de services basés sur des fichiers de Prometheus. Pour plus d'informations sur la découverte de services basée sur des fichiers de Prometheus, consultez <file_sd_config>.

Supposons que le fichier de résultat soit écrit dans `/tmp/cwagent_ecs_auto_sd.yaml`. La configuration de récupération Prometheus suivante la consommera.

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: cwagent-ecs-file-sd-config
    sample_limit: 10000
    file_sd_configs:
      - files: [ "/tmp/cwagent_ecs_auto_sd.yaml" ]
```

L' CloudWatch agent ajoute également les étiquettes supplémentaires suivantes pour les cibles découvertes.

- `container_name`
- `TaskDefinitionFamily`
- `TaskRevision`
- `TaskGroup`
- `StartedBy`
- `LaunchType`
- `job`
- `__metrics_path__`
- Étiquettes Docker

Lorsque le cluster comporte le type de lancement EC2, les trois étiquettes suivantes sont ajoutées.

- `InstanceType`
- `VpcId`
- `SubnetId`

> **Note**
>
> Les étiquettes Docker qui ne correspondent pas à l'expression régulière `[a-zA-Z_][a-zA-Z0-9_]*` sont filtrés. Cela correspond aux conventions Prometheus répertoriées dans `label_name` de Fichier de configuration, compris dans la documentation Prometheus.

Exemples de configuration de la découverte de service ECS

Cette section contient des exemples qui illustrent la découverte de services ECS.

Exemple 1

```
"ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
  }
}
```

Cet exemple montre comment activer la découverte de service basée sur les étiquettes docker. L' CloudWatch agent interrogera les métadonnées des tâches ECS une fois par minute et inscrira les cibles découvertes dans le `/tmp/cwagent_ecs_auto_sd.yaml` fichier du conteneur de l' CloudWatch agent.

La valeur par défaut de `sd_port_label` dans la section `docker_label` est `ECS_PROMETHEUS_EXPORTER_PORT`. Si un conteneur en cours d'exécution dans les tâches ECS possède une étiquette `ECS_PROMETHEUS_EXPORTER_PORT` docker, l' CloudWatch agent utilise sa valeur as `container port` pour scanner tous les ports exposés du conteneur. En cas de correspondance, le port hôte mappé ainsi que l'adresse IP privée du conteneur sont utilisés pour construire la cible de l'exportateur Prometheus au format suivant : `private_ip:host_port`.

La valeur par défaut de `sd_metrics_path_label` dans la section `docker_label` est `ECS_PROMETHEUS_METRICS_PATH`. Si le conteneur possède cette étiquette docker, sa valeur sera

utilisée comme `__metrics_path__`. Si le conteneur ne possède pas cette étiquette, la valeur par défaut `/metrics` est utilisée.

La valeur par défaut de `sd_job_name_label` dans la section `docker_label` est `job`. Si le conteneur possède cette étiquette docker, sa valeur sera ajoutée comme l'une des étiquettes de la cible pour remplacer le nom de tâche par défaut spécifié dans la configuration Prometheus. La valeur de cette étiquette Docker est utilisée comme nom du flux de journaux dans le groupe de CloudWatch journaux Logs.

Exemple 2

```
"ecs_service_discovery": {
  "sd_frequency": "15s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A",
    "sd_job_name_label": "ECS_PROMETHEUS_JOB_NAME"
  }
}
```

Cet exemple montre comment activer la découverte de service basée sur les étiquettes docker. L' CloudWatch agent interroge les métadonnées des tâches ECS toutes les 15 secondes et écrit les cibles découvertes dans le `/tmp/cwagent_ecs_auto_sd.yaml` fichier du conteneur de l' CloudWatch agent. Les conteneurs possédant une étiquette docker `ECS_PROMETHEUS_EXPORTER_PORT_SUBSET_A` seront analysés. La valeur de l'étiquette docker `ECS_PROMETHEUS_JOB_NAME` est utilisée comme nom de tâche.

Exemple 3

```
"ecs_service_discovery": {
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "task_definition_list": [
    {
      "sd_job_name": "java-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9404; 9406",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*javajmx.*:[0-9]+"
    },
    {
      "sd_job_name": "envoy-prometheus",
```

```
      "sd_metrics_path": "/stats/prometheus",
      "sd_container_name_pattern": "^envoy$",
      "sd_metrics_ports": "9901",
      "sd_task_definition_arn_pattern": ".*:task-definition/.*appmesh.*:23"
    }
  ]
}
```

Cet exemple montre comment activer la découverte de service basée sur l'expression régulière ARN de la tâche ECS. L' CloudWatch agent interroge les métadonnées des tâches ECS toutes les cinq minutes et écrit les cibles découvertes dans le `/tmp/cwagent_ecs_auto_sd.yaml` fichier du conteneur de l' CloudWatch agent.

Deux sections d'expressions régulières ARN de définition de tâche sont définies :

- Pour la première section, les tâches ECS avec `javajmx` dans leur ARN de définition de tâche ECS sont filtrées pour l'analyse des ports de conteneur. Si les conteneurs de ces tâches ECS exposent le port de conteneur sur 9404 ou 9406, le port hôte mappé ainsi que l'adresse IP privée du conteneur sont utilisés pour créer les cibles d'exportation Prometheus. La valeur de `sd_metrics_path` définit `__metrics_path__` sur `/metrics`. L' CloudWatch agent extraira donc les métriques Prometheus, qui seront envoyées au flux `private_ip:host_port/metrics` CloudWatch de journaux dans Logs `java-prometheus` du groupe de journaux. `/aws/ecs/containerinsights/cluster_name/prometheus`
- Pour la deuxième section, les tâches ECS avec `appmesh` dans leur ARN de définition de tâche ECS et avec la `version :23` sont filtrées pour l'analyse des ports de conteneur. Pour les conteneurs dont le nom est `envoy` qui exposent le port de conteneur sur `9901`, le port hôte mappé ainsi que l'adresse IP privée du conteneur sont utilisés pour créer les cibles d'exportation Prometheus. La valeur de ces tâches ECS expose le port de conteneur sur 9404 ou 9406, le port hôte mappé ainsi que l'adresse IP privée du conteneur sont utilisés pour créer les cibles d'exportation Prometheus. La valeur de `sd_metrics_path` définit `__metrics_path__` sur `/stats/prometheus`. L' CloudWatchagent extraira donc les métriques Prometheus et les enverra au flux `private_ip:host_port/stats/prometheus` CloudWatch de journaux dans Logs `envoy-prometheus` du groupe de journaux. `/aws/ecs/containerinsights/cluster_name/prometheus`

Exemple 4

```
"ecs_service_discovery": {
```

```
  "sd_frequency": "5m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "service_name_list_for_tasks": [
    {
      "sd_job_name": "nginx-prometheus",
      "sd_metrics_path": "/metrics",
      "sd_metrics_ports": "9113",
      "sd_service_name_pattern": "^nginx-.*"
    },
    {
      "sd_job_name": "haproxy-prometheus",
      "sd_metrics_path": "/stats/metrics",
      "sd_container_name_pattern": "^haproxy$",
      "sd_metrics_ports": "8404",
      "sd_service_name_pattern": ".*haproxy-service.*"
    }
  ]
}
```

Cet exemple montre comment activer la découverte de service basée sur l'expression régulière du nom du service ECS. L' CloudWatch agent interroge les métadonnées des services ECS toutes les cinq minutes et écrit les cibles découvertes dans le `/tmp/cwagent_ecs_auto_sd.yaml` fichier du conteneur de l' CloudWatch agent.

Deux sections d'expressions régulières de nom de service sont définies :

- Dans la première section, les tâches ECS associées aux services ECS dont les noms correspondent à l'expression régulière `^nginx-.*` sont filtrées pour l'analyse du port du conteneur. Si les conteneurs de ces tâches ECS exposent le port de conteneur sur 9113, le port hôte mappé ainsi que l'adresse IP privée du conteneur sont utilisés pour créer les cibles d'exportation Prometheus. La valeur de `sd_metrics_path` définit `__metrics_path__` sur `/metrics`. L' CloudWatch agent extraira donc les métriques Prometheus, et les métriques `private_ip:host_port/metrics` supprimées seront envoyées au flux CloudWatch de journaux dans Logs `nginx-prometheus` du groupe de journaux. `/aws/ecs/containerinsights/cluster_name/prometheus`
- ou la seconde section, les tâches ECS associées aux services ECS dont les noms correspondent à l'expression régulière `.*haproxy-service.*` sont filtrées pour l'analyse du port du conteneur. Pour les conteneurs dont le nom est `haproxy` qui exposent le port de conteneur sur 8404, le port hôte mappé ainsi que l'adresse IP privée du conteneur sont utilisés pour créer les cibles d'exportation Prometheus. La valeur de `sd_metrics_path` définit `__metrics_path__`

sur `/stats/metrics`. L' CloudWatch agent extraira donc les métriques Prometheus, et les métriques `private_ip:host_port/stats/metrics` supprimées seront envoyées au flux CloudWatch de journaux dans Logs `haproxy-prometheus` du groupe de journaux. `/aws/ecs/containerinsights/cluster_name/prometheus`

Exemple 5

```
"ecs_service_discovery": {
  "sd_frequency": "1m30s",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
  "docker_label": {
    "sd_port_label": "MY_PROMETHEUS_EXPORTER_PORT_LABEL",
    "sd_metrics_path_label": "MY_PROMETHEUS_METRICS_PATH_LABEL",
    "sd_job_name_label": "MY_PROMETHEUS_METRICS_NAME_LABEL"
  }
  "task_definition_list": [
    {
      "sd_metrics_ports": "9150",
      "sd_task_definition_arn_pattern": "*memcached.*"
    }
  ]
}
```

Cet exemple montre comment activer les deux modes de découverte de service ECS. L' CloudWatch agent interroge les métadonnées des tâches ECS toutes les 90 secondes et écrit les cibles découvertes dans le `/tmp/cwagent_ecs_auto_sd.yaml` fichier du conteneur de l' CloudWatch agent.

Pour la configuration de découverte de service basée sur Docker :

- Les tâches ECS avec l'étiquette docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` seront filtrées pour l'analyse du port Prometheus. Le port de conteneur Prometheus cible est spécifié par la valeur de l'étiquette `MY_PROMETHEUS_EXPORTER_PORT_LABEL`.
- La valeur de l'étiquette docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` est utilisée pour `__metrics_path__`. Si le conteneur ne possède pas cette étiquette docker, la valeur par défaut `/metrics` est utilisée.
- La valeur de l'étiquette docker `MY_PROMETHEUS_EXPORTER_PORT_LABEL` est utilisée comme étiquette de tâche. Si le conteneur n'a pas cette étiquette docker, le nom de tâche défini dans la configuration Prometheus est utilisé.

Pour la configuration de la découverte de service basée sur l'expression régulière ARN de la tâche ECS :

- Les tâches ECS avec `memcached` dans l'ARN de définition de tâche ECS sont filtrées pour l'analyse des ports de conteneur. Le port de conteneur Prometheus cible est 9150 tel que défini par `sd_metrics_ports`. Le chemin d'accès aux métriques par défaut `/metrics` est utilisé. Le nom de tâche défini dans la configuration Prometheus est utilisé.

(En option) Configuration d'exemples d'applications Amazon ECS conteneurisées pour les test de métriques Prometheus

Pour tester la prise en charge des métriques Prometheus CloudWatch dans Container Insights, vous pouvez configurer une ou plusieurs des charges de travail conteneurisées suivantes. L' CloudWatch agent prenant en charge Prometheus collecte automatiquement les métriques de chacune de ces charges de travail. Pour voir les métriques collectées par défaut, consultez Métriques Prometheus collectées par l'agent CloudWatch .

Rubriques

- Exemple d'application App Mesh pour les clusters Amazon ECS
- Exemple d'application Java/JMX pour les clusters Amazon ECS
- Exemple d'application NGINX pour les clusters Amazon ECS
- Exemple d'application NGINX Plus pour les clusters Amazon ECS
- Didacticiel pour l'ajout d'une nouvelle cible de récupération Prometheus : Memcached sur Amazon ECS
- Didacticiel pour récupérer les métriques Redis Prometheus sur Amazon ECS Fargate

Exemple d'application App Mesh pour les clusters Amazon ECS

Pour collecter des métriques à partir d'un exemple d'application Prometheus pour Amazon ECS, vous devez exécuter Container Insights dans le cluster. Pour plus d'informations sur l'installation de Container Insights, consultez Configuration de Container Insights sur Amazon ECS.

Tout d'abord, suivez cette démonstration pour déployer l'exemple d'application de couleur dans votre cluster Amazon ECS. Une fois que vous avez terminé, les métriques App Mesh Prometheus seront exposées sur le port 9901.

Procédez ensuite comme suit pour installer l' CloudWatch agent avec Prometheus monitoring sur le même cluster Amazon ECS où vous avez installé l'application couleur. Les étapes décrites dans cette section permettent d'installer l' CloudWatch agent en mode réseau en mode pont.

Les variables d'environnement `ENVIRONMENT_NAME`,`AWS_PROFILE`, et `AWS_DEFAULT_REGION` que vous définissez dans la démonstration seront également utilisées dans les étapes suivantes.

Pour installer l' CloudWatch agent avec la surveillance Prometheus à des fins de test

1. Téléchargez le AWS CloudFormation modèle en saisissant la commande suivante.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
bridge-host.yaml
```

2. Définissez le mode réseau en saisissant les commandes suivantes.

```
export ECS_CLUSTER_NAME=${ENVIRONMENT_NAME}
export ECS_NETWORK_MODE=bridge
```

3. Créez la AWS CloudFormation pile en saisissant les commandes suivantes.

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=${ECS_CLUSTER_NAME} \
                 ParameterKey=CreateIAMRoles,ParameterValue=True \
                 ParameterKey=ECSNetworkMode,ParameterValue=${ECS_NETWORK_MODE} \
                 ParameterKey=TaskRoleName,ParameterValue=CWAgent-Prometheus-
TaskRole-${ECS_CLUSTER_NAME} \
                 ParameterKey=ExecutionRoleName,ParameterValue=CWAgent-Prometheus-
ExecutionRole-${ECS_CLUSTER_NAME} \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --profile ${AWS_PROFILE}
```

4. (Facultatif) Lorsque la AWS CloudFormation pile est créée, un `CREATE_COMPLETE` message s'affiche. Si vous souhaitez vérifier l'état avant d'afficher ce message, saisissez la commande suivante.

```
aws cloudformation describe-stacks \
```

```
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--query 'Stacks[0].StackStatus' \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Dépannage

Les étapes de la démonstration utilisent jq pour analyser le résultat de sortie de la AWS CLI. Pour plus d'informations sur l'installation de jq, consultez jq. Utilisez la commande suivante pour définir le format de sortie par défaut de votre AWS CLI sur JSON afin que jq puisse l'analyser correctement.

```
$ aws configure
```

Lorsque la réponse arrive à `Default output format`, saisissez **`json`**.

Désinstallez l' CloudWatch agent avec le système de surveillance Prometheus

Lorsque vous avez terminé le test, entrez la commande suivante pour désinstaller l' CloudWatchagent en supprimant la AWS CloudFormation pile.

```
aws cloudformation delete-stack \
--stack-name CWAgent-Prometheus-ECS-${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
--region ${AWS_DEFAULT_REGION} \
--profile ${AWS_PROFILE}
```

Exemple d'application Java/JMX pour les clusters Amazon ECS

JMX Exporter est un exportateur Prometheus officiel qui peut récupérer et exposer des mBeans JMX en tant que métriques Prometheus. Pour plus d'informations, consultez prometheus/jmx_exporter.

L' CloudWatch agent compatible avec Prometheus supprime les métriques Java/JMX Prometheus en fonction de la configuration de découverte de services dans le cluster Amazon ECS. Vous pouvez configurer JMX Exporter pour qu'il expose les métriques sur un autre port ou chemin d'accès. Si vous modifiez le port ou le chemin, mettez à jour la `ecs_service_discovery` section par défaut dans la configuration de l' CloudWatchagent.

Pour collecter des métriques à partir d'un exemple d'application Prometheus pour Amazon ECS, vous devez exécuter Container Insights dans le cluster. Pour plus d'informations sur l'installation de Container Insights, consultez Configuration de Container Insights sur Amazon ECS.

Pour installer l'exemple d'application Java/JMX pour les clusters Amazon ECS

1. Suivez les étapes décrites dans ces sections pour créer vos images Docker.

   - Exemple : image Docker d'application Java Jar avec métriques Prometheus
   - Exemple : image Apache Tomcat Docker avec métriques Prometheus

2. Spécifiez les deux étiquettes docker suivantes dans le fichier de définition de tâche Amazon ECS. Vous pouvez ensuite exécuter la définition de tâche en tant que Amazon ECS service ou tâche Amazon ECS dans le cluster.

   - Définissez `ECS_PROMETHEUS_EXPORTER_PORT` pour pointer vers le ContainerPort où les métriques Prometheus sont exposées.
   - Définissez `Java_EMF_Metrics` sur `true`. L' CloudWatch agent utilise cet indicateur pour générer le format de métrique intégré dans le journal des événements.

   Voici un exemple :

```
{
  "family": "workload-java-ec2-bridge",
  "taskRoleArn": "{{task-role-arn}}",
  "executionRoleArn": "{{execution-role-arn}}",
  "networkMode": "bridge",
  "containerDefinitions": [
    {
      "name": "tomcat-prometheus-workload-java-ec2-bridge-dynamic-port",
      "image": "your_docker_image_tag_for_tomcat_with_prometheus_metrics",
      "portMappings": [
        {
          "hostPort": 0,
          "protocol": "tcp",
          "containerPort": 9404
        }
      ],
      "dockerLabels": {
        "ECS_PROMETHEUS_EXPORTER_PORT": "9404",
        "Java_EMF_Metrics": "true"
      }
    }
  ],
  "requiresCompatibilities": [
    "EC2"  ],
```

```
"cpu": "256",
"memory": "512"
}
```

Le paramètre par défaut de l' CloudWatch agent dans le AWS CloudFormation modèle permet à la fois la découverte de services basée sur les étiquettes docker et la découverte de services basée sur l'ARN par définition de tâches. Pour consulter ces paramètres par défaut, reportez-vous à la ligne 65 du fichier de configuration YAML de l' CloudWatch agent. Les conteneurs avec l'étiquette `ECS_PROMETHEUS_EXPORTER_PORT` seront automatiquement découverts en fonction du port de conteneur spécifié pour la récupération Prometheus.

Le paramètre par défaut de l' CloudWatch agent inclut également le `metric_declaration` paramètre Java/JMX à la ligne 112 du même fichier. Toutes les étiquettes docker des conteneurs cibles seront ajoutées en tant qu'étiquettes supplémentaires dans les métriques Prometheus et envoyées à Logs. CloudWatch Pour les conteneurs Java/JMX avec étiquette docker `Java_EMF_Metrics="true"`, le format de métrique intégrée sera généré.

Exemple d'application NGINX pour les clusters Amazon ECS

NGINX Prometheus Exporter peut récupérer et exposer les données NGINX en tant que métriques Prometheus. Cet exemple utilise l'exportateur conjointement au service de proxy inverse NGINX pour Amazon ECS.

Pour plus d'informations sur l'exportateur Prometheus de NGINX, rendez-vous sur Github. nginx-prometheus-exporter Pour plus d'informations sur le proxy inverse NGINX, consultez ecs-nginx-reverse-proxyGithub.

L' CloudWatch agent compatible avec Prometheus extrait les métriques NGINX Prometheus en fonction de la configuration de découverte de services dans le cluster Amazon ECS. Vous pouvez configurer NGINX Prometheus Exporter pour qu'il expose les métriques sur un autre port ou chemin d'accès. Si vous modifiez le port ou le chemin, mettez à jour la `ecs_service_discovery` section dans le fichier de configuration de l' CloudWatch agent.

Installation de l'application d'exemple de proxy inverse NGINX pour les clusters Amazon ECS

Suivez ces étapes pour installer l'application d'exemple de proxy inverse NGINX.

Création des images Docker

Pour créer les images Docker pour l'application d'exemple de proxy inverse NGINX

1. Téléchargez le dossier suivant depuis le dépôt de proxy inverse NGINX : https://github.com/awslabs/ /tree/master/reverse-proxy/. ecs-nginx-reverse-proxy
2. Recherchez le répertoire app et créez une image à partir de ce répertoire :

   ```
   docker build -t web-server-app ./path-to-app-directory
   ```

3. Créez une image personnalisée pour NGINX. Pour commencer, créez un répertoire avec les deux fichiers suivants :

   - Un exemple de fichier Docker :

     ```
     FROM nginx
     COPY nginx.conf /etc/nginx/nginx.conf
     ```

   - Un `nginx.conf` fichier, modifié depuis https://github.com/awslabs/ ecs-nginx-reverse-proxy / tree/master/reverse-proxy/ :

     ```
     events {
       worker_connections 768;
     }

     http {
       # Nginx will handle gzip compression of responses from the app server
       gzip on;
       gzip_proxied any;
       gzip_types text/plain application/json;
       gzip_min_length 1000;

       server{
         listen 8080;
         location /stub_status {
             stub_status   on;
         }
       }

       server {
         listen 80;
     ```

```
    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://app:3000;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

> **Note**
>
> `stub_status` doit être activé sur le même port sur lequel `nginx-prometheus-exporter` est configuré pour récupérer les métriques. Dans notre exemple de définition de tâche, `nginx-prometheus-exporter` est configuré pour extraire les métriques depuis le port 8080.

4. Créez une image à partir des fichiers de votre nouveau répertoire :

```
docker build -t nginx-reverse-proxy ./path-to-your-directory
```

5. Téléchargez vos nouvelles images dans un répertoire d'images pour une utilisation ultérieure.

Création de la définition de tâche pour exécuter NGINX et l'application serveur web dans Amazon ECS

Ensuite, vous configurez la définition de tâche.

Cette définition de tâche permet la collecte et l'exportation des métriques NGINX Prometheus. Le conteneur NGINX suit les entrées de l'application et expose ces données au port 8080, comme défini

dans `nginx.conf`. Le conteneur d'exportation NGINX Prometheus récupère ces métriques et les publie sur le port 9113, pour les utiliser dans. CloudWatch

Pour configurer la définition de tâche pour l'application Amazon ECS d'exemple NGINX

1. Créez un fichier JSON de définition de tâche avec le contenu suivant. Remplacez *your-customized-nginx-iamge*par l'URI de l'image de votre image NGINX personnalisée et remplacez *your-web-server-app-image* par l'URI de l'image de votre application de serveur Web.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 256,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
        "-nginx.scrape-uri",
```

```
        "http://nginx:8080/stub_status"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-sample-stack"
}
```

2. Enregistrez la définition de tâche en saisissant la commande suivante.

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Créez un service pour exécuter la tâche en saisissant la commande suivante :

   Veillez à ne pas modifier le nom du service. Nous exécuterons un service d' CloudWatch agent utilisant une configuration qui recherche les tâches en utilisant les modèles de noms des services qui les ont lancées. Par exemple, pour que l' CloudWatch agent trouve la tâche lancée par cette commande, vous pouvez spécifier la valeur de `sd_service_name_pattern` to be`^nginx-service$`. La section suivante fournit plus de détails.

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-service \
 --task-definition nginx-sample-stack:1 \
 --desired-count 1
```

Configurer l' CloudWatch agent pour extraire les métriques NGINX Prometheus

La dernière étape consiste à configurer l' CloudWatch agent pour récupérer les métriques NGINX. Dans cet exemple, l' CloudWatch agent découvre la tâche via le modèle de nom de service et le port

9113, où l'exportateur expose les métriques prometheus pour NGINX. Une fois la tâche découverte et les métriques disponibles, l' CloudWatch agent commence à publier les métriques collectées dans le flux de log nginx-prometheus-exporter.

Pour configurer l' CloudWatch agent afin de récupérer les métriques NGINX

1. Téléchargez la dernière version du fichier YAML nécessaire en saisissant la commande suivante.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

2. Ouvrez le fichier dans un éditeur de texte et trouvez la configuration complète de l' CloudWatch agent dans la `value` clé de la section. `resource:CWAgentConfigSSMParameter` Ensuite, dans la section `ecs_service_discovery`, ajoutez la section `service_name_list_for_tasks` suivante.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-service$"
   }
],
```

3. Dans le même fichier, ajoutez la section suivante dans la section `metric_declaration` pour autoriser les métriques NGINX. Veillez à suivre le modèle d'indentation existant.

```
{
  "source_labels": ["job"],
  "label_matcher": ".*nginx.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginx_.*$"
  ]
},
```

4. Si l' CloudWatch agent n'est pas déjà déployé dans ce cluster, passez à l'étape 8.

Si l' CloudWatch agent est déjà déployé dans le cluster Amazon ECS à l'aide de AWS CloudFormation, vous pouvez créer un ensemble de modifications en saisissant les commandes suivantes :

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-scraping-support
```

5. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
6. Passez en revue le nouvel ensemble de modifications. nginx-scraping-support Vous devriez voir une modification appliquée à la ressource CW AgentConfig SSMParameter. Exécutez le changeset et redémarrez la tâche de l' CloudWatch agent en saisissant la commande suivante :

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Patientez environ 10 secondes, puis saisissez la commande suivante.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
```

```
--region $AWS_REGION
```

8. Si vous installez l' CloudWatch agent avec la collecte de métriques Prometheus sur le cluster pour la première fois, entrez les commandes suivantes.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Affichage des journaux et métriques NGINX

Vous pouvez maintenant afficher les métriques NGINX collectées.

Pour examiner les métriques de votre application NGINX

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans la région où votre cluster s'exécute, choisissez Metrics (Métriques) dans le volet de navigation de gauche. Trouvez l'espace de noms ContainerInsights/Prometheus pour voir les métriques.
3. Pour voir les événements CloudWatch Logs, choisissez Log groups dans le volet de navigation. *Les événements se trouvent dans le groupe de journaux* ***/aws/containerinsights/*** *your_cluster_name /prometheus, dans le flux de journaux. nginx-prometheus-exporter*

Exemple d'application NGINX Plus pour les clusters Amazon ECS

NGINX Plus est la version commerciale de NGINX. Vous devez disposer d'une licence pour l'utiliser. Pour en savoir plus, consultez NGINX Plus.

NGINX Prometheus Exporter peut récupérer et exposer les données NGINX en tant que métriques Prometheus. Cet exemple utilise l'exportateur conjointement au service de proxy inverse NGINX Plus pour Amazon ECS.

Pour plus d'informations sur l'exportateur Prometheus de NGINX, rendez-vous sur Github. nginx-prometheus-exporter Pour plus d'informations sur le proxy inverse NGINX, consultez ecs-nginx-reverse-proxyGithub.

L' CloudWatch agent compatible avec Prometheus extrait les métriques NGINX Plus Prometheus en fonction de la configuration de découverte de services dans le cluster Amazon ECS. Vous pouvez configurer NGINX Prometheus Exporter pour qu'il expose les métriques sur un autre port ou chemin d'accès. Si vous modifiez le port ou le chemin, mettez à jour la `ecs_service_discovery` section dans le fichier de configuration de l' CloudWatch agent.

Installation de l'application d'exemple de proxy inverse NGINX Plus pour les clusters Amazon ECS

Suivez ces étapes pour installer l'application d'exemple de proxy inverse NGINX.

Création des images Docker

Pour créer les images Docker pour l'application d'exemple de proxy inverse NGINX Plus

1. Téléchargez le dossier suivant depuis le dépôt de proxy inverse NGINX : https://github.com/awslabs/ /tree/master/reverse-proxy/. ecs-nginx-reverse-proxy
2. Recherchez le répertoire `app` et créez une image à partir de ce répertoire :

   ```
   docker build -t web-server-app ./path-to-app-directory
   ```

3. Créez une image personnalisée pour NGINX Plus. Avant de pouvoir créer l'image pour NGINX Plus, vous devez obtenir la clé nommée `nginx-repo.key` et le certificat SSL `nginx-repo.crt` pour votre licence NGINX Plus. Créez un répertoire et stockez-les dans vos fichiers `nginx-repo.key` et `nginx-repo.crt`.

   Dans le répertoire que vous venez de créer, créez les deux fichiers suivants :

- Créez un exemple de fichier Docker incluant le contenu suivant. Ce fichier docker est adopté à partir d'un exemple de fichier fourni à l'adresse https://docs.nginx.com/nginx/admin-guide/installing-nginx/installing-nginx-docker/#docker_plus_image. Le changement important que nous faisons est que nous chargeons un fichier séparé, appelé `nginx.conf`, qui sera créé à l'étape suivante.

```
FROM debian:buster-slim

LABEL maintainer="NGINX Docker Maintainers <docker-maint@nginx.com>“

# Define NGINX versions for NGINX Plus and NGINX Plus modules
# Uncomment this block and the versioned nginxPackages block in the main RUN
# instruction to install a specific release
# ENV NGINX_VERSION 21
# ENV NJS_VERSION 0.3.9
# ENV PKG_RELEASE 1~buster

# Download certificate and key from the customer portal (https://cs.nginx.com
 (https://cs.nginx.com/))
# and copy to the build context
COPY nginx-repo.crt /etc/ssl/nginx/
COPY nginx-repo.key /etc/ssl/nginx/
# COPY nginx.conf /etc/ssl/nginx/nginx.conf

RUN set -x \
# Create nginx user/group first, to be consistent throughout Docker variants
&& addgroup --system --gid 101 nginx \
&& adduser --system --disabled-login --ingroup nginx --no-create-home --home /
nonexistent --gecos "nginx user" --shell /bin/false --uid 101 nginx \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y ca-
certificates gnupg1 \
&& \
NGINX_GPGKEY=573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62; \
found=''; \
for server in \
ha.pool.sks-keyservers.net (http://ha.pool.sks-keyservers.net/) \
hkp://keyserver.ubuntu.com:80 \
hkp://p80.pool.sks-keyservers.net:80 \
pgp.mit.edu (http://pgp.mit.edu/) \
; do \
echo "Fetching GPG key $NGINX_GPGKEY from $server"; \
```

```
apt-key adv --keyserver "$server" --keyserver-options timeout=10 --recv-keys
 "$NGINX_GPGKEY" && found=yes && break; \
done; \
test -z "$found" && echo >&2 "error: failed to fetch GPG key $NGINX_GPGKEY" &&
 exit 1; \
apt-get remove --purge --auto-remove -y gnupg1 && rm -rf /var/lib/apt/lists/* \
# Install the latest release of NGINX Plus and/or NGINX Plus modules
# Uncomment individual modules if necessary
# Use versioned packages over defaults to specify a release
&& nginxPackages=" \
nginx-plus \
# nginx-plus=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-xslt \
# nginx-plus-module-xslt=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-geoip \
# nginx-plus-module-geoip=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-image-filter \
# nginx-plus-module-image-filter=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-perl \
# nginx-plus-module-perl=${NGINX_VERSION}-${PKG_RELEASE} \
# nginx-plus-module-njs \
# nginx-plus-module-njs=${NGINX_VERSION}+${NJS_VERSION}-${PKG_RELEASE} \
" \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Peer \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::Verify-Host \"true\";" >> /etc/apt/
apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslCert \"/etc/ssl/nginx/nginx-
repo.crt\";" >> /etc/apt/apt.conf.d/90nginx \
&& echo "Acquire::https::plus-pkgs.nginx.com::SslKey \"/etc/ssl/nginx/nginx-
repo.key\";" >> /etc/apt/apt.conf.d/90nginx \
&& printf "deb https://plus-pkgs.nginx.com/debian buster nginx-plus\n" > /etc/
apt/sources.list.d/nginx-plus.list \
&& apt-get update \
&& apt-get install --no-install-recommends --no-install-suggests -y \
$nginxPackages \
gettext-base \
curl \
&& apt-get remove --purge --auto-remove -y && rm -rf /var/lib/apt/lists/* /etc/
apt/sources.list.d/nginx-plus.list \
&& rm -rf /etc/apt/apt.conf.d/90nginx /etc/ssl/nginx

# Forward request logs to Docker log collector
RUN ln -sf /dev/stdout /var/log/nginx/access.log \
```

```
&& ln -sf /dev/stderr /var/log/nginx/error.log

COPY nginx.conf /etc/nginx/nginx.conf

EXPOSE 80

STOPSIGNAL SIGTERM

CMD ["nginx", "-g", "daemon off;"]
```

- Un `nginx.conf` fichier, modifié depuis https://github.com/awslabs/ ecs-nginx-reverse-proxy / tree/master/reverse-proxy/nginx.

```
events {
  worker_connections 768;
}

http {
  # Nginx will handle gzip compression of responses from the app server
  gzip on;
  gzip_proxied any;
  gzip_types text/plain application/json;
  gzip_min_length 1000;

  upstream backend {
    zone name 10m;
    server app:3000    weight=2;
    server app2:3000    weight=1;
  }

  server{
    listen 8080;
    location /api {
      api write=on;
    }
  }

  match server_ok {
    status 100-599;
  }

  server {
    listen 80;
```

```
    status_zone zone;
    # Nginx will reject anything not matching /api
    location /api {
      # Reject requests with unsupported HTTP method
      if ($request_method !~ ^(GET|POST|HEAD|OPTIONS|PUT|DELETE)$) {
        return 405;
      }

      # Only requests matching the whitelist expectations will
      # get sent to the application server
      proxy_pass http://backend;
      health_check uri=/lorem-ipsum match=server_ok;
      proxy_http_version 1.1;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection 'upgrade';
      proxy_set_header Host $host;
      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_cache_bypass $http_upgrade;
    }
  }
}
```

4. Créez une image à partir des fichiers de votre nouveau répertoire :

```
docker build -t nginx-plus-reverse-proxy ./path-to-your-directory
```

5. Téléchargez vos nouvelles images dans un répertoire d'images pour une utilisation ultérieure.

Création de la définition de tâche pour exécuter NGINX Plus et l'application serveur web dans Amazon ECS

Ensuite, vous configurez la définition de tâche.

Cette définition de tâche permet la collecte et l'exportation des métriques NGINX Plus Prometheus. Le conteneur NGINX suit les entrées de l'application et expose ces données au port 8080, comme défini dans `nginx.conf`. Le conteneur d'exportation NGINX Prometheus récupère ces métriques et les publie sur le port 9113, pour les utiliser dans. CloudWatch

Pour configurer la définition de tâche pour l'application Amazon ECS d'exemple NGINX

1. Créez un fichier JSON de définition de tâche avec le contenu suivant. Remplacez *your-customized-nginx-plus-image* par l'URI d'image pour votre image NGINX Plus

personnalisée, et remplacez *your-web-server-app-image par l'URI d'image* pour l'image d'application de votre serveur Web.

```
{
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "your-customized-nginx-plus-image",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "links": [
        "app",
        "app2"
      ]
    },
    {
      "name": "app",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "app2",
      "image": "your-web-server-app-image",
      "memory": 256,
      "cpu": 128,
      "essential": true
    },
    {
      "name": "nginx-prometheus-exporter",
      "image": "docker.io/nginx/nginx-prometheus-exporter:0.8.0",
      "memory": 256,
      "cpu": 256,
      "essential": true,
      "command": [
```

```
        "-nginx.plus",
        "-nginx.scrape-uri",
         "http://nginx:8080/api"
    ],
    "links":[
      "nginx"
    ],
      "portMappings":[
        {
          "containerPort": 9113,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "networkMode": "bridge",
  "placementConstraints": [],
  "family": "nginx-plus-sample-stack"
}
```

2. Enregistrez la définition de tâche :

```
aws ecs register-task-definition --cli-input-json file://path-to-your-task-definition-json
```

3. Créez un service pour exécuter la tâche en saisissant la commande suivante :

```
aws ecs create-service \
 --cluster your-cluster-name \
 --service-name nginx-plus-service \
 --task-definition nginx-plus-sample-stack:1 \
 --desired-count 1
```

Veillez à ne pas modifier le nom du service. Nous exécuterons un service d' CloudWatch agent utilisant une configuration qui recherche les tâches en utilisant les modèles de noms des services qui les ont lancées. Par exemple, pour que l' CloudWatch agent trouve la tâche lancée par cette commande, vous pouvez spécifier la valeur de `sd_service_name_pattern` to be`^nginx-plus-service$`. La section suivante fournit plus de détails.

Configurer l' CloudWatch agent pour récupérer les métriques NGINX Plus Prometheus

La dernière étape consiste à configurer l' CloudWatch agent pour récupérer les métriques NGINX. Dans cet exemple, l' CloudWatch agent découvre la tâche via le modèle de nom de service et le port 9113, où l'exportateur expose les métriques prometheus pour NGINX. Une fois la tâche découverte et les métriques disponibles, l' CloudWatch agent commence à publier les métriques collectées dans le flux de log nginx-prometheus-exporter.

Pour configurer l' CloudWatch agent afin de récupérer les métriques NGINX

1. Téléchargez la dernière version du fichier YAML nécessaire en saisissant la commande suivante.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-bridge-host.yaml
```

2. Ouvrez le fichier dans un éditeur de texte et trouvez la configuration complète de l' CloudWatch agent dans la `value` clé de la section. `resource:CWAgentConfigSSMParameter` Ensuite, dans la section `ecs_service_discovery`, ajoutez la section `service_name_list_for_tasks` suivante.

```
"service_name_list_for_tasks": [
  {
    "sd_job_name": "nginx-plus-prometheus-exporter",
    "sd_metrics_path": "/metrics",
    "sd_metrics_ports": "9113",
    "sd_service_name_pattern": "^nginx-plus.*"
   }
],
```

3. Dans le même fichier, ajoutez la section suivante dans la section `metric_declaration` pour autoriser les métriques NGINX Plus. Veillez à suivre le modèle d'indentation existant.

```
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName"]],
  "metric_selectors": [
    "^nginxplus_connections_accepted$",
```

```
    "^nginxplus_connections_active$",
    "^nginxplus_connections_dropped$",
    "^nginxplus_connections_idle$",
    "^nginxplus_http_requests_total$",
    "^nginxplus_ssl_handshakes$",
    "^nginxplus_ssl_handshakes_failed$",
    "^nginxplus_up$",
    "^nginxplus_upstream_server_health_checks_fails$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName",
 "upstream"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_response_time$"
  ]
},
{
  "source_labels": ["job"],
  "label_matcher": "^nginx-plus.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily", "ServiceName", "code"]],
  "metric_selectors": [
    "^nginxplus_upstream_server_responses$",
    "^nginxplus_server_zone_responses$"
  ]
},
```

4. Si l' CloudWatch agent n'est pas déjà déployé dans ce cluster, passez à l'étape 8.

   Si l' CloudWatch agent est déjà déployé dans le cluster Amazon ECS à l'aide de AWS CloudFormation, vous pouvez créer un ensemble de modifications en saisissant les commandes suivantes :

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
```

```
aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name nginx-plus-scraping-support
```

5. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.

6. Passez en revue le nouvel ensemble de modifications. nginx-plus-scraping-support Vous devriez voir une modification appliquée à la ressource CW AgentConfig SSMParameter. Exécutez le changeset et redémarrez la tâche de l' CloudWatch agent en saisissant la commande suivante :

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Patientez environ 10 secondes, puis saisissez la commande suivante.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

8. Si vous installez l' CloudWatch agent avec la collecte de métriques Prometheus sur le cluster pour la première fois, entrez les commandes suivantes.

```
ECS_CLUSTER_NAME=your_cluster_name
AWS_REGION=your_aws_region
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
```

```
aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Affichage des journaux et métriques NGINX Plus

Vous pouvez maintenant afficher les métriques NGINX Plus collectées.

Pour examiner les métriques de votre application NGINX

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans la région où votre cluster s'exécute, choisissez Metrics (Métriques) dans le volet de navigation de gauche. Trouvez l'espace de noms ContainerInsights/Prometheus pour voir les métriques.
3. Pour voir les événements CloudWatch Logs, choisissez Log groups dans le volet de navigation. *Les événements se trouvent dans le groupe de journaux* ***/aws/containerinsights/*** *your_cluster_name /prometheus, dans le flux de journaux. nginx-plus-prometheus-exporter*

Didacticiel pour l'ajout d'une nouvelle cible de récupération Prometheus : Memcached sur Amazon ECS

Ce didacticiel propose une présentation pratique pour récupérer les métriques Prometheus d'un exemple d'application Memcached sur un cluster Amazon ECS avec le type de lancement EC2. La cible de l'exportateur Prometheus Memcached sera découverte automatiquement par l' CloudWatch agent par le biais de la découverte de services basée sur la définition des tâches ECS.

Memcached est un système de mise en cache de mémoire distribuée polyvalent. Il est souvent utilisé pour accélérer la vitesse des sites web dynamiques de base de données en mettant en cache les données et les objets dans la mémoire RAM pour réduire le nombre de fois où une source de

données externe (telle qu'une base de données ou une API) doit être lue. Pour en savoir plus, consultez Qu'est-ce que Memcached ?.

Le memchached_exporter (licence Apache 2.0) est l'un des exportateurs officiels de Prometheus. Par défaut, le memcache_exporter sert sur le port 0.0.0.0:9150 à l'adresse `/metrics`.

Les images Docker dans les deux référentiels Docker Hub suivants sont utilisées dans ce didacticiel :

- Memcached
- prom/memcached-exporter

Prérequis

Pour collecter des métriques à partir d'un exemple d'application Prometheus pour Amazon ECS, vous devez exécuter Container Insights dans le cluster. Pour plus d'informations sur l'installation de Container Insights, consultez Configuration de Container Insights sur Amazon ECS.

Rubriques

- Définition des variables d'environnement de cluster Amazon ECS EC2
- Installation de l'exemple d'application Memcached
- Configurer l' CloudWatch agent pour extraire les métriques Prometheus de Memcached
- Affichage de vos métriques Memcached

Définition des variables d'environnement de cluster Amazon ECS EC2

Pour définir des variables d'environnement de cluster Amazon ECS EC2

1. Installez l'interface de ligne de commande (CLI) Amazon ECS, si vous ne l'avez pas déjà fait. Pour en savoir plus, consultez Installation de l'interface de ligne de commande (CLI) Amazon ECS.
2. Définissez le nouveau nom de cluster Amazon ECS et la nouvelle région. Par exemple :

   ```
   ECS_CLUSTER_NAME=ecs-ec2-memcached-tutorial
   AWS_DEFAULT_REGION=ca-central-1
   ```

3. (Facultatif) Si vous ne possédez pas encore de cluster Amazon ECS de type de lancement EC2 dans lequel vous souhaitez installer l'exemple de charge de travail et d' CloudWatch agent Memcached, vous pouvez en créer un en saisissant la commande suivante.

```
ecs-cli up --capability-iam --size 1 \
--instance-type t3.medium \
--cluster $ECS_CLUSTER_NAME \
--region $AWS_REGION
```

Le résultat attendu de cette commande est le suivant :

```
WARN[0000] You will not be able to SSH into your EC2 instances without a key pair.
INFO[0000] Using recommended Amazon Linux 2 AMI with ECS Agent 1.44.4 and Docker
 version 19.03.6-ce
INFO[0001] Created cluster                               cluster=ecs-ec2-memcached-
tutorial region=ca-central-1
INFO[0002] Waiting for your cluster resources to be created...
INFO[0002] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0063] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
INFO[0124] Cloudformation stack status
 stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Security Group created: sg-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Installation de l'exemple d'application Memcached

Pour installer l'exemple d'application Memcached qui expose les métriques Prometheus

1. Téléchargez le AWS CloudFormation modèle Memcached en saisissant la commande suivante.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/memcached/memcached-traffic-sample.yaml
```

2. Définissez les noms de rôles IAM à créer pour Memcached en saisissant les commandes suivantes.

```
MEMCACHED_ECS_TASK_ROLE_NAME=memcached-prometheus-demo-ecs-task-role-name
```

```
MEMCACHED_ECS_EXECUTION_ROLE_NAME=memcached-prometheus-demo-ecs-execution-role-name
```

3. Installez l'exemple d'application Memcached en saisissant la commande suivante. Cet exemple installe l'application dans le mode réseau `host`.

```
MEMCACHED_ECS_NETWORK_MODE=host

aws cloudformation create-stack --stack-name Memcached-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-EC2-$MEMCACHED_ECS_NETWORK_MODE \
    --template-body file://memcached-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=ECSNetworkMode,ParameterValue=
$MEMCACHED_ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=
$MEMCACHED_ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$MEMCACHED_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

La AWS CloudFormation pile crée quatre ressources :

- Un rôle de tâche ECS
- Un rôle d'exécution de tâche ECS
- Une définition de tâche Memcached
- Un service Memcached

Dans la définition de la tâche Memcached, deux conteneurs sont définis :

- Le conteneur principal exécute une application Memcached simple et ouvre le port 11211 pour l'accès.
- L'autre conteneur exécute le processus d'exportation Redis pour exposer les métriques Prometheus sur le port 9150. Il s'agit du contenant à découvrir et à gratter par l' CloudWatch agent.

Configurer l' CloudWatch agent pour extraire les métriques Prometheus de Memcached

Pour configurer l' CloudWatch agent afin de récupérer les métriques Prometheus mises en cache par Memcached

1. Téléchargez la dernière version du fichier `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` en saisissant la commande suivante.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-awsvpc.yaml
   ```

2. Ouvrez le fichier dans un éditeur de texte et recherchez la configuration complète de l' CloudWatch agent située derrière la `value` clé dans la `resource:CWAgentConfigSSMParameter` section.

   Ensuite, dans la section `ecs_service_discovery`, ajoutez la configuration suivante dans la section `task_definition_list`.

   ```
   {
       "sd_job_name": "ecs-memcached",
       "sd_metrics_ports": "9150",
       "sd_task_definition_arn_pattern": ".*:task-definition/memcached-prometheus-demo.*:[0-9]+"
   },
   ```

   Pour la section `metric_declaration`, le paramètre par défaut n'autorise aucune métrique Memcached. Ajoutez la section suivante pour autoriser les métriques Memcached. Veillez à suivre le modèle d'indentation existant.

   ```
   {
     "source_labels": ["container_name"],
     "label_matcher": "memcached-exporter-.*",
     "dimensions": [["ClusterName", "TaskDefinitionFamily"]],
     "metric_selectors": [
       "^memcached_current_(bytes|items|connections)$",
       "^memcached_items_(reclaimed|evicted)_total$",
       "^memcached_(written|read)_bytes_total$",
       "^memcached_limit_bytes$",
       "^memcached_commands_total$"
   ```

```
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "memcached-exporter-.*",
  "dimensions": [["ClusterName", "TaskDefinitionFamily","status","command"],
 ["ClusterName", "TaskDefinitionFamily","command"]],
  "metric_selectors": [
    "^memcached_commands_total$"
  ]
},
```

3. Si l' CloudWatch agent est déjà déployé dans le cluster Amazon ECS par AWS CloudFormation, vous pouvez créer un ensemble de modifications en saisissant les commandes suivantes.

```
ECS_NETWORK_MODE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name memcached-scraping-support
```

4. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.

5. Vérifiez le jeu de modifications `memcached-scraping-support` nouvellement créé. Vous devriez voir une modification appliquée à la ressource `CWAgentConfigSSMParameter`. Exécutez l'ensemble de modifications et redémarrez la tâche de l' CloudWatch agent en saisissant les commandes suivantes.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
```

```
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

6. Patientez environ 10 secondes, puis saisissez la commande suivante.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-EC2-$ECS_NETWORK_MODE \
--region $AWS_REGION
```

7. Si vous installez l' CloudWatch agent avec la collecte de métriques Prometheus pour le cluster pour la première fois, entrez les commandes suivantes :

```
ECS_NETWORK_MODEE=bridge
CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION
```

Affichage de vos métriques Memcached

Ce didacticiel envoie les métriques suivantes à l'espace de noms ECS/ ContainerInsights / Prometheus dans. CloudWatch Vous pouvez utiliser la CloudWatch console pour voir les métriques de cet espace de noms.

Nom de la métrique	Dimensions	
`memcached _current_items`	`ClusterName` , `TaskDefinitionFamily`	

Nom de la métrique	Dimensions
memcached_current_connections	ClusterName , TaskDefinitionFamily
memcached_limit_bytes	ClusterName , TaskDefinitionFamily
memcached_current_bytes	ClusterName , TaskDefinitionFamily
memcached_written_bytes_total	ClusterName , TaskDefinitionFamily
memcached_read_bytes_total	ClusterName , TaskDefinitionFamily
memcached_items_evicted_total	ClusterName , TaskDefinitionFamily
memcached_items_reclaimed_total	ClusterName , TaskDefinitionFamily
memcached_commands_total	ClusterName , TaskDefinitionFamily ClusterName TaskDefinitionFamily, commande ClusterName TaskDefinitionFamily, statut, commande

> **Note**
>
> La valeur de la dimension command peut être : `delete`, `get`, `cas`, `set`, `decr`, `touch`, `incr` ou `flush`.
> La valeur de la dimension status peut être `hit`, `miss`, ou `badval`.

Vous pouvez également créer un CloudWatch tableau de bord pour vos métriques Prometheus Memcached.

Pour créer un tableau de bord pour les métriques Prometheus Memcached

1. Créez des variables d'environnement, en remplaçant les valeurs ci-dessous pour correspondre à votre déploiement.

```
DASHBOARD_NAME=your_memcached_cw_dashboard_name
ECS_TASK_DEF_FAMILY=memcached-prometheus-demo-$ECS_CLUSTER_NAME-EC2-
$MEMCACHED_ECS_NETWORK_MOD
```

2. Saisissez la commande suivante pour créer le tableau de bord.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-
container-insights/latest/ecs-task-definition-templates/deployment-mode/
replica-service/cwagent-prometheus/sample_cloudwatch_dashboards/memcached/
cw_dashboard_memcached.json \
| sed "s/{{YOUR_AWS_REGION}}/$AWS_REGION/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/$ECS_CLUSTER_NAME/g" \
| sed "s/{{YOUR_TASK_DEF_FAMILY}}/$ECS_TASK_DEF_FAMILY/g" \
| xargs -0 aws cloudwatch put-dashboard --dashboard-name ${DASHBOARD_NAME} --region
 $AWS_REGION --dashboard-body
```

Didacticiel pour récupérer les métriques Redis Prometheus sur Amazon ECS Fargate

Ce didacticiel propose une présentation pratique de l'utilisation des métriques Prometheus d'un exemple d'application Redis dans un cluster Amazon ECS Fargate. La cible de l'exportateur Redis Prometheus sera découverte automatiquement par l' CloudWatch agent avec le support métrique Prometheus sur la base des étiquettes docker du conteneur.

Redis (https://redis.io/) est un magasin de structures de données open source (sous licence BSD) en mémoire, utilisé comme base de données, cache et agent de messages. Pour en savoir plus, consultez redis.

redis_exportateur (sous licence MIT) est utilisé pour exposer les métriques Redis Prometheus sur le port spécifié (par défaut : 0.0.0.0:9121). Pour en savoir plus, consultez redis_exporter.

Les images Docker dans les deux référentiels Docker Hub suivants sont utilisées dans ce didacticiel :

- Redis
- redis_exporter

Prérequis

Pour collecter des métriques à partir d'un exemple d'application Prometheus pour Amazon ECS, vous devez exécuter Container Insights dans le cluster. Pour plus d'informations sur l'installation de Container Insights, consultez Configuration de Container Insights sur Amazon ECS.

Rubriques

- Définition de la variable d'environnement de cluster Amazon ECS Fargate
- Définition des variables d'environnement réseau pour le cluster Amazon ECS Fargate
- Installez l'exemple de charge de travail Redis
- Configurer l' CloudWatch agent pour récupérer les métriques Redis Prometheus
- Affichage de vos métriques Redis

Définition de la variable d'environnement de cluster Amazon ECS Fargate

Pour définir la variable d'environnement de cluster Amazon ECS Fargate

1. Installez l'interface de ligne de commande (CLI) Amazon ECS, si vous ne l'avez pas déjà fait. Pour en savoir plus, consultez Installation de l'interface de ligne de commande (CLI) Amazon ECS.
2. Définissez le nouveau nom de cluster Amazon ECS et la nouvelle région. Par exemple :

```
ECS_CLUSTER_NAME=ecs-fargate-redis-tutorial
AWS_DEFAULT_REGION=ca-central-1
```

3. (Facultatif) Si vous ne possédez pas encore de cluster Amazon ECS Fargate dans lequel vous souhaitez installer l'exemple de charge de travail CloudWatch et d'agent Redis, vous pouvez en créer un en saisissant la commande suivante.

```
ecs-cli up --capability-iam \
--cluster $ECS_CLUSTER_NAME \
--launch-type FARGATE \
--region $AWS_DEFAULT_REGION
```

Le résultat attendu de cette commande est le suivant :

```
INFO[0000] Created cluster   cluster=ecs-fargate-redis-tutorial region=ca-central-1
INFO[0001] Waiting for your cluster resources to be created...
INFO[0001] Cloudformation stack status   stackStatus=CREATE_IN_PROGRESS
VPC created: vpc-xxxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxx
Subnet created: subnet-xxxxxxxxxxxxxxxx
Cluster creation succeeded.
```

Définition des variables d'environnement réseau pour le cluster Amazon ECS Fargate

Pour définir les variables d'environnement réseau pour le cluster Amazon ECS Fargate

1. Définissez votre VPC et votre ID de sous-réseau du cluster Amazon ECS. Si vous avez créé un nouveau cluster dans la procédure précédente, ces valeurs s'affichent dans le résultat de la commande finale. Sinon, utilisez les ID du cluster existant que vous allez utiliser avec Redis.

```
ECS_CLUSTER_VPC=vpc-xxxxxxxxxxxxxxxxx
ECS_CLUSTER_SUBNET_1=subnet-xxxxxxxxxxxxxxxx
ECS_CLUSTER_SUBNET_2=subnet-xxxxxxxxxxxxxxxx
```

2. Dans ce didacticiel, nous allons installer l'application Redis et l' CloudWatchagent dans le groupe de sécurité par défaut du VPC du cluster Amazon ECS. Le groupe de sécurité par défaut autorise toutes les connexions réseau au sein du même groupe de sécurité afin que l' CloudWatch agent puisse supprimer les métriques Prometheus exposées sur les conteneurs Redis. Dans un environnement de production réel, vous souhaiterez peut-être créer des groupes de sécurité dédiés pour l'application et l' CloudWatch agent Redis et définir des autorisations personnalisées pour eux.

Saisissez la commande suivante pour obtenir l'ID du groupe de sécurité par défaut.

```
aws ec2 describe-security-groups \
--filters Name=vpc-id,Values=$ECS_CLUSTER_VPC  \
--region $AWS_DEFAULT_REGION
```

Définissez ensuite la variable de groupe de sécurité par défaut du cluster Fargate en saisissant la commande suivante, en la *my-default-security-group*remplaçant par la valeur que vous avez trouvée dans la commande précédente.

```
ECS_CLUSTER_SECURITY_GROUP=my-default-security-group
```

Installez l'exemple de charge de travail Redis

Pour installer l'exemple d'application Redis qui expose les métriques Prometheus

1. Téléchargez le AWS CloudFormation modèle Redis en saisissant la commande suivante.

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
cwagent-prometheus/sample_traffic/redis/redis-traffic-sample.yaml
```

2. Définissez les noms de rôles IAM à créer pour Redis en saisissant les commandes suivantes.

```
REDIS_ECS_TASK_ROLE_NAME=redis-prometheus-demo-ecs-task-role-name
REDIS_ECS_EXECUTION_ROLE_NAME=redis-prometheus-demo-ecs-execution-role-name
```

3. Installez l'exemple d'application Redis en saisissant la commande suivante.

```
aws cloudformation create-stack --stack-name Redis-Prometheus-Demo-ECS-
$ECS_CLUSTER_NAME-fargate-awsvpc \
    --template-body file://redis-traffic-sample.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET_1 \
                 ParameterKey=TaskRoleName,ParameterValue=$REDIS_ECS_TASK_ROLE_NAME
 \
```

```
                  ParameterKey=ExecutionRoleName,ParameterValue=
$REDIS_ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_DEFAULT_REGION
```

La AWS CloudFormation pile crée quatre ressources :

- Un rôle de tâche ECS
- Un rôle d'exécution de tâche ECS
- Une définition de tâche Redis
- Un service Redis

Dans la définition de la tâche Redis, deux conteneurs sont définis :

- Le conteneur principal exécute une application Redis simple et ouvre le port 6379 pour l'accès.
- L'autre conteneur exécute le processus d'exportation Redis pour exposer les métriques Prometheus sur le port 9121. Il s'agit du contenant à découvrir et à gratter par l' CloudWatch agent. L'étiquette docker suivante est définie afin que l' CloudWatch agent puisse découvrir ce conteneur en fonction de celle-ci.

```
ECS_PROMETHEUS_EXPORTER_PORT: 9121
```

Configurer l' CloudWatch agent pour récupérer les métriques Redis Prometheus

Pour configurer l' CloudWatch agent afin de récupérer les métriques Redis Prometheus

1. Téléchargez la dernière version du fichier `cwagent-ecs-prometheus-metric-for-awsvpc.yaml` en saisissant la commande suivante.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/ecs-task-definition-templates/deployment-mode/replica-service/
   cwagent-prometheus/cloudformation-quickstart/cwagent-ecs-prometheus-metric-for-
   awsvpc.yaml
   ```

2. Ouvrez le fichier dans un éditeur de texte et recherchez la configuration complète de l' CloudWatch agent située derrière la `value` clé dans la `resource:CWAgentConfigSSMParameter` section.

Ensuite, dans la section `ecs_service_discovery` présentée ici, la découverte de service basée sur `docker_label` est activée avec les paramètres par défaut qui sont basés sur `ECS_PROMETHEUS_EXPORTER_PORT`, qui correspond à l'étiquette docker que nous avons définie dans la définition de tâche Redis ECS. Nous n'avons donc pas besoin d'apporter des modifications dans cette section :

```
ecs_service_discovery": {
  "sd_frequency": "1m",
  "sd_result_file": "/tmp/cwagent_ecs_auto_sd.yaml",
*  "docker_label": {
  },*
  ...
```

Pour la section `metric_declaration`, le paramètre par défaut n'autorise aucune métrique Redis. Ajoutez la section suivante pour autoriser les métriques Redis. Veillez à suivre le modèle d'indentation existant.

```
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily"]],
  "metric_selectors": [
    "^redis_net_(in|out)put_bytes_total$",
    "^redis_(expired|evicted)_keys_total$",
    "^redis_keyspace_(hits|misses)_total$",
    "^redis_memory_used_bytes$",
    "^redis_connected_clients$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["container_name"],
  "label_matcher": "^redis-exporter-.*$",
  "dimensions": [["ClusterName","TaskDefinitionFamily","db"]],
```

```
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

3. Si l' CloudWatch agent est déjà déployé dans le cluster Amazon ECS par AWS CloudFormation, vous pouvez créer un ensemble de modifications en saisissant les commandes suivantes.

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION} \
    --change-set-name redis-scraping-support
```

4. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.

5. Vérifiez le jeu de modifications `redis-scraping-support` nouvellement créé. Vous devriez voir une modification appliquée à la ressource `CWAgentConfigSSMParameter`. Exécutez l'ensemble de modifications et redémarrez la tâche de l' CloudWatch agent en saisissant les commandes suivantes.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

6. Patientez environ 10 secondes, puis saisissez la commande suivante.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service cwagent-prometheus-replica-service-$ECS_LAUNCH_TYPE-awsvpc \
--region ${AWS_DEFAULT_REGION}
```

7. Si vous installez l' CloudWatch agent avec la collecte de métriques Prometheus pour le cluster pour la première fois, entrez les commandes suivantes :

```
ECS_LAUNCH_TYPE=FARGATE
CREATE_IAM_ROLES=True
ECS_CLUSTER_SUBNET=$ECS_CLUSTER_SUBNET_1
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-stack --stack-name CWAgent-Prometheus-ECS-
$ECS_CLUSTER_NAME-$ECS_LAUNCH_TYPE-awsvpc \
    --template-body file://cwagent-ecs-prometheus-metric-for-awsvpc.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSLaunchType,ParameterValue=$ECS_LAUNCH_TYPE \
                 ParameterKey=SecurityGroupID,ParameterValue=
$ECS_CLUSTER_SECURITY_GROUP \
                 ParameterKey=SubnetID,ParameterValue=$ECS_CLUSTER_SUBNET \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=
$ECS_EXECUTION_ROLE_NAME \
    --capabilities CAPABILITY_NAMED_IAM \
    --region ${AWS_DEFAULT_REGION}
```

Affichage de vos métriques Redis

Ce didacticiel envoie les métriques suivantes à l'espace de noms ECS/ ContainerInsights / Prometheus dans. CloudWatch Vous pouvez utiliser la CloudWatch console pour voir les métriques de cet espace de noms.

Nom de la métrique	Dimensions	
redis_net_input_bytes_total	ClusterName, TaskDefinitionFamily	
redis_net_output_bytes_total	ClusterName, TaskDefinitionFamily	
redis_expired_keys_total	ClusterName, TaskDefinitionFamily	
redis_evicted_keys_total	ClusterName, TaskDefinitionFamily	
redis_keyspace_hits_total	ClusterName, TaskDefinitionFamily	
redis_keyspace_misses_total	ClusterName, TaskDefinitionFamily	
redis_memory_used_bytes	ClusterName, TaskDefinitionFamily	
redis_connected_clients	ClusterName, TaskDefinitionFamily	
redis_commands_total	ClusterName , TaskDefinitionFamily , cmd	
redis_db_keys	ClusterName , TaskDefinitionFamily , db	

> **Note**
>
> Les valeurs de la dimension cmd peuvent être : `append`, `client`, `command`, `config`, `dbsize`, `flushall`, `get`, `incr`, `info`, `latency` ou `slowlog`.
> Les valeurs de la dimension db peuvent être `db0` ou `db15`.

Vous pouvez également créer un CloudWatch tableau de bord pour vos métriques Redis Prometheus.

Pour créer un tableau de bord pour les métriques Prometheus Redis

1. Créez des variables d'environnement, en remplaçant les valeurs ci-dessous pour correspondre à votre déploiement.

```
DASHBOARD_NAME=your_cw_dashboard_name
ECS_TASK_DEF_FAMILY=redis-prometheus-demo-$ECS_CLUSTER_NAME-fargate-awsvpc
```

2. Saisissez la commande suivante pour créer le tableau de bord.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Configuration de la collecte de métriques Prometheus sur des clusters Amazon EKS et Kubernetes

Pour collecter les métriques Prometheus à partir de clusters exécutant Amazon EKS ou Kubernetes, vous pouvez utiliser CloudWatch l'agent comme collecteur ou utiliser Distro pour collecteur. AWS OpenTelemetry Pour plus d'informations sur l'utilisation de AWS Distro for OpenTelemetry Collector, consultez https://aws-otel.github.io/docs/getting-started/container-insights/eks-prometheus.

Les sections suivantes expliquent comment collecter les métriques Prometheus à l'aide de l'agent. CloudWatch Ils expliquent comment installer l' CloudWatch agent avec Prometheus monitoring sur des clusters exécutant Amazon EKS ou Kubernetes, et comment configurer l'agent pour récupérer

des cibles supplémentaires. Elles fournissent également des didacticiels facultatifs pour configurer des exemples d'applications à utiliser pour les tests avec la surveillance Prometheus.

Rubriques

Installez l' CloudWatch agent avec la collecte de métriques Prometheus sur les clusters Amazon EKS et Kubernetes

Cette section explique comment configurer l' CloudWatch agent avec la surveillance Prometheus dans un cluster exécutant Amazon EKS ou Kubernetes. Après cela, l'agent récupère et importe automatiquement les métriques pour les applications suivantes exécutées dans ce cluster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

Vous pouvez également configurer l'agent pour récupérer et importer à partir d'applications et sources Prometheus supplémentaires.

Avant de suivre ces étapes pour installer l' CloudWatch agent de collecte de métriques Prometheus, vous devez disposer d'un cluster exécuté sur Amazon EKS ou d'un cluster Kubernetes exécuté sur une instance Amazon EC2.

Exigences de groupe de sécurité VPC

Les règles d'entrée des groupes de sécurité pour les charges de travail Prometheus doivent ouvrir les ports Prometheus à l'agent pour récupérer les métriques Prometheus par CloudWatch l'adresse IP privée.

Les règles de sortie du groupe de sécurité pour l' CloudWatch agent doivent permettre à l'agent de se connecter au CloudWatch port des charges de travail Prometheus via une adresse IP privée.

Rubriques

- Installez l' CloudWatch agent avec la collecte de métriques Prometheus sur les clusters Amazon EKS et Kubernetes
- Récupération de sources Prometheus supplémentaires et importation de ces métriques
- (En option) Configuration d'exemples d'applications Amazon EKS conteneurisées pour les test de métriques Prometheus

Installez l' CloudWatch agent avec la collecte de métriques Prometheus sur les clusters Amazon EKS et Kubernetes

Cette section explique comment configurer l' CloudWatch agent avec la surveillance Prometheus dans un cluster exécutant Amazon EKS ou Kubernetes. Après cela, l'agent récupère et importe automatiquement les métriques pour les applications suivantes exécutées dans ce cluster.

- AWS App Mesh
- NGINX
- Memcached
- Java/JMX
- HAProxy
- Fluent Bit

Vous pouvez également configurer l'agent pour récupérer et importer à partir d'applications et sources Prometheus supplémentaires.

Avant de suivre ces étapes pour installer l' CloudWatch agent de collecte de métriques Prometheus, vous devez disposer d'un cluster exécuté sur Amazon EKS ou d'un cluster Kubernetes exécuté sur une instance Amazon EC2.

Exigences de groupe de sécurité VPC

Les règles d'entrée des groupes de sécurité pour les charges de travail Prometheus doivent ouvrir les ports Prometheus à l'agent pour récupérer les métriques Prometheus par CloudWatch l'adresse IP privée.

Les règles de sortie du groupe de sécurité pour l' CloudWatch agent doivent permettre à l'agent de se connecter au CloudWatch port des charges de travail Prometheus via une adresse IP privée.

Rubriques

- Configuration de rôles IAM
- Installation de l' CloudWatchagent pour collecter les métriques Prometheus

Configuration de rôles IAM

La première étape consiste à configurer le rôle IAM nécessaire dans le cluster. Il existe deux méthodes :

- Configurez un rôle IAM pour un compte de service, également appelé fonction du service. Cette méthode fonctionne à la fois pour le type de lancement EC2 et le type de lancement Fargate.
- Ajouter une politique IAM au rôle IAM utilisé pour le cluster. Cela ne fonctionne que pour le type de lancement EC2.

Configurer une fonction du service (type de lancement EC2 et type de lancement Fargate)

Pour configurer une fonction du service, saisissez la commande suivante. Remplacez *MyCluster*par le nom du cluster.

```
eksctl create iamserviceaccount \
 --name cwagent-prometheus \
--namespace amazon-cloudwatch \
 --cluster MyCluster \
--attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
--approve \
--override-existing-serviceaccounts
```

Ajouter une politique au rôle IAM du cluster (type de lancement EC2 uniquement)

Pour configurer la politique IAM dans un cluster pour la prise en charge de Prometheus

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans le panneau de navigation, sélectionnez Instances.
3. Vous devez trouver le préfixe du nom du rôle IAM pour le cluster. Pour ce faire, cochez la case en regard du nom d'une instance qui se trouve dans le cluster et choisissez Actions, Paramètres de l'instance, Attacher/Remplacer des rôles IAM. Copiez ensuite le préfixe du rôle IAM, par exemple `eksctl-dev303-workshop-nodegroup`.
4. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
5. Dans le panneau de navigation, sélectionnez Rôles.

6. Utilisez la zone de recherche pour rechercher le préfixe que vous avez copié précédemment au cours de cette procédure et choisissez ce rôle.
7. Choisissez Attach Policies (Attacher des politiques).
8. Utilisez le champ de recherche pour trouver CloudWatchAgentServerPolicy. Cochez la case à côté de CloudWatchAgentServerPolicy, puis choisissez Attacher une politique.

Installation de l' CloudWatchagent pour collecter les métriques Prometheus

Vous devez installer l' CloudWatch agent dans le cluster pour collecter les métriques. La méthode à appliquer pour installer l'agent est différente s'il s'agit de clusters Amazon EKS ou de clusters Kubernetes.

Supprimer les versions précédentes de l' CloudWatch agent avec le support de Prometheus

Si vous avez déjà installé une version de l' CloudWatch agent compatible avec Prometheus dans votre cluster, vous devez supprimer cette version en saisissant la commande suivante. Cette étape est nécessaire uniquement pour les versions précédentes de l'agent avec prise en charge de Prometheus. Il n'est pas nécessaire de supprimer l' CloudWatch agent qui active Container Insights sans l'assistance de Prometheus.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

Installation de l' CloudWatch agent sur des clusters Amazon EKS avec le type de lancement EC2

Pour installer l' CloudWatch agent compatible avec Prometheus sur un cluster Amazon EKS, procédez comme suit.

Pour installer l' CloudWatch agent avec le support de Prometheus sur un cluster Amazon EKS

1. Entrez la commande suivante pour vérifier si l'espace de noms `amazon-cloudwatch` a déjà été créé :

   ```
   kubectl get namespace
   ```

2. Si le fichier `amazon-cloudwatch` n'est pas affiché dans les résultats, créez-le en entrant la commande suivante :

   ```
   kubectl create namespace amazon-cloudwatch
   ```

3. Pour déployer l'agent avec la configuration par défaut et lui demander d'envoyer des données à la AWS région dans laquelle il est installé, entrez la commande suivante :

```
kubectl apply -f https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
```

   En revanche; pour que l'agent envoie des données à une autre région, procédez comme suit :

   a. Téléchargez le fichier YAML de l'agent en entrant la commande suivante :

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
```

   b. Ouvrez le fichier avec un éditeur de texte et recherchez le bloc `cwagentconfig.json` du fichier.

   c. Ajoutez les lignes en surbrillance, en spécifiant la région souhaitée :

```
cwagentconfig.json: |
    {
      "agent": {
        "region": "us-east-2"
      },
      "logs": { ...
```

   d. Enregistrez le fichier et déployez l'agent à l'aide de votre fichier mis à jour.

```
kubectl apply -f prometheus-eks.yaml
```

Installation de l' CloudWatch agent sur des clusters Amazon EKS avec le type de lancement Fargate

Pour installer l' CloudWatch agent compatible avec Prometheus sur un cluster Amazon EKS avec le type de lancement Fargate, procédez comme suit.

Pour installer l' CloudWatch agent compatible avec Prometheus sur un cluster Amazon EKS avec le type de lancement Fargate

1. Entrez la commande suivante pour créer un profil Fargate pour CloudWatch l'agent afin qu'il puisse s'exécuter dans le cluster. Remplacez *`MyCluster`*par le nom du cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--name amazon-cloudwatch \
--namespace amazon-cloudwatch
```

2. Pour installer l' CloudWatch agent, entrez la commande suivante. Remplacez *MyCluster*par le nom du cluster. Ce nom est utilisé dans le nom du groupe de journaux qui stocke les événements de journaux collectés par l'agent et est également utilisé comme dimension pour les métriques collectées par l'agent.

   Remplacez *region* par le nom de la région dans laquelle vous souhaitez envoyer les métriques. Par exemple, `us-west-1`.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
   prometheus/prometheus-eks-fargate.yaml |
   sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
   kubectl apply -f -
   ```

Installation de l' CloudWatch agent sur un cluster Kubernetes

Pour installer l' CloudWatch agent compatible avec Prometheus sur un cluster exécutant Kubernetes, entrez la commande suivante :

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/prometheus-k8s.yaml |
sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" |
kubectl apply -f -
```

Remplacez *MyCluster*par le nom du cluster. Ce nom est utilisé dans le nom du groupe de journaux qui stocke les événements de journaux collectés par l'agent et est également utilisé comme dimension pour les métriques collectées par l'agent.

Remplacez *la région* par le nom de la AWS région dans laquelle vous souhaitez que les métriques soient envoyées. Par exemple, **`us-west-1`**.

Vérification de l'exécution de l'agent

Sur les clusters Amazon EKS et Kubernetes, vous pouvez entrer la commande suivante pour vérifier que l'agent est en cours d'exécution.

```
kubectl get pod -l "app=cwagent-prometheus" -n amazon-cloudwatch
```

Si les résultats incluent un module d' CloudWatch agent unique dans l'`Running`état, l'agent est en cours d'exécution et collecte les métriques Prometheus. Par défaut, l' CloudWatchagent collecte des métriques pour App Mesh, NGINX, Memcached, Java/JMX et HAProxy chaque minute. Pour plus d'informations sur ces métriques, consultez Métriques Prometheus collectées par l'agent CloudWatch . Pour obtenir des instructions sur la façon d'afficher vos statistiques Prometheus dans, voir CloudWatch Affichage de vos métriques Prometheus

Vous pouvez également configurer l' CloudWatch agent pour collecter des métriques auprès d'autres exportateurs Prometheus. Pour plus d'informations, consultez Récupération de sources Prometheus supplémentaires et importation de ces métriques.

Récupération de sources Prometheus supplémentaires et importation de ces métriques

L' CloudWatch agent chargé de surveiller Prometheus a besoin de deux configurations pour récupérer les métriques Prometheus. L'une concerne les configurations standard Prometheus, comme décrit dans <scrape_config> dans la documentation Prometheus. L'autre concerne la configuration de l' CloudWatch agent.

Pour les clusters Amazon EKS, les configurations sont définies dans `prometheus-eks.yaml` (pour le type de lancement EC2) ou`prometheus-eks-fargate.yaml` (pour le type de lancement Fargate) sous la forme de deux cartes de configuration :

- La section `name: prometheus-config` contient les paramètres pour la récupération Prometheus.
- La `name: prometheus-cwagentconfig` section contient la configuration de l' CloudWatch agent. Vous pouvez utiliser cette section pour configurer la manière dont les métriques Prometheus sont collectées par. CloudWatch Par exemple, vous spécifiez les métriques dans lesquelles vous souhaitez CloudWatch importer et définissez leurs dimensions.

Pour les clusters Kubernetes s'exécutant sur des instances Amazon EC2, les configurations sont définies dans le fichier YAML `prometheus-k8s.yaml` sous la forme de deux cartes de configuration :

- La section `name: prometheus-config` contient les paramètres pour la récupération Prometheus.
- La `name: prometheus-cwagentconfig` section contient la configuration de l' CloudWatch agent.

Pour extraire des sources de métriques Prometheus supplémentaires et les importer, CloudWatch vous devez modifier à la fois la configuration de Prometheus Scrape et la configuration de l'agent, puis redéployer CloudWatch l'agent avec la configuration mise à jour.

Exigences de groupe de sécurité VPC

Les règles d'entrée des groupes de sécurité pour les charges de travail Prometheus doivent ouvrir les ports Prometheus à l'agent pour récupérer les métriques Prometheus par CloudWatch l'adresse IP privée.

Les règles de sortie du groupe de sécurité pour l' CloudWatch agent doivent permettre à l'agent de se connecter au CloudWatch port des charges de travail Prometheus via une adresse IP privée.

Configuration de récupération Prometheus

L' CloudWatch agent prend en charge les configurations standard de Prometheus scrape, comme indiquéhttps://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config <scrape_config>dans la documentation de Prometheus. Vous pouvez modifier cette section pour mettre à jour les configurations déjà présentes dans ce fichier et ajouter des cibles de récupération Prometheus supplémentaires. Par défaut, l'exemple de fichier de configuration contient les lignes de configuration globale suivantes :

```
global:
  scrape_interval: 1m
  scrape_timeout: 10s
```

- scrape_interval– Définit la fréquence à laquelle récupérer les cibles.
- scrape_timeout– Définit le temps d'attente avant l'expiration d'une requête de récupération.

Vous pouvez également définir différentes valeurs pour ces paramètres au niveau de la tâche, afin de remplacer les configurations globales.

Tâches de récupération Prometheus

Certaines tâches de scraping par défaut sont déjà configurées dans les fichiers YAML de l' CloudWatch agent. Par exemple, dans `prometheus-eks.yaml`, les tâches de récupération par défaut sont configurées dans les lignes `job_name` de la section `scrape_configs`. Dans ce fichier, la section `kubernetes-pod-jmx` par défaut suivante recoupe les métriques de JMX Exporter.

```
- job_name: 'kubernetes-pod-jmx'
   sample_limit: 10000
   metrics_path: /metrics
   kubernetes_sd_configs:
   - role: pod
   relabel_configs:
   - source_labels: [__address__]
     action: keep
     regex: '.*:9404$'
   - action: labelmap
     regex: __meta_kubernetes_pod_label_(.+)
   - action: replace
     source_labels:
     - __meta_kubernetes_namespace
     target_label: Namespace
   - source_labels: [__meta_kubernetes_pod_name]
     action: replace
     target_label: pod_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_container_name
     target_label: container_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_controller_name
     target_label: pod_controller_name
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_controller_kind
     target_label: pod_controller_kind
   - action: replace
     source_labels:
     - __meta_kubernetes_pod_phase
     target_label: pod_phase
```

Chacune de ces cibles par défaut est supprimée et les métriques sont envoyées dans le journal des événements CloudWatch à l'aide d'un format de métrique intégré. Pour plus d'informations, consultez Intégration de métriques dans les journaux.

Les événements de journal des clusters Amazon EKS et Kubernetes sont stockés dans le groupe de journaux ***/aws/containerinsights/ cluster_name / prometheus*** *dans Logs.* CloudWatch Les évènements de journaux des clusters Amazon ECS sont stockés dans le groupe de journaux /aws/ecs/containerinsights/***cluster_name***/prometheus.

Chaque tâche de récupération est contenue dans un flux de journaux différent au sein de ce groupe de journaux. Par exemple, la tâche de récupération Prometheus `kubernetes-pod-appmesh-envoy` est définie pour App Mesh. ***Toutes les métriques App Mesh Prometheus provenant des clusters Amazon EKS et Kubernetes sont envoyées au flux de journal nommé /aws/containerinsights/ cluster_name >prometheus//.*** kubernetes-pod-appmesh-envoy

Pour ajouter une nouvelle cible de récupération, vous ajoutez une nouvelle section `job_name` à la section `scrape_configs` du fichier YAML et redémarrez l'agent. Pour un exemple de ce processus, consultez Didacticiel pour l'ajout d'une nouvelle cible de récupération Prometheus : métrique du serveur d'API Prometheus.

CloudWatch configuration de l'agent pour Prometheus

Le fichier de configuration de l' CloudWatch agent contient une `prometheus` section `metrics_collected` dédiée à la configuration du scraping de Prometheus. Elle inclut les options de configuration suivantes :

- cluster_name– Spécifie le nom du cluster à ajouter en tant qu'étiquette dans l'évènement du journal. Ce champ est facultatif. Si vous l'omettez, l'agent peut détecter le nom du cluster Amazon EKS ou Kubernetes.
- log_group_name– Spécifie le nom du groupe de journaux pour les métriques Prometheus récupérées. Ce champ est facultatif. Si vous l'omettez, utilisez CloudWatch /aws/containerinsights/***cluster_name /prometheus pour les journaux des clusters*** Amazon EKS et Kubernetes.
- prometheus_config_path– Spécifie le chemin d'accès du fichier de configuration de récupération Prometheus. Si la valeur de ce champ commence par `env:`, le contenu du fichier de configuration de récupération Prometheus sera récupéré à partir de la variable d'environnement du conteneur. Ne modifiez pas ce champ.

- ecs_service_discovery– Il s'agit de la section qui spécifie la configuration de la découverte de service Amazon ECS Prometheus. Pour plus d'informations, consultez . Guide détaillé de la découverte automatique sur les clusters Amazon ECS.

  La section `ecs_service_discovery` peut contenir les champs suivants :

  - `sd_frequency` est la fréquence de découverte des exportateurs Prometheus. Spécifiez un nombre et un suffixe d'unité. Par exemple, `1m` pour une fois par minute ou `30s` pour une fois toutes les 30 secondes. Les suffixes d'unités valides sont `ns`, `us`, `ms`, `s`, `m` et `h`.

    Ce champ est facultatif. La valeur par défaut est de 60 secondes (1 minute).
  - `sd_target_cluster` est le nom du cluster Amazon ECS cible pour la découverte automatique. Ce champ est facultatif. Le nom par défaut est le nom du cluster Amazon ECS sur lequel l' CloudWatch agent est installé.
  - `sd_cluster_region` est la région du cluster Amazon ECS cible. Ce champ est facultatif. La valeur par défaut est la région du cluster Amazon ECS dans laquelle l' CloudWatch agent est installé.
  - `sd_result_file` est le chemin d'accès du fichier YAML pour les résultats de la cible Prometheus. La configuration de récupération Prometheus fera référence à ce fichier.
  - `docker_label` est une section facultative que vous pouvez utiliser pour spécifier la configuration de la découverte de service basée sur des étiquettes de docker. Si vous omettez cette section, la découverte basée sur les étiquettes docker n'est pas utilisée. Cette section peut contenir les champs suivants :
    - `sd_port_label` est le nom de l'étiquette docker du conteneur qui spécifie le port du conteneur pour les métriques Prometheus. La valeur par défaut est `ECS_PROMETHEUS_EXPORTER_PORT`. Si le conteneur ne possède pas cette étiquette docker, l' CloudWatch agent l'ignorera.
    - `sd_metrics_path_label` est le nom de l'étiquette docker du conteneur qui spécifie le chemin d'accès aux métriques Prometheus. La valeur par défaut est `ECS_PROMETHEUS_METRICS_PATH`. Si le conteneur n'a pas cette étiquette docker, l'agent utilise le chemin par défaut `/metrics`.
    - `sd_job_name_label` est le nom de l'étiquette docker du conteneur qui spécifie le nom de la tâche de récupération Prometheus. La valeur par défaut est `job`. Si le conteneur ne possède pas cette étiquette docker, l' CloudWatch agent utilise le nom de la tâche dans la configuration Prometheus Scrape.

- `task_definition_list` est une section facultative que vous pouvez utiliser pour spécifier la configuration de la découverte de service basée sur les définitions de tâches. Si vous omettez cette section, la découverte basée sur les définitions de tâches n'est pas utilisée. Cette section peut contenir les champs suivants :
  - `sd_task_definition_arn_pattern` est le modèle à utiliser pour spécifier les définitions de tâches Amazon ECS à découvrir. Il s'agit d'une expression régulière.
  - `sd_metrics_ports` répertorie le containerPort pour les métriques Prometheus. Séparez les containerPorts par des points-virgules.
  - `sd_container_name_pattern` spécifie les noms des conteneurs de tâches Amazon ECS. Il s'agit d'une expression régulière.
  - `sd_metrics_path` spécifie le chemin de métrique Prometheus. Si vous ne spécifiez pas ce paramètre, l'agent utilise le chemin par défaut `/metrics`
  - `sd_job_name` spécifie le nom de la tâche de récupération Prometheus. Si vous omettez ce champ, l' CloudWatch agent utilise le nom de la tâche dans la configuration Prometheus Scrape.
- metric_declaration – Ce sont des sections qui spécifient le tableau de journaux avec le format de métrique intégré à générer. Il existe des `metric_declaration` sections pour chaque source Prometheus à partir de laquelle CloudWatch l'agent importe par défaut. Chacune de ces sections comprend les champs suivants :
  - `label_matcher` est une expression régulière qui vérifie la valeur des étiquettes répertoriées dans `source_labels`. Les métriques correspondantes sont activées pour être incluses dans le format de métrique intégré envoyé à CloudWatch.

    Si plusieurs étiquettes sont spécifiées dans `source_labels`, nous vous recommandons de ne pas utiliser les caractères ^ ou $ dans l'expression régulière pour `label_matcher`.
  - `source_labels` spécifie la valeur des étiquettes qui sont vérifiées par la ligne `label_matcher`.
  - `label_separator` spécifie le séparateur à utiliser dans la ligne `label_matcher` si plusieurs `source_labels` sont spécifiées. La valeur par défaut est `;`. Vous pouvez voir cette valeur par défaut utilisée dans la ligne `label_matcher` dans l'exemple suivant.
  - `metric_selectors`est une expression régulière qui spécifie les métriques à collecter et à envoyer CloudWatch.
  - `dimensions`est la liste des étiquettes à utiliser comme CloudWatch dimensions pour chaque métrique sélectionnée.

Consultez l'exemple `metric_declaration` suivant.

```
"metric_declaration": [
  {
     "source_labels":[ "Service", "Namespace"],
     "label_matcher":"(.*node-exporter.*|.*kube-dns.*);kube-system",
     "dimensions":[
        ["Service", "Namespace"]
     ],
     "metric_selectors":[
        "^coredns_dns_request_type_count_total$"
     ]
  }
]
```

Cet exemple montre comment configurer une section de format de métrique intégrée à envoyer en tant qu'événement de journaux si les conditions suivantes sont remplies :

- La valeur de `Service` contient `node-exporter` ou `kube-dns`.
- La valeur de `Namespace` est `kube-system`.
- La métrique Prometheus `coredns_dns_request_type_count_total` contient les deux étiquettes `Namespace` et `Service`.

L'événement de journal envoyé inclut la section en surbrillance suivante :

```
{
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"coredns_dns_request_type_count_total"
            }
         ],
         "Dimensions":[
            [
               "Namespace",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
```

```
    ],
    "Namespace":"kube-system",
    "Service":"kube-dns",
    "coredns_dns_request_type_count_total":2562,
    "eks_amazonaws_com_component":"kube-dns",
    "instance":"192.168.61.254:9153",
    "job":"kubernetes-service-endpoints",
    ...
}
```

Didacticiel pour l'ajout d'une nouvelle cible de récupération Prometheus : métrique du serveur d'API Prometheus

Le serveur d'API Kubernetes expose les métriques Prometheus sur les points de terminaison par défaut. L'exemple officiel de la configuration de récupération du serveur d'API Kubernetes est disponible sur Github.

Le didacticiel suivant montre comment effectuer les étapes suivantes pour commencer à importer les métriques du serveur d'API Kubernetes dans : CloudWatch

- Ajout de la configuration de scraping Prometheus pour le serveur d'API Kubernetes au fichier YAML de l'agent. CloudWatch
- Configuration des définitions de métriques intégrées au format métrique dans le fichier YAML de l' CloudWatch agent.
- (Facultatif) Création d'un CloudWatch tableau de bord pour les métriques du serveur d'API Kubernetes.

> **Note**
>
> Le serveur d'API Kubernetes expose des métriques de jauge, de compteur, d'histogramme et de synthèse. Dans cette version de Prometheus Metrics Support CloudWatch , seule les métriques de type jauge, compteur et résumé sont importées.

Pour commencer à collecter les métriques Prometheus du serveur d'API Kubernetes dans CloudWatch

1. Téléchargez la dernière version du fichier `prometheus-eks.yaml`, `prometheus-eks-fargate.yaml` ou `prometheus-k8s.yaml` en saisissant l'une des commandes suivantes.

Pour un cluster Amazon EKS avec le type de lancement EC2, saisissez la commande suivante :

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
```

Pour un cluster Amazon EKS avec le type de lancement Fargate, saisissez la commande suivante :

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
```

Pour un cluster Kubernetes s'exécutant sur une instance Amazon EC2, saisissez la commande suivante :

```
curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
```

2. Ouvrez le fichier avec un éditeur de texte, recherchez la section `prometheus-config` et ajoutez la section suivante à l'intérieur de cette section. Ensuite, enregistrez les modifications :

```
    # Scrape config for API servers
    - job_name: 'kubernetes-apiservers'
      kubernetes_sd_configs:
        - role: endpoints
          namespaces:
            names:
              - default
      scheme: https
      tls_config:
        ca_file: /var/run/secrets/kubernetes.io/serviceaccount/ca.crt
        insecure_skip_verify: true
      bearer_token_file: /var/run/secrets/kubernetes.io/serviceaccount/token
      relabel_configs:
      - source_labels: [__meta_kubernetes_service_name,
 __meta_kubernetes_endpoint_port_name]
        action: keep
        regex: kubernetes;https
```

```
      - action: replace
        source_labels:
        - __meta_kubernetes_namespace
        target_label: Namespace
      - action: replace
        source_labels:
        - __meta_kubernetes_service_name
        target_label: Service
```

3. Pendant que le fichier YAML est toujours ouvert dans l'éditeur de texte, recherchez la section `cwagentconfig.json`. Ajoutez la sous-section suivante et enregistrez les modifications. Cette section place les métriques du serveur d'API dans la liste des CloudWatch agents autorisés. Trois types de métriques du serveur d'API sont ajoutés à la liste d'autorisation :

   - nombre d'objets etcd
   - Métriques du contrôleur d'enregistrement du serveur d'API
   - Métriques de demande du serveur API

```
{"source_labels": ["job", "resource"],
  "label_matcher": "^kubernetes-apiservers;(services|daemonsets.apps|
deployments.apps|configmaps|endpoints|secrets|serviceaccounts|replicasets.apps)",
  "dimensions": [["ClusterName","Service","resource"]],
  "metric_selectors": [
  "^etcd_object_counts$"
  ]
},
{"source_labels": ["job", "name"],
   "label_matcher": "^kubernetes-apiservers;APIServiceRegistrationController$",
   "dimensions": [["ClusterName","Service","name"]],
   "metric_selectors": [
   "^workqueue_depth$",
   "^workqueue_adds_total$",
   "^workqueue_retries_total$"
  ]
},
{"source_labels": ["job","code"],
  "label_matcher": "^kubernetes-apiservers;2[0-9]{2}$",
  "dimensions": [["ClusterName","Service","code"]],
  "metric_selectors": [
   "^apiserver_request_total$"
  ]
```

```
},
{"source_labels": ["job"],
  "label_matcher": "^kubernetes-apiservers",
  "dimensions": [["ClusterName","Service"]],
  "metric_selectors": [
  "^apiserver_request_total$"
  ]
},
```

4. Si l' CloudWatch agent compatible avec Prometheus est déjà déployé dans le cluster, vous devez le supprimer en saisissant la commande suivante :

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

5. Déployez l' CloudWatch agent avec votre configuration mise à jour en saisissant l'une des commandes suivantes. Pour un cluster Amazon EKS avec le type de lancement EC2, saisissez :

```
kubectl apply -f prometheus-eks.yaml
```

Pour un cluster Amazon EKS avec le type de lancement Fargate, saisissez la commande suivante. Remplacez *MyCluster*et *régionalisez* par des valeurs correspondant à votre déploiement.

```
cat prometheus-eks-fargate.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Pour un cluster Kubernetes, saisissez la commande suivante : Remplacez *MyCluster*et *régionalisez* par des valeurs correspondant à votre déploiement.

```
cat prometheus-k8s.yaml \
| sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
| kubectl apply -f -
```

Ensuite, vous devriez voir un nouveau flux de journaux nommé kubernetes-apiservers dans le groupe de journaux /aws/containerinsights/***cluster_name***/prometheus. Ce flux de journal doit inclure les événements de journaux avec une définition intégrée du format de métrique comme suit :

```
{
```

```
   "CloudWatchMetrics":[
      {
         "Metrics":[
            {
               "Name":"apiserver_request_total"
            }
         ],
         "Dimensions":[
            [
               "ClusterName",
               "Service"
            ]
         ],
         "Namespace":"ContainerInsights/Prometheus"
      }
   ],
   "ClusterName":"my-cluster-name",
   "Namespace":"default",
   "Service":"kubernetes",
   "Timestamp":"1592267020339",
   "Version":"0",
   "apiserver_request_count":0,
   "apiserver_request_total":0,
   "code":"0",
   "component":"apiserver",
   "contentType":"application/json",
   "instance":"192.0.2.0:443",
   "job":"kubernetes-apiservers",
   "prom_metric_type":"counter",
   "resource":"pods",
   "scope":"namespace",
   "verb":"WATCH",
   "version":"v1"
}
```


Vous pouvez consulter vos métriques dans la CloudWatch console, dans l'espace de noms ContainerInsights/Prometheus. Vous pouvez également créer éventuellement un CloudWatch tableau de bord pour les métriques de votre serveur d'API Prometheus Kubernetes.

(En option) Création d'un tableau de bord pour les métriques du serveur d'API Kubernetes.

Pour voir les statistiques du serveur d'API Kubernetes dans votre tableau de bord, vous devez d'abord avoir effectué les étapes décrites dans les sections précédentes pour commencer à collecter ces statistiques dans. CloudWatch

Pour créer un tableau de bord pour les métriques du serveur d'API Kubernetes

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Assurez-vous d'avoir sélectionné la bonne AWS région.
3. Dans le panneau de navigation, choisissez Dashboards (Tableaux de bord).
4. Choisissez Créer un tableau de bord. Entrez un nom pour le nouveau tableau de bord, puis choisissez Créer un tableau de bord.
5. Dans Ajouter à ce tableau de bord, choisissez Annuler.
6. Choisissez Actions, View/Edit Attributes (Afficher/Modifier la source).
7. Téléchargez le fichier JSON suivant : Source de tableau de bord d'API Kubernetes.
8. Ouvrez le fichier JSON que vous avez téléchargé avec un éditeur de texte et apportez les modifications suivantes :
   - Remplacez toutes les chaînes `{{YOUR_CLUSTER_NAME}}` par le nom exact de votre cluster. Assurez-vous de ne pas ajouter d'espaces avant ou après le texte.
   - Remplacez toutes les chaînes `{{YOUR_AWS_REGION}}` par le nom de la région dans laquelle les métriques sont collectées. Par exemple, `us-west-2`. Assurez-vous de ne pas ajouter d'espaces avant ou après le texte.
9. Copiez l'intégralité du blob JSON et collez-le dans la zone de texte de la CloudWatch console, en remplaçant ce qui s'y trouve déjà.
10. Choisissez Mettre à jour, Enregistrer le tableau de bord.

(En option) Configuration d'exemples d'applications Amazon EKS conteneurisées pour les test de métriques Prometheus

Pour tester la prise en charge des métriques Prometheus CloudWatch dans Container Insights, vous pouvez configurer une ou plusieurs des charges de travail conteneurisées suivantes. L' CloudWatch agent prenant en charge Prometheus collecte automatiquement les métriques de chacune de ces charges de travail. Pour voir les métriques collectées par défaut, consultez Métriques Prometheus collectées par l'agent CloudWatch .

Avant de pouvoir installer l'une de ces charges de travail, vous devez installer Helm 3.x en entrant les commandes suivantes :

```
brew install helm
```

Pour plus d'informations, consultez Helm.

Rubriques

- Configuration d'un exemple d'application AWS App Mesh pour Amazon EKS et Kubernetes
- Configuration de NGINX avec un exemple de trafic sur Amazon EKS et Kubernetes
- Configuration de memcached avec un exportateur de métriques sur Amazon EKS et Kubernetes
- Configuration d'un exemple d'application Java/JMX pour Amazon EKS et Kubernetes
- Configuration de HAProxy avec un exportateur de métriques sur Amazon EKS et Kubernetes
- Didacticiel pour ajouter une nouvelle cible de récupération Prometheus : Redis sur les clusters Amazon EKS et Kubernetes

Configuration d'un exemple d'application AWS App Mesh pour Amazon EKS et Kubernetes

Le support de Prometheus dans Container Insights prend CloudWatch en charge. AWS App Mesh Les sections suivantes expliquent comment configurer App Mesh.

CloudWatch Container Insights peut également collecter les journaux d'accès d'App Mesh Envoy. Pour plus d’informations, consultez (En option) Activez les journaux d'accès App Mesh Envoy..

Rubriques

- Configuration d'un exemple d'application AWS App Mesh sur un cluster Amazon EKS avec le type de lancement EC2 ou un cluster Kubernetes
- Configurer un AWS App Mesh exemple de charge de travail sur un cluster Amazon EKS avec le type de lancement Fargate

Configuration d'un exemple d'application AWS App Mesh sur un cluster Amazon EKS avec le type de lancement EC2 ou un cluster Kubernetes

Utilisez ces instructions si vous configurez App Mesh sur un cluster exécutant Amazon EKS avec le type de lancement EC2 ou un cluster Kubernetes.

Configuration des autorisations IAM

Vous devez ajouter la AWSAppMeshFullAccesspolitique au rôle IAM pour votre groupe de nœuds Amazon EKS ou Kubernetes. Sur Amazon EKS, le nom du groupe de nœuds ressemble à `eksctl-integ-test-eks-prometheus-NodeInstanceRole-ABCDEFHIJKL`. Sur Kubernetes, il peut ressembler à `nodes.integ-test-kops-prometheus.k8s.local`.

Installation d'App Mesh

Pour installer le contrôleur App Mesh Kubernetes, suivez les instructions dans App Mesh Controller.

Installation d'un exemple d'application

aws-app-mesh-examplescontient plusieurs procédures pas à pas de Kubernetes App Mesh. Pour ce didacticiel, vous installez un exemple d'application de couleur qui montre comment les acheminements http peuvent utiliser les en-têtes pour faire correspondre les requêtes entrantes.

Pour utiliser un exemple d'application App Mesh pour tester Container Insights

1. Installez l'application en suivant ces instructions : https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-http-headers.
2. Lancez un pod curler pour générer du trafic :

```
kubectl -n default run -it curler --image=tutum/curl /bin/bash
```

3. Enroulez différents points de terminaison en changeant les en-têtes HTTP. Exécutez la commande curl plusieurs fois, comme indiqué :

```
curl -H "color_header: blue" front.howto-k8s-http-headers.svc.cluster.local:8080/;
 echo;

curl -H "color_header: red" front.howto-k8s-http-headers.svc.cluster.local:8080/;
 echo;

curl -H "color_header: yellow" front.howto-k8s-http-
headers.svc.cluster.local:8080/; echo;
```

4. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
5. Dans la AWS région dans laquelle votre cluster est exécuté, choisissez Metrics dans le volet de navigation. La métrique se trouve dans l'espace de noms ContainerInsights/Prometheus.

6. Pour voir les événements CloudWatch Logs, choisissez Log groups dans le volet de navigation. Les évènements sont dans le groupe de journaux `/aws/containerinsights/your_cluster_name/prometheus` du flux de journaux `kubernetes-pod-appmesh-envoy`.

Suppression de l'environnement de test App Mesh

Lorsque vous avez terminé d'utiliser App Mesh et l'exemple d'application, utilisez les commandes suivantes pour supprimer les ressources inutiles. Supprimez l'exemple d'application en saisissant la commande suivante :

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-http-headers/
kubectl delete -f _output/manifest.yaml
```

Supprimez App Mesh Controller en saisissant la commande suivante :

```
helm delete appmesh-controller -n appmesh-system
```

Configurer un AWS App Mesh exemple de charge de travail sur un cluster Amazon EKS avec le type de lancement Fargate

Utilisez ces instructions si vous configurez App Mesh sur un cluster exécutant Amazon EKS avec le type de lancement Fargate.

Configuration des autorisations IAM

Pour configurer les autorisations IAM, saisissez la commande suivante : *MyCluster*Remplacez-le par le nom de votre cluster.

```
eksctl create iamserviceaccount --cluster MyCluster \
 --namespace howto-k8s-fargate \
 --name appmesh-pod \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
 --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
 --override-existing-serviceaccounts \
```

```
  --approve
```

Installation d'App Mesh

Pour installer le contrôleur App Mesh Kubernetes, suivez les instructions dans App Mesh Controller. Veillez à suivre les instructions pour Amazon EKS avec le type de lancement Fargate.

Installation d'un exemple d'application

aws-app-mesh-examplescontient plusieurs procédures pas à pas de Kubernetes App Mesh. Pour ce didacticiel, vous installez un exemple d'application de couleur qui fonctionne pour les clusters Amazon EKS avec le type de lancement Fargate.

Pour utiliser un exemple d'application App Mesh pour tester Container Insights

1. Installez l'application en suivant ces instructions : https://github.com/aws/aws-app-mesh-examples/tree/main/walkthroughs/howto-k8s-fargate.

   Ces instructions supposent que vous créez un nouveau cluster avec le bon profil Fargate. Si vous souhaitez utiliser un cluster Amazon EKS que vous avez déjà configuré, vous pouvez utiliser les commandes suivantes pour configurer ce cluster pour cette démonstration. *MyCluster*Remplacez-le par le nom de votre cluster.

   ```
   eksctl create iamserviceaccount --cluster MyCluster \
    --namespace howto-k8s-fargate \
    --name appmesh-pod \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshEnvoyAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapDiscoverInstanceAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSXRayDaemonWriteAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchLogsFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSAppMeshFullAccess \
    --attach-policy-arn arn:aws:iam::aws:policy/AWSCloudMapFullAccess \
    --override-existing-serviceaccounts \
    --approve
   ```

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --namespace howto-k8s-fargate --name howto-k8s-fargate
   ```

2. Port avant le déploiement de l'application frontale :

   ```
   kubectl -n howto-k8s-fargate port-forward deployment/front 8080:8080
   ```

3. Enroulez l'application frontale :

```
while true; do  curl -s http://localhost:8080/color; sleep 0.1; echo ; done
```

4. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
5. Dans la AWS région dans laquelle votre cluster est exécuté, choisissez Metrics dans le volet de navigation. La métrique se trouve dans l'espace de noms ContainerInsights/Prometheus.
6. Pour voir les événements CloudWatch Logs, choisissez Log groups dans le volet de navigation. Les évènements sont dans le groupe de journaux `/aws/containerinsights/your_cluster_name/prometheus` du flux de journaux `kubernetes-pod-appmesh-envoy`.

Suppression de l'environnement de test App Mesh

Lorsque vous avez terminé d'utiliser App Mesh et l'exemple d'application, utilisez les commandes suivantes pour supprimer les ressources inutiles. Supprimez l'exemple d'application en saisissant la commande suivante :

```
cd aws-app-mesh-examples/walkthroughs/howto-k8s-fargate/
kubectl delete -f _output/manifest.yaml
```

Supprimez App Mesh Controller en saisissant la commande suivante :

```
helm delete appmesh-controller -n appmesh-system
```

Configuration de NGINX avec un exemple de trafic sur Amazon EKS et Kubernetes

NGINX est un serveur web qui peut également être utilisé comme équilibreur de charge et proxy inverse. Pour plus d'informations sur la façon dont Kubernetes utilise NGINX pour es entrées, veuillez consulter kubernetes/ingress-nginx.

Pour installer NGINX avec un exemple de service de trafic pour tester la prise en charge de Container Insights Prometheus

1. Saisissez la commande suivante pour ajouter le référentiel ingress-nginx Helm.

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
```

2. Entrez la commande suivante :

```
kubectl create namespace nginx-ingress-sample

helm install my-nginx ingress-nginx/ingress-nginx \
--namespace nginx-ingress-sample \
--set controller.metrics.enabled=true \
--set-string controller.metrics.service.annotations."prometheus\.io/port"="10254" \
--set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
```

3. Vérifiez si les services ont démarré correctement en entrant la commande suivante :

```
kubectl get service -n nginx-ingress-sample
```

   La sortie de cette commande doit afficher plusieurs colonnes, y compris une colonne `EXTERNAL-IP`.

4. Définissez une variable `EXTERNAL-IP` avec la valeur de la colonne `EXTERNAL-IP` dans la ligne du contrôleur d'entrée NGINX.

```
EXTERNAL_IP=your-nginx-controller-external-ip
```

5. Démarrez un exemple de trafic NGINX en entrant la commande suivante.

```
SAMPLE_TRAFFIC_NAMESPACE=nginx-sample-traffic
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_traffic/nginx-traffic/nginx-traffic-sample.yaml |
sed "s/{{external_ip}}/$EXTERNAL_IP/g" |
sed "s/{{namespace}}/$SAMPLE_TRAFFIC_NAMESPACE/g" |
kubectl apply -f -
```

6. Entrez la commande suivante pour vérifier que l'état des trois pods est `Running`.

```
kubectl get pod -n $SAMPLE_TRAFFIC_NAMESPACE
```

   S'ils sont en cours d'exécution, vous devriez bientôt voir des métriques dans l'espace de noms ContainerInsights/Prometheus.

Pour désinstaller NGINX et l'exemple d'application de trafic

1. Supprimez l'exemple de service de trafic en entrant la commande suivante :

```
kubectl delete namespace $SAMPLE_TRAFFIC_NAMESPACE
```

2. Supprimez la sortie NGINX par le nom de la version Helm.

```
helm uninstall my-nginx --namespace nginx-ingress-sample
kubectl delete namespace nginx-ingress-sample
```

Configuration de memcached avec un exportateur de métriques sur Amazon EKS et Kubernetes

memcached est un système de mise en cache d'objets mémoire open source. Pour plus d'informations, consultez What is Memcached?.

Si vous exécutez memcached sur un cluster avec le type de lancement Fargate, vous devez configurer un profil Fargate avant de réaliser les étapes de cette procédure. Pour configurer le profil, saisissez la commande suivante. *MyCluster*Remplacez-le par le nom de votre cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace memcached-sample --name memcached-sample
```

Pour installer memcached avec un exportateur de métriques afin de tester la prise en charge de Container Insights Prometheus

1. Saisissez la commande suivante pour ajouter le référentiel :

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

2. Entrez la commande suivante pour créer un nouvel espace de noms :

```
kubectl create namespace memcached-sample
```

3. Entrez la commande suivante pour installer Memcached

```
helm install my-memcached bitnami/memcached --namespace memcached-sample \
--set metrics.enabled=true \
--set-string serviceAnnotations.prometheus\\.io/port="9150" \
--set-string serviceAnnotations.prometheus\\.io/scrape="true"
```

4. Entrez la commande suivante pour vérifier l'annotation du service en cours d'exécution :

```
kubectl describe service my-memcached-metrics -n memcached-sample
```

Les deux annotations suivantes doivent s'afficher :

```
Annotations:   prometheus.io/port: 9150
               prometheus.io/scrape: true
```

Pour désinstaller memcached

- Entrez la commande suivante :

```
helm uninstall my-memcached --namespace memcached-sample
kubectl delete namespace memcached-sample
```

Configuration d'un exemple d'application Java/JMX pour Amazon EKS et Kubernetes

JMX Exporter est un exportateur Prometheus officiel qui peut récupérer et exposer des mBeans JMX en tant que métriques Prometheus. Pour plus d'informations, consultez prometheus/jmx_exporter.

Container Insights peut collecter des métriques Prometheus prédéfinies à partir de Java Virtual Machine (JVM), Java et Tomcat (Catalina) à l'aide de JMX Exporter.

Configuration de récupération Prometheus par défaut

Par défaut, l' CloudWatch agent compatible avec Prometheus supprime les métriques `http://CLUSTER_IP:9404/metrics` Prometheus Java/JMX présentes sur chaque pod d'un cluster Amazon EKS ou Kubernetes. Cette opération est exécutée par la découverte `role: pod` de Prometheus `kubernetes_sd_config`. 9404 est le port par défaut alloué pour JMX Exporter par Prometheus. Pour plus d'informations sur la découverte `role: pod`, consultez pod. Vous pouvez configurer JMX Exporter pour qu'il expose les métriques sur un autre port ou chemin d'accès. Si vous modifiez le port ou le chemin, mettez à jour le jmx scrape_config par défaut dans la carte de configuration de l' CloudWatch agent. Exécutez la commande suivante pour obtenir la configuration actuelle de l' CloudWatch agent Prometheus :

```
kubectl describe cm prometheus-config -n amazon-cloudwatch
```

Les champs à modifier sont les champs `/metrics` et `regex: '.*:9404$'`, comme mis en évidence dans l'exemple suivant.

```
job_name: 'kubernetes-jmx-pod'
sample_limit: 10000
metrics_path: /metrics
kubernetes_sd_configs:
- role: pod
relabel_configs:
- source_labels: [__address__]
  action: keep
  regex: '.*:9404$'
- action: replace
  regex: (.+)
  source_labels:
```

Autre configuration de récupération Prometheus

Si vous exposez votre application en cours d'exécution sur un ensemble de pods avec les exportateurs Prometheus Java/JMX par un service Kubernetes, vous pouvez également basculer pour utiliser la découverte `role: service` ou `role: endpoint` de Prometheus `kubernetes_sd_config`. Pour plus d'informations sur ces méthodes de découverte, consultez service, endpoints (points de terminaison), et <kubernetes_sd_config>.

D'autres méta-étiquettes sont fournies par ces deux modes de découverte de services, ce qui peut vous être utile pour créer les dimensions CloudWatch des métriques. Par exemple, vous pouvez réétiqueter `__meta_kubernetes_service_name` sur `Service` et l'inclure dans la dimension de vos métriques. Pour plus d'informations sur la personnalisation de vos CloudWatch indicateurs et de leurs dimensions, consultez. CloudWatch configuration de l'agent pour Prometheus

Image Docker avec JMX Exporter

Ensuite, créez une image Docker. Les sections suivantes fournissent deux exemples de fichiers Docker.

Lorsque vous avez créé l'image, chargez-la dans Amazon EKS ou Kubernetes, puis exécutez la commande suivante pour vérifier que les métriques Prometheus sont exposées par `JMX_EXPORTER` sur le port 9404. Remplacez *`$JAR_SAMPLE_TRAFIC_POD`* par le nom du pod en cours d'exécution et remplacez *`$JAR_SAMPLE_TRAFIC_NAMESPACE`* par l'espace de noms de votre application.

Si vous exécutez JMX Exporter sur un cluster avec le type de lancement Fargate, vous devez configurer un profil Fargate avant de réaliser les étapes de cette procédure. Pour configurer le profil, saisissez la commande suivante. *MyCluster*Remplacez-le par le nom de votre cluster.

```
eksctl create fargateprofile --cluster MyCluster \
--namespace $JAR_SAMPLE_TRAFFIC_NAMESPACE\
 --name $JAR_SAMPLE_TRAFFIC_NAMESPACE
```

```
kubectl exec $JAR_SAMPLE_TRAFFIC_POD -n $JARCAT_SAMPLE_TRAFFIC_NAMESPACE -- curl
 http://localhost:9404
```

Exemple : image Apache Tomcat Docker avec métriques Prometheus

Le serveur Apache Tomcat expose les mBeans JMX par défaut. Vous pouvez intégrer JMX Exporter avec Tomcat pour exposer les mBeans JMX en tant que métriques Prometheus. L'exemple de fichier Docker suivant montre les étapes à suivre pour créer une image de test :

```
# From Tomcat 9.0 JDK8 OpenJDK
FROM tomcat:9.0-jdk8-openjdk

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter
COPY ./setenv.sh /usr/local/tomcat/bin
COPY your web application.war /usr/local/tomcat/webapps/

RUN chmod  o+x /usr/local/tomcat/bin/setenv.sh

ENTRYPOINT ["catalina.sh", "run"]
```

La liste suivante explique les quatre lignes `COPY` de ce fichier Docker.

- Téléchargez le dernier fichier jar de JMX Exporter à partir de https://github.com/prometheus/jmx_exporter.
- `config.yaml` est le fichier de configuration de JMX Exporter. Pour plus d'informations, consultez https://github.com/prometheus/jmx_exporter#Configuration.

  Voici un exemple de fichier de configuration pour Java et Tomcat :

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE
```

```
- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `setenv.sh` est un script de démarrage Tomcat pour démarrer JMX exporter avec Tomcat et exposer les métriques Prometheus sur le port 9404 de l'hôte local. Il fournit également à JMX Exporter le chemin d'accès au fichier `config.yaml`.

```
$ cat setenv.sh
export JAVA_OPTS="-javaagent:/opt/jmx_exporter/
jmx_prometheus_javaagent-0.12.0.jar=9404:/opt/jmx_exporter/config.yaml $JAVA_OPTS"
```

- Votre application web .war est le fichier `war` de votre application web à charger par Tomcat.

Créez une image Docker avec cette configuration et téléchargez-la dans un référentiel d'images.

Exemple : image Docker d'application Java Jar avec métriques Prometheus

L'exemple de fichier Docker suivant montre les étapes à suivre pour créer une image de test :

```
# Alpine Linux with OpenJDK JRE
FROM openjdk:8-jre-alpine

RUN mkdir -p /opt/jmx_exporter

COPY ./jmx_prometheus_javaagent-0.12.0.jar /opt/jmx_exporter
COPY ./SampleJavaApplication-1.0-SNAPSHOT.jar /opt/jmx_exporter
COPY ./start_exporter_example.sh /opt/jmx_exporter
COPY ./config.yaml /opt/jmx_exporter

RUN chmod -R o+x /opt/jmx_exporter
RUN apk add curl

ENTRYPOINT exec /opt/jmx_exporter/start_exporter_example.sh
```

La liste suivante explique les quatre lignes `COPY` de ce fichier Docker.

- Téléchargez le dernier fichier jar de JMX Exporter à partir de https://github.com/prometheus/jmx_exporter.
- `config.yaml` est le fichier de configuration de JMX Exporter. Pour plus d'informations, consultez https://github.com/prometheus/jmx_exporter#Configuration.

  Voici un exemple de fichier de configuration pour Java et Tomcat :

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
```

```
  name: catalina_threadpool_$3
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

- `start_exporter_example.sh` est le script pour démarrer l'application JAR avec les métriques Prometheus exportées. Il fournit également à JMX Exporter le chemin d'accès au fichier `config.yaml`.

```
$ cat start_exporter_example.sh
java -javaagent:/opt/jmx_exporter/jmx_prometheus_javaagent-0.12.0.jar=9404:/
opt/jmx_exporter/config.yaml -cp  /opt/jmx_exporter/SampleJavaApplication-1.0-
SNAPSHOT.jar com.gubupt.sample.app.App
```

- SampleJavaApplication-1.0-Snapshot.jar est l'exemple de fichier jar de l'application Java. Remplacez-le par l'application Java que vous souhaitez surveiller.

Créez une image Docker avec cette configuration et téléchargez-la dans un référentiel d'images.

Configuration de HAProxy avec un exportateur de métriques sur Amazon EKS et Kubernetes

HAProxy est une application proxy open source. Pour plus d'informations, consultez HAProxy.

Si vous exécutez HAProxy sur un cluster avec le type de lancement Fargate, vous devez configurer un profil Fargate avant de réaliser les étapes de cette procédure. Pour configurer le profil, saisissez la commande suivante. *`MyCluster`*Remplacez-le par le nom de votre cluster.

```
eksctl create fargateprofile --cluster MyCluster \
```

```
--namespace haproxy-ingress-sample --name haproxy-ingress-sample
```

Pour installer HAProxy avec un exportateur de métriques afin de tester la prise en charge de Container Insights Prometheus

1. Entrez la commande suivante pour ajouter le référentiel de l'incubateur Helm :

   ```
   helm repo add haproxy-ingress https://haproxy-ingress.github.io/charts
   ```

2. Entrez la commande suivante pour créer un nouvel espace de noms :

   ```
   kubectl create namespace haproxy-ingress-sample
   ```

3. Entrez les commandes suivantes pour installer HAProxy :

   ```
   helm install haproxy haproxy-ingress/haproxy-ingress \
   --namespace haproxy-ingress-sample \
   --set defaultBackend.enabled=true \
   --set controller.stats.enabled=true \
   --set controller.metrics.enabled=true \
   --set-string controller.metrics.service.annotations."prometheus\.io/port"="9101" \
   --set-string controller.metrics.service.annotations."prometheus\.io/scrape"="true"
   ```

4. Entrez la commande suivante pour confirmer l'annotation du service :

   ```
   kubectl describe service haproxy-haproxy-ingress-metrics -n haproxy-ingress-sample
   ```

   Les annotations suivantes doivent s'afficher.

   ```
   Annotations:   prometheus.io/port: 9101
                  prometheus.io/scrape: true
   ```

Pour désinstaller HAProxy

- Entrez la commande suivante :

  ```
  helm uninstall haproxy --namespace haproxy-ingress-sample
  kubectl delete namespace haproxy-ingress-sample
  ```

Didacticiel pour ajouter une nouvelle cible de récupération Prometheus : Redis sur les clusters Amazon EKS et Kubernetes

Ce didacticiel propose une présentation pratique de l'utilisation des métriques Prometheus d'un exemple d'application Redis sur Amazon EKS et Kubernetes. Redis (https://redis.io/) est un magasin de structures de données open source (sous licence BSD) en mémoire, utilisé comme base de données, cache et agent de messages. Pour en savoir plus, consultez redis.

redis_exportateur (sous licence MIT) est utilisé pour exposer les métriques Redis Prometheus sur le port spécifié (par défaut : 0.0.0.0:9121). Pour en savoir plus, consultez redis_exporter.

Les images Docker dans les deux référentiels Docker Hub suivants sont utilisées dans ce didacticiel :

- Redis
- redis_exporter

Pour installer un exemple d'application Redis qui expose les métriques Prometheus

1. Définissez l'espace de noms pour l'exemple d'application Redis.

   ```
   REDIS_NAMESPACE=redis-sample
   ```

2. Si vous exécutez Redis sur un cluster avec le type de lancement Fargate, vous devez configurer un profil Fargate. Pour configurer le profil, saisissez la commande suivante. *MyCluster*Remplacez-le par le nom de votre cluster.

   ```
   eksctl create fargateprofile --cluster MyCluster \
   --namespace $REDIS_NAMESPACE --name $REDIS_NAMESPACE
   ```

3. Saisissez la commande suivante pour installer l'exemple d'application Redis.

   ```
   curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
   insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
   prometheus/sample_traffic/redis/redis-traffic-sample.yaml \
   | sed "s/{{namespace}}/$REDIS_NAMESPACE/g" \
   | kubectl apply -f -
   ```

4. L'installation comprend un service nommé `my-redis-metrics` qui expose les métriques Redis Prometheus sur le port 9121. Saisissez la commande suivante pour obtenir les détails du service :

```
kubectl describe service/my-redis-metrics  -n $REDIS_NAMESPACE
```

Dans la `Annotations` section des résultats, vous verrez deux annotations correspondant à la configuration Prometheus Scrape de l'agent, afin qu'il puisse découvrir CloudWatch automatiquement les charges de travail :

```
prometheus.io/port: 9121
prometheus.io/scrape: true
```

La configuration de récupération Prometheus associée est disponible dans la section `- job_name: kubernetes-service-endpoints` de `kubernetes-eks.yaml` ou `kubernetes-k8s.yaml`.

Pour commencer à collecter les métriques Redis Prometheus dans CloudWatch

1. Téléchargez la dernière version du fichier `kubernetes-eks.yaml` ou `kubernetes-k8s.yaml` en saisissant l'une des commandes suivantes. Pour un cluster Amazon EKS avec le type de lancement EC2, saisissez cette commande.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks.yaml
   ```

   Pour un cluster Amazon EKS avec le type de lancement Fargate, saisissez cette commande.

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-eks-fargate.yaml
   ```

   Pour un cluster Kubernetes s'exécutant sur une instance Amazon EC2, saisissez cette commande :

   ```
   curl -O https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-prometheus/prometheus-k8s.yaml
   ```

2. Ouvrez le fichier avec un éditeur de texte et trouvez la section `cwagentconfig.json`. Ajoutez la sous-section suivante et enregistrez les modifications. Assurez-vous que l'indentation suive le modèle existant.

```
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName"]],
  "metric_selectors": [
    "^redis_net_(in|out)put_bytes_total$",
    "^redis_(expired|evicted)_keys_total$",
    "^redis_keyspace_(hits|misses)_total$",
    "^redis_memory_used_bytes$",
    "^redis_connected_clients$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","cmd"]],
  "metric_selectors": [
    "^redis_commands_total$"
  ]
},
{
  "source_labels": ["pod_name"],
  "label_matcher": "^redis-instance$",
  "dimensions": [["Namespace","ClusterName","db"]],
  "metric_selectors": [
    "^redis_db_keys$"
  ]
},
```

La section que vous avez ajoutée place les métriques Redis dans la liste des CloudWatch agents autorisés. Pour une liste de ces métriques, reportez-vous à la section suivante.

3. Si l' CloudWatch agent compatible avec Prometheus est déjà déployé dans ce cluster, vous devez le supprimer en saisissant la commande suivante.

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
```

4. Déployez l' CloudWatch agent avec votre configuration mise à jour en saisissant l'une des commandes suivantes. Remplacez *MyCluster*et *région* en fonction de vos paramètres.

   Pour un cluster Amazon EKS avec le type de lancement EC2, saisissez cette commande.

   ```
   kubectl apply -f prometheus-eks.yaml
   ```

   Pour un cluster Amazon EKS avec le type de lancement Fargate, saisissez cette commande.

   ```
   cat prometheus-eks-fargate.yaml \
   | sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
   | kubectl apply -f -
   ```

   Pour un cluster Kubernetes, saisissez cette commande.

   ```
   cat prometheus-k8s.yaml \
   | sed "s/{{cluster_name}}/MyCluster/;s/{{region_name}}/region/" \
   | kubectl apply -f -
   ```

Affichage de vos métriques Prometheus Redis

Ce didacticiel envoie les métriques suivantes à l'espace de noms ContainerInsights/Prometheus dans. CloudWatch Vous pouvez utiliser la CloudWatch console pour voir les métriques de cet espace de noms.

Nom de la métrique	Dimensions	
`redis_net_input_bytes_total`	ClusterName, `Namespace`	
`redis_net_output_bytes_total`	ClusterName, `Namespace`	
`redis_expired_keys_total`	ClusterName, `Namespace`	

Nom de la métrique	Dimensions	
redis_evicted_keys_total	ClusterName, Namespace	
redis_keyspace_hits_total	ClusterName, Namespace	
redis_keyspace_misses_total	ClusterName, Namespace	
redis_memory_used_bytes	ClusterName, Namespace	
redis_connected_clients	ClusterName, Namespace	
redis_commands_total	ClusterNameNamespace , cmd	
redis_db_keys	ClusterName,Namespace , base de données	

> **Note**
>
> Les valeurs de la dimension cmd peuvent être : append, client, command, config, dbsize, flushall, get, incr, info, latency ou slowlog.
> Les valeurs de la dimension db peuvent être db0 ou db15.

Vous pouvez également créer un CloudWatch tableau de bord pour vos métriques Redis Prometheus.

Pour créer un tableau de bord pour les métriques Prometheus Redis

1. Créez des variables d'environnement, en remplaçant les valeurs ci-dessous pour correspondre à votre déploiement.

```
DASHBOARD_NAME=your_cw_dashboard_name
REGION_NAME=your_metric_region_such_as_us-east-1
CLUSTER_NAME=your_k8s_cluster_name_here
NAMESPACE=your_redis_service_namespace_here
```

2. Saisissez la commande suivante pour créer le tableau de bord.

```
curl https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-
insights/latest/k8s-deployment-manifest-templates/deployment-mode/service/cwagent-
prometheus/sample_cloudwatch_dashboards/redis/cw_dashboard_redis.json \
| sed "s/{{YOUR_AWS_REGION}}/${REGION_NAME}/g" \
| sed "s/{{YOUR_CLUSTER_NAME}}/${CLUSTER_NAME}/g" \
| sed "s/{{YOUR_NAMESPACE}}/${NAMESPACE}/g" \
```

Conversion du type métrique Prometheus par l'Agent CloudWatch

Les bibliothèques clientes Prometheus proposent quatre types de métriques principaux :

- Compteur
- Jauge
- Récapitulatif
- Histogramme

L' CloudWatch agent prend en charge les types de métriques compteur, jauge et récapitulatif. La prise en charge des métriques d'histogramme sera ajoutée lors d'une prochaine mise à jour.

Les métriques Prometheus dont le type de métrique d'histogramme n'est pas pris en charge sont supprimées par l'agent. CloudWatch Pour plus d’informations, consultez Journalisation des métriques Prometheus ignorées.

Métriques de jauge

Une métrique de jauge Prometheus est une métrique qui représente une valeur numérique unique qui peut arbitrairement monter et descendre. L' CloudWatch agent extrait les métriques de jauge et envoie ces valeurs directement.

Métriques de compteur

Une métrique de compteur Prometheus est une métrique cumulative qui représente un compteur monotone unique dont la valeur ne peut qu'augmenter ou être réinitialisée à zéro. L' CloudWatchagent calcule un delta à partir du scrape précédent et envoie la valeur delta sous forme de valeur métrique dans le journal des événements. L' CloudWatch agent commencera donc à produire un événement journal à partir de la deuxième éraflure et continuera avec les éraflures suivantes, le cas échéant.

Métriques de résumé

Une métrique de résumé Prometheus est un type de métrique complexe qui est représenté par plusieurs points de données. Elle fournit un nombre total d'observations et une somme de toutes les valeurs observées. Elle calcule les quantiles configurables sur une fenêtre temporelle coulissante.

La somme et le nombre d'une métrique de résumé sont cumulatifs, mais les quantiles ne le sont pas. L'exemple suivant illustre la variance des quantiles.

```
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 7.123e-06
go_gc_duration_seconds{quantile="0.25"} 9.204e-06
go_gc_duration_seconds{quantile="0.5"} 1.1065e-05
go_gc_duration_seconds{quantile="0.75"} 2.8731e-05
go_gc_duration_seconds{quantile="1"} 0.003841496
go_gc_duration_seconds_sum 0.37630427
go_gc_duration_seconds_count 9774
```

L' CloudWatch agent gère la somme et le nombre d'une métrique récapitulative de la même manière qu'il gère les contre-métriques, comme décrit dans la section précédente. L' CloudWatch agent conserve les valeurs quantiles telles qu'elles ont été initialement indiquées.

Métriques Prometheus collectées par l'agent CloudWatch

L' CloudWatch agent bénéficiant du support de Prometheus collecte automatiquement les métriques de plusieurs services et charges de travail. Les métriques collectées par défaut sont répertoriées dans les sections suivantes. Vous pouvez également configurer l'agent pour collecter plus de métriques à partir de ces services, et pour collecter des métriques Prometheus à partir d'autres

applications et services. Pour plus d'informations sur la collecte de métriques supplémentaires, consultez CloudWatch configuration de l'agent pour Prometheus.

Les métriques Prometheus collectées à partir des clusters Amazon EKS et Kubernetes se trouvent dans l'espace de noms /Prometheus. ContainerInsights Les métriques Prometheus collectées à partir des clusters Amazon ECS se trouvent dans l'espace de noms ECS/ /Prometheus. ContainerInsights

Rubriques

Métriques Prometheus pour App Mesh

Les métriques suivantes sont automatiquement collectées à partir de App Mesh.

CloudWatch Container Insights peut également collecter les journaux d'accès d'App Mesh Envoy. Pour plus d'informations, consultez (En option) Activez les journaux d'accès App Mesh Envoy..

Métriques Prometheus pour App Mesh sur les clusters Amazon EKS et Kubernetes

Nom de la métrique	Dimensions	
`envoy_http_downstream_rq_total`	ClusterName, `Namespace`	
`envoy_http_downstream_rq_xx`	ClusterName, `Namespace` ClusterName, envoy_http_conn_manager_prefix`Namespace` , envoy_response_code_class	
`envoy_cluster_upstream_cx_rx_bytes_total`	ClusterName, `Namespace`	

Nom de la métrique	Dimensions	
envoy_cluster_upstream_cx_tx_bytes_total	ClusterName, Namespace	
envoy_cluster_membership_healthy	ClusterName, Namespace	
envoy_cluster_membership_total	ClusterName, Namespace	
envoy_server_memory_heap_size	ClusterName, Namespace	
envoy_server_memory_allocated	ClusterName, Namespace	
envoy_cluster_upstream_cx_connect_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_pending_failure_eject	ClusterName, Namespace	

Nom de la métrique	Dimensions	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, Namespace	
envoy_cluster_upstream_rq_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, Namespace	
envoy_cluster_upstream_rq_rx_reset	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, Namespace	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, Namespace	

Nom de la métrique	Dimensions	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, Namespace	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, Namespace	

Nom de la métrique	Dimensions	
envoy_cluster_upstream_rq_retry	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_success	ClusterName, Namespace	
envoy_cluster_upstream_rq_retry_overflow	ClusterName, Namespace	
envoy_server_live	ClusterName, Namespace	
envoy_server_uptime	ClusterName, Namespace	

Métriques Prometheus pour App Mesh sur les clusters Amazon ECS

Nom de la métrique	Dimensions	
envoy_http_downstream_rq_total	ClusterName, TaskDefinitionFamily	
envoy_http_downstream_rq_xx	ClusterName, TaskDefinitionFamily	
envoy_cluster_upst	ClusterName, TaskDefinitionFamily	

Nom de la métrique	Dimensions	
ream_cx_r x_bytes_total		
envoy_clu ster_upst ream_cx_t x_bytes_total	ClusterName, TaskDefinitionFamily	
envoy_clu ster_memb ership_healthy	ClusterName, TaskDefinitionFamily	
envoy_clu ster_memb ership_total	ClusterName, TaskDefinitionFamily	
envoy_ser ver_memor y_heap_size	ClusterName, TaskDefinitionFamily	
envoy_ser ver_memor y_allocated	ClusterName, TaskDefinitionFamily	
envoy_clu ster_upst ream_cx_c onnect_timeout	ClusterName, TaskDefinitionFamily	
envoy_clu ster_upst ream_rq_p ending_fa ilure_eject	ClusterName, TaskDefinitionFamily	

Nom de la métrique	Dimensions	
envoy_cluster_upstream_rq_pending_overflow	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_try_per_timeout	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_rx_reset	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_local_with_active_rq	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_cx_destroy_remote_active_rq	ClusterName, TaskDefinitionFamily	

Nom de la métrique	Dimensions	
envoy_cluster_upstream_rq_maintenance_mode	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_paused_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_resumed_reading_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_backed_up_total	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_flow_control_drained_total	ClusterName, TaskDefinitionFamily	

Nom de la métrique	Dimensions	
envoy_cluster_upstream_rq_retry	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_retry_success	ClusterName, TaskDefinitionFamily	
envoy_cluster_upstream_rq_retry_overflow	ClusterName, TaskDefinitionFamily	
envoy_server_live	ClusterName, TaskDefinitionFamily	
envoy_server_uptime	ClusterName, TaskDefinitionFamily	
envoy_http_downstream_rq_xx	ClusterName, envoy_http_conn_manager_prefix TaskDefinitionFamily, envoy_response_code_class ClusterName, TaskDefinitionFamily envoy_response_code_class	

> **Note**
>
> TaskDefinitionFamily est l'espace de noms Kubernetes du maillage.
> La valeur de envoy_http_conn_manager_prefix peut être ingress, egress ou admin.
> La valeur de envoy_response_code_class peut être 1 (signifie 1xx), 2 (signifie 2xx), 3 (signifie 3xx), 4 (signifie 4xx), ou 5 (signifie 5xx).

Métriques Prometheus pour NGINX

Les métriques suivantes sont automatiquement collectées à partir de NGINX sur les clusters Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions	
`nginx_ingress_controller_nginx_process_cpu_seconds_total`	ClusterName,`Namespace` , Service	
`nginx_ingress_controller_success`	ClusterName,`Namespace` , Service	
`nginx_ingress_controller_requests`	ClusterName,`Namespace` , Service	
`nginx_ingress_controller_nginx_process_connections`	ClusterName,`Namespace` , Service	
`nginx_ingress_controller_nginx_process_connections_total`	ClusterName,`Namespace` , Service	
`nginx_ingress_cont`	ClusterName,`Namespace` , Service	

Nom de la métrique	Dimensions	
roller_nginx_process_resident_memory_bytes		
nginx_ingress_controller_config_last_reload_successful	ClusterName,Namespace , Service	
nginx_ingress_controller_requests	ClusterName,Namespace , Service, statut	

Métriques Prometheus pour Memcached

Les métriques suivantes sont automatiquement collectées à partir de Memcached sur les clusters Amazon EKS et Kubernetes.

Nom de la métrique	Dimensions	
memcached_current_items	ClusterName,Namespace , Service	
memcached_current_connections	ClusterName,Namespace , Service	
memcached_limit_bytes	ClusterName,Namespace , Service	

Nom de la métrique	Dimensions
memcached_current_bytes	ClusterName,Namespace , Service
memcached_written_bytes_total	ClusterName,Namespace , Service
memcached_read_bytes_total	ClusterName,Namespace , Service
memcached_items_evicted_total	ClusterName,Namespace , Service
memcached_items_reclaimed_total	ClusterName,Namespace , Service
memcached_commands_total	ClusterName,Namespace , Service ClusterName,Namespace , Service, commande ClusterName,Namespace , Service, statut, commande

Métriques Prometheus pour Java/JMX

Métriques collectées sur les clusters Amazon EKS et Kubernetes

Sur les clusters Amazon EKS et Kubernetes, Container Insights peut collecter les métriques Prometheus prédéfinies suivantes à partir de Java Virtual Machine (JVM), Java et Tomcat (Catalina) à l'aide de JMX Exporter. Pour plus d'informations, consultez prometheus/jmx_exporter sur Github.

Java/JMX sur les clusters Amazon EKS et Kubernetes

Nom de la métrique	Dimensions
jvm_classes_loaded	ClusterName , Namespace
jvm_threads_current	ClusterName , Namespace
jvm_threads_daemon	ClusterName , Namespace
java_lang_operatingsystem_totalswapspacesize	ClusterName , Namespace
java_lang_operatingsystem_systemcpuload	ClusterName , Namespace
java_lang_operatingsystem_processcpuload	ClusterName , Namespace
java_lang_operatingsystem_freeswapspacesize	ClusterName , Namespace
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , Namespace

Nom de la métrique	Dimensions	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , Namespace	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , Namespace	
java_lang_operatingsystem_availableprocessors	ClusterName , Namespace	
jvm_memory_bytes_used	ClusterName , Namespace , area	
jvm_memory_pool_bytes_used	ClusterName , Namespace , pool	

> **Note**
>
> Les valeurs de la dimension `area` peuvent être `heap` ou `nonheap`.
> Les valeurs de la dimension `pool` peuvent être `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` ou `Metaspace`.

Tomcat/JMX sur les clusters Amazon EKS et Kubernetes

Outre les métriques Java/JMX du tableau précédent, les métriques suivantes sont également collectées pour la charge de travail Tomcat.

Nom de la métrique	Dimensions	
`catalina_manager_activesessions`	`ClusterName` , `Namespace`	
`catalina_manager_rejectedsessions`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytesreceived`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_bytessent`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_requestcount`	`ClusterName` , `Namespace`	
`catalina_globalrequestprocessor_errorcount`	`ClusterName` , `Namespace`	

Nom de la métrique	Dimensions	
catalina_globalrequestprocessor_processingtime	ClusterName , Namespace	

Java/JMX sur les clusters Amazon ECS

Nom de la métrique	Dimensions	
jvm_classes_loaded	ClusterName , TaskDefinitionFamily	
jvm_threads_current	ClusterName , TaskDefinitionFamily	
jvm_threads_daemon	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_systemcpuload	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_processcpuload	ClusterName , TaskDefinitionFamily	

Nom de la métrique	Dimensions	
java_lang_operatingsystem_freeswapspacesize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_totalphysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_freephysicalmemorysize	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_openfiledescriptorcount	ClusterName , TaskDefinitionFamily	
java_lang_operatingsystem_availableprocessors	ClusterName , TaskDefinitionFamily	
jvm_memory_bytes_used	ClusterName , TaskDefinitionFamily, région	
jvm_memory_pool_bytes_used	ClusterName TaskDefinitionFamily, piscine	

> **Note**
>
> Les valeurs de la dimension `area` peuvent être `heap` ou `nonheap`.
> Les valeurs de la dimension `pool` peuvent être `Tenured Gen`, `Compress Class Space`, `Survivor Space`, `Eden Space`, `Code Cache` ou `Metaspace`.

Tomcat/JMX sur les clusters Amazon ECS

Outre les métriques Java/JMX du tableau précédent, les métriques suivantes sont également collectées pour l'application Tomcat sur les clusters Amazon ECS.

Nom de la métrique	Dimensions	
`catalina_manager_activesessions`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_manager_rejectedsessions`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestprocessor_bytesreceived`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestprocessor_bytessent`	`ClusterName` , `TaskDefinitionFamily`	
`catalina_globalrequestproce`	`ClusterName` , `TaskDefinitionFamily`	

Nom de la métrique	Dimensions	
ssor_requ estcount		
catalina_ globalreq uestproce ssor_erro rcount	ClusterName , TaskDefinitionFamily	
catalina_ globalreq uestproce ssor_proc essingtime	ClusterName , TaskDefinitionFamily	

Métriques Prometheus pour HAProxy

Les métriques suivantes sont automatiquement collectées à partir de HAProxy sur les clusters Amazon EKS et Kubernetes.

Les métriques collectées dépendent de la version de HAProxy Ingress que vous utilisez. Pour plus d'informations sur HAProxy Ingress et ses versions, consultez haproxy-ingress.

Nom de la métrique	Dimensions	Disponibilité
haproxy_b ackend_by tes_in_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_b ackend_by tes_out_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_b ackend_co	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress

Nom de la métrique	Dimensions	Disponibilité
nnection_ errors_total		
haproxy_b ackend_co nnections _total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_b ackend_cu rrent_sessions	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_b ackend_ht tp_respon ses_total	ClusterName , Namespace , Service, code, backend	Toutes les versions de HAProxy Ingress
haproxy_b ackend_status	ClusterName , Namespace , Service	Uniquement dans les versions 0.10 ou ultérieures de HAProxy Ingress
haproxy_b ackend_up	ClusterName , Namespace , Service	Uniquement dans les versions de HAProxy Ingress antérieures à 0.10
haproxy_f rontend_b ytes_in_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_f rontend_b ytes_out_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress

Nom de la métrique	Dimensions	Disponibilité
haproxy_frontend_connections_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_frontend_current_sessions	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_frontend_http_requests_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_frontend_http_responses_total	ClusterName , Namespace , Service, code, frontend	Toutes les versions de HAProxy Ingress
haproxy_frontend_request_errors_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress
haproxy_frontend_requests_denied_total	ClusterName , Namespace , Service	Toutes les versions de HAProxy Ingress

Note

Les valeurs de la dimension code peuvent être 1xx, 2xx, 3xx, 4xx, 5xx ou other. Les valeurs de la dimension backend peuvent être :

- `http-default-backend`, `http-shared-backend`, ou `httpsback-shared-backend` pour HAProxy Ingress version 0.0.27 ou antérieure.
- `_default_backend` pour les versions HAProxy Ingress ultérieures à 0.0.27.

Les valeurs de la dimension `frontend` peuvent être :

- `httpfront-default-backend`, `httpfront-shared-frontend`, ou `httpfronts` pour HAProxy Ingress version 0.0.27 ou antérieure.
- `_front_http` ou `_front_https` pour les versions HAProxy Ingress ultérieures à 0.0.27.

Affichage de vos métriques Prometheus

Vous pouvez contrôler toutes vos métriques Prometheus et définir des alertes sur ces métriques, y compris pour les métriques organisées et regroupées au préalable à partir de App Mesh, NGINX, Java/JMX, Memcached et HAProxy, ainsi que tout autre exportateur Prometheus configuré manuellement que vous avez peut-être ajouté. Pour plus d'informations sur la collecte de métriques à partir d'autres exportateurs Prometheus, consultez Didacticiel pour l'ajout d'une nouvelle cible de récupération Prometheus : métrique du serveur d'API Prometheus.

Dans la CloudWatch console, Container Insights fournit les rapports prédéfinis suivants :

- Pour les clusters Amazon EKS et Kubernetes, il existe des rapports prédéfinis pour App Mesh, NGINX, HAPROXY, Memcached et Java/JMX.
- Pour les clusters Amazon ECS, il existe des rapports prédéfinis pour App Mesh et Java/JMX.

Container Insights fournit également des tableaux de bord personnalisés pour chacune des applications à partir desquelles Container Insights collecte des métriques organisées. Vous pouvez télécharger ces tableaux de bord sur GitHub

Pour afficher toutes vos métriques Prometheus

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Dans la liste des espaces de noms, choisissez ContainerInsights/Prometheus ou ECS/ / Prometheus. ContainerInsights

4. Choisissez l'un des ensembles de dimensions de la liste suivante. Cochez ensuite la case en regard des métriques que vous souhaitez afficher.

Pour afficher les rapports prédéfinis sur vos métriques Prometheus

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Surveillance des performances.
3. Dans la liste déroulante située en haut de la page, choisissez l'une des options de Prometheus.

   Dans l'autre liste déroulante, choisissez un cluster à afficher

Nous avons également fourni des tableaux de bord personnalisés pour NGINX, App Mesh, Memcached, HAProxy et Java/JMX.

Pour utiliser un tableau de bord personnalisé fourni par Amazon

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 Créer un tableau de bord. Entrez un nom pour le nouveau tableau de bord, puis choisissez Créer un tableau de bord.
4. Dans Ajouter à ce tableau de bord, choisissez Annuler.
5. Choisissez Actions, View/Edit Attributes (Afficher/Modifier la source).
6. Téléchargez l'un des fichiers JSON suivants :

   - Source de tableau de bord personnalisé NGINX sur Github.
   - Source de tableau de bord personnalisé App Mesh sur Github.
   - Source de tableau de bord personnalisé Memcached sur Github
   - Source de tableau de bord personnalisé HAProxy-Ingress sur Github
   - Source de tableau de bord Java/JMX personnalisé sur Github.
7. Ouvrez le fichier JSON que vous avez téléchargé avec un éditeur de texte et apportez les modifications suivantes :

   - Remplacez toutes les chaînes `{{YOUR_CLUSTER_NAME}}` par le nom exact de votre cluster. Assurez-vous de ne pas ajouter d'espaces avant ou après le texte.

- Remplacez toutes les `{{YOUR_REGION}}` chaînes par la AWS région dans laquelle votre cluster est exécuté. Par exemple, **`us-west-1`** assurez-vous de ne pas ajouter d'espaces avant ou après le texte.
- Remplacez toutes les chaînes `{{YOUR_NAMESPACE}}` par l'espace de noms exact de votre application.
- Remplacez toutes les chaînes `{{YOUR_SERVICE_NAME}}` par le nom de service exact de votre application. Par exemple, **`haproxy-haproxy-ingress-controller-metrics`**

8. Copiez l'intégralité du blob JSON et collez-le dans la zone de texte de la CloudWatch console, en remplaçant ce qui s'y trouve déjà.
9. Choisissez Mettre à jour, Enregistrer le tableau de bord.

Résolution des problèmes rencontrés avec les métriques Prometheus

Cette section fournit de l'aide pour résoudre les problèmes de configuration des métriques Prometheus.

Rubriques

Résolution des problèmes rencontrés avec les métriques Prometheus sur Amazon ECS

Cette section fournit de l'aide pour résoudre les problèmes de configuration des métriques Prometheus sur les clusters Amazon ECS.

Je ne vois pas les métriques de Prometheus envoyées à Logs CloudWatch

Les métriques Prometheus doivent être ingérées en tant qu'évènements de journaux dans le groupe de journaux /aws/ecs/containerinsights/cluster-name/Prometheus. Si le groupe de journaux n'est pas créé ou si les métriques Prometheus ne sont pas envoyées au groupe de journaux, vous devez d'abord vérifier si les cibles Prometheus ont été correctement découvertes par l'agent. CloudWatch Vérifiez ensuite le groupe de sécurité et les paramètres d'autorisation de l' CloudWatchagent. Les étapes suivantes vous guident pour réaliser le débogage.

Étape 1 : activer le mode de débogage de CloudWatch l'agent

Tout d'abord, passez l' CloudWatch agent en mode de débogage en ajoutant les lignes en gras suivantes à votre fichier AWS CloudFormation modèle, `cwagent-ecs-prometheus-metric-for-bridge-host.yaml` ou`cwagent-ecs-prometheus-metric-for-awsvpc.yaml`. Ensuite, enregistrez le fichier.

```
cwagentconfig.json: |
    {
      "agent": {
        "debug": true
      },
      "logs": {
        "metrics_collected": {
```

Créez un nouvel ensemble de AWS CloudFormation modifications par rapport à la pile existante. Définissez les autres paramètres du changeset sur les mêmes valeurs que dans votre AWS CloudFormation pile existante. L'exemple suivant concerne un CloudWatch agent installé dans un cluster Amazon ECS utilisant le type de lancement EC2 et le mode réseau de pont.

```
ECS_NETWORK_MODE=bridge
 CREATE_IAM_ROLES=True
ECS_TASK_ROLE_NAME=your_selected_ecs_task_role_name
ECS_EXECUTION_ROLE_NAME=your_selected_ecs_execution_role_name
NEW_CHANGESET_NAME=your_selected_ecs_execution_role_name

aws cloudformation create-change-set --stack-name CWAgent-Prometheus-ECS-
${ECS_CLUSTER_NAME}-EC2-${ECS_NETWORK_MODE} \
    --template-body file://cwagent-ecs-prometheus-metric-for-bridge-host.yaml \
    --parameters ParameterKey=ECSClusterName,ParameterValue=$ECS_CLUSTER_NAME \
                 ParameterKey=CreateIAMRoles,ParameterValue=$CREATE_IAM_ROLES \
                 ParameterKey=ECSNetworkMode,ParameterValue=$ECS_NETWORK_MODE \
                 ParameterKey=TaskRoleName,ParameterValue=$ECS_TASK_ROLE_NAME \
                 ParameterKey=ExecutionRoleName,ParameterValue=$ECS_EXECUTION_ROLE_NAME
 \
    --capabilities CAPABILITY_NAMED_IAM \
    --region $AWS_REGION \
    --change-set-name $NEW_CHANGESET_NAME
```

Accédez à la AWS CloudFormation console pour consulter le nouvel ensemble de modifications,`$NEW_CHANGESET_NAME`. Une modification doit être appliquée à la ressource CW AgentConfig SSMParameter. Exécutez l'ensemble de modifications et redémarrez la tâche de l' CloudWatch agent en saisissant les commandes suivantes.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 0 \
--service your_service_name_here \
--region $AWS_REGION
```

Patientez environ 10 secondes, puis saisissez la commande suivante.

```
aws ecs update-service --cluster $ECS_CLUSTER_NAME \
--desired-count 1 \
--service your_service_name_here \
--region $AWS_REGION
```

Étape 2 : vérification des journaux de découverte de service ECS

La définition de tâche ECS de l' CloudWatch agent active les journaux par défaut dans la section ci-dessous. Les journaux sont envoyés à CloudWatch Logs dans le groupe de journaux /ecs/ ecs-cwagent-prometheus.

```
LogConfiguration:
  LogDriver: awslogs
    Options:
      awslogs-create-group: 'True'
      awslogs-group: "/ecs/ecs-cwagent-prometheus"
      awslogs-region: !Ref AWS::Region
      awslogs-stream-prefix: !Sub 'ecs-${ECSLaunchType}-awsvpc'
```

Filtrez les journaux par la chaîne `ECS_SD_Stats` pour obtenir les métriques liées à la découverte de service ECS, comme illustré dans l'exemple suivant.

```
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeContainerInstances: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeInstancesRequest: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_DescribeTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: AWSCLI_ListTasks: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: Exporter_DiscoveredTargetCount: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_EC2MetaData: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Get_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_ContainerInstance: 1
2020-09-1T01:53:14Z D! ECS_SD_Stats: LRUCache_Size_TaskDefinition: 2
2020-09-1T01:53:14Z D! ECS_SD_Stats: Latency: 43.399783ms
```

La signification de chaque métrique pour un cycle de découverte de service ECS particulier est la suivante :

- AWSCLI_DescribeContainerInstances— le nombre d'appels d'`ECS::DescribeContainerInstances`API effectués.
- AWSCLI_DescribeInstancesRequest— le nombre d'appels d'`ECS::DescribeInstancesRequest`API effectués.
- AWSCLI_DescribeTaskDefinition— le nombre d'appels d'`ECS::DescribeTaskDefinition`API effectués.
- AWSCLI_DescribeTasks— le nombre d'appels d'`ECS::DescribeTasks`API effectués.
- AWSCLI_ListTasks— le nombre d'appels d'`ECS::ListTasks`API effectués.
- ExporterDiscoveredTargetCount— le nombre de cibles Prometheus découvertes et exportées avec succès dans le fichier de résultats cible contenu dans le conteneur.
- LruCache_get_ec2 MetaData — le nombre de fois où les métadonnées des instances de conteneur ont été extraites du cache.
- LruCache_get_ TaskDefinition — le nombre de fois où les métadonnées de définition de tâche ECS ont été extraites du cache.
- LruCache_Size_ ContainerInstance — le nombre de métadonnées d'une instance de conteneur unique mises en cache en mémoire.
- LruCache_Size_ TaskDefinition — le nombre de définitions de tâches ECS uniques mises en cache en mémoire.
- Latency – combien de temps prend le cycle de découverte de service.

Vérifiez la valeur de `ExporterDiscoveredTargetCount` pour voir si les cibles Prometheus découvertes correspondent à vos attentes. Si ce n'est pas le cas, les raisons possibles sont les suivantes :

- La configuration de la découverte de service ECS peut ne pas correspondre aux paramètres de votre application. Pour la découverte de services basée sur des étiquettes docker, il est possible que l'étiquette docker nécessaire ne soit pas configurée dans l'agent CloudWatch pour les détecter automatiquement dans vos conteneurs cibles. Pour la découverte de services basée sur l'expression régulière ARN de définition de tâche ECS, le paramètre regex de l' CloudWatch agent peut ne pas correspondre à la définition de tâche de votre application.

- Le rôle de tâche ECS de l' CloudWatch agent n'est peut-être pas autorisé à récupérer les métadonnées des tâches ECS. Vérifiez que les autorisations de lecture seule suivantes ont été accordées à l' CloudWatch agent :
  - `ec2:DescribeInstances`
  - `ecs:ListTasks`
  - `ecs:DescribeContainerInstances`
  - `ecs:DescribeTasks`
  - `ecs:DescribeTaskDefinition`

Étape 3 : Vérification de la connexion réseau et de la politique de rôle de tâche ECS

Si aucun événement de journal n'est toujours envoyé au groupe de CloudWatch journaux Logs cible, même si la valeur de `Exporter_DiscoveredTargetCount` indique que des cibles Prometheus ont été découvertes, cela peut être dû à l'un des facteurs suivants :

- L' CloudWatch agent ne sera peut-être pas en mesure de se connecter aux ports cibles de Prometheus. Vérifiez le paramètre du groupe de sécurité sous-jacent à l' CloudWatch agent. L'adresse IP privée doit permettre à l' CloudWatch agent de se connecter aux ports de l'exportateur Prometheus.
- Le rôle de tâche ECS de l' CloudWatch agent n'a peut-être pas la politique CloudWatchAgentServerPolicygérée. Le rôle de tâche ECS de l' CloudWatch agent doit respecter cette politique pour pouvoir envoyer les métriques Prometheus sous forme d'événements de journal. Si vous avez utilisé l'exemple de AWS CloudFormation modèle pour créer automatiquement les rôles IAM, le rôle de tâche ECS et le rôle d'exécution ECS sont dotés du moindre privilège pour effectuer la surveillance Prometheus.

Résolution des problèmes liés aux métriques Prometheus sur les clusters Amazon EKS et Kubernetes

Cette section fournit de l'aide pour résoudre les problèmes de configuration des métriques Prometheus sur les clusters Amazon EKS et Kubernetes.

Étapes de résolution des problèmes générales sur Amazon EKS

Pour vérifier que l' CloudWatch agent est en cours d'exécution, entrez la commande suivante.

```
kubectl get pod -n amazon-cloudwatch
```

La sortie doit inclure une ligne avec `cwagent-prometheus-id` dans la colonne `NAME` et `Running` dans la colonne `STATUS column.`

Pour afficher des détails sur le pod en cours d'exécution, entrez la commande suivante. Remplacez *pod-name* par le nom complet de votre pod dont le nom commence par `cw-agent-prometheus`.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

Si CloudWatch Container Insights est installé, vous pouvez utiliser CloudWatch Logs Insights pour interroger les journaux de l' CloudWatch agent qui collecte les métriques Prometheus.

Pour interroger les journaux d'application

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez CloudWatch Logs Insights.
3. Sélectionnez le groupe de journaux pour les journaux d'application, /aws/containerinsights/***nom-du-cluster***/application
4. Remplacez l'expression de requête de recherche par la requête suivante, puis choisissez Exécuter la requête

```
fields ispresent(kubernetes.pod_name) as haskubernetes_pod_name, stream,
 kubernetes.pod_name, log |
filter haskubernetes_pod_name and kubernetes.pod_name like /cwagent-prometheus
```

Vous pouvez également confirmer que les métriques et les métadonnées de Prometheus sont CloudWatch ingérées sous forme d'événements Logs.

Pour vérifier que les données Prometheus sont ingérées

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez CloudWatch Logs Insights.
3. Sélectionnez /aws/containerinsights/***nom-du-cluster***/prometheus
4. Remplacez l'expression de requête de recherche par la requête suivante, puis choisissez Exécuter la requête

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

Journalisation des métriques Prometheus ignorées

Cette version ne collecte pas les métriques Prometheus de type histogramme. Vous pouvez utiliser l' CloudWatch agent pour vérifier si des métriques Prometheus sont supprimées car il s'agit de métriques d'histogrammes. Vous pouvez également enregistrer une liste des 500 premières métriques Prometheus supprimées et non envoyées car il s'agit de métriques CloudWatch d'histogrammes.

Pour voir si des métriques sont ignorées, entrez la commande suivante :

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

Si des métriques sont ignorées, les lignes suivantes s'affichent dans le fichier `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`.

```
I! Drop Prometheus metrics with unsupported types. Only Gauge, Counter and Summary are
 supported.
I! Please enable CWAgent debug mode to view the first 500 dropped metrics
```

Si ces lignes s'affichent et si vous souhaitez savoir quelles métriques sont ignorées, procédez comme suit.

Pour enregistrer une liste des métriques Prometheus ignorées

1. Passez l' CloudWatch agent en mode de débogage en ajoutant les lignes en gras suivantes à votre `prometheus-k8s.yaml` fichier `prometheus-eks.yaml` ou en enregistrant le fichier.

   ```
   {
         "agent": {
           "debug": true
         },
   ```

   Cette section du fichier doit alors ressembler à ceci :

   ```
   cwagentconfig.json: |
       {
         "agent": {
           "debug": true
         },
         "logs": {
   ```

```
        "metrics_collected": {
```

2. Réinstallez l' CloudWatch agent pour activer le mode de débogage en saisissant les commandes suivantes :

```
kubectl delete deployment cwagent-prometheus -n amazon-cloudwatch
kubectl apply -f prometheus.yaml
```

   Les métriques supprimées sont enregistrées dans le module de l' CloudWatch agent.

3. Pour récupérer les journaux depuis le module de l' CloudWatch agent, entrez la commande suivante :

```
kubectl logs -l "app=cwagent-prometheus" -n amazon-cloudwatch --tail=-1
```

   Ou, si la journalisation Fluentd de Container Insights est installée, les journaux sont également enregistrés dans le groupe de journaux Logs CloudWatch ***/aws/containerinsights/ cluster_name /application.***

   Pour interroger ces journaux, vous pouvez suivre les étapes nécessaires pour interroger les journaux d'application dans Étapes de résolution des problèmes générales sur Amazon EKS.

Où sont ingérées CloudWatch les métriques Prometheus lorsque Logs enregistre les événements ?

L' CloudWatch agent crée un flux de journal pour chaque configuration de tâche Prometheus Scrape. Par exemple, dans les fichiers `prometheus-eks.yaml` et `prometheus-k8s.yaml`, la ligne `job_name: 'kubernetes-pod-appmesh-envoy'` récupère les métriques App Mesh. La cible Prometheus est définie en tant que `kubernetes-pod-appmesh-envoy`. Ainsi, toutes les métriques App Mesh Prometheus sont ingérées sous forme d'événements Logs dans le kubernetes-pod-appmesh-envoyflux de journaux sous le groupe de journaux CloudWatch nommé /AWS/ ContainerInsights/cluster-name/Prometheus.

Je ne vois pas les métriques Amazon EKS ou Kubernetes Prometheus dans les métriques CloudWatch

Tout d'abord, assurez-vous que les métriques Prometheus sont ingérées en tant qu'événements de journaux dans le groupe de journaux /aws/containerinsights/cluster-name/Prometheus. Utilisez les informations de Où sont ingérées CloudWatch les métriques Prometheus lorsque Logs enregistre les événements ? pour vous aider à vérifier le flux de journaux cible. Si le flux de journaux n'est pas créé

ou s'il n'y a pas de nouveaux événements de journaux dans le flux de journaux, vérifiez les points suivants :

- Vérifiez que les points de terminaison de l'exportateur de métriques Prometheus sont correctement configurés
- Vérifiez que les configurations de scraping de Prometheus dans `config map: cwagent-prometheus` la section du fichier YAML de CloudWatch l'agent sont correctes. La configuration doit être la même que dans un fichier de configuration Prometheus. Pour plus d'informations, consultez <scrape_config> dans la documentation Prometheus.

Si les métriques Prometheus sont correctement ingérées sous forme d'événements de journal, vérifiez que les paramètres de format de métrique intégrés sont ajoutés aux événements du journal pour générer les métriques. CloudWatch

```
"CloudWatchMetrics":[
   {
      "Metrics":[
         {
            "Name":"envoy_http_downstream_cx_destroy_remote_active_rq"
         }
      ],
      "Dimensions":[
         [
            "ClusterName",
            "Namespace"
         ]
      ],
      "Namespace":"ContainerInsights/Prometheus"
   }
],
```

Pour plus d'informations sur le format de métrique intégrée, consultez Spécifications : format de métrique intégrée .

Si aucun format métrique n'est intégré dans le journal des événements, vérifiez que la `metric_declaration` section est correctement configurée dans la `config map: prometheus-cwagentconfig` section du fichier YAML d'installation de l' CloudWatch agent. Pour plus d'informations, consultez Didacticiel pour l'ajout d'une nouvelle cible de récupération Prometheus : métrique du serveur d'API Prometheus.

Intégration à Application Insights

Amazon CloudWatch Application Insights vous aide à surveiller vos applications, à identifier et à configurer les indicateurs clés, les journaux et les alarmes pour l'ensemble de vos ressources applicatives et de votre infrastructure technologique. Pour plus d'informations, consultez Informations sur les CloudWatch applications Amazon.

Vous pouvez activer Application Insights pour recueillir des données supplémentaires à partir de vos applications et microservices conteneurisés. Si vous ne l'avez pas déjà fait, vous pouvez l'activer en sélectionnant Auto-configure Application Insights (Configuration automatique d'Application Insights) sous la vue des performances dans le tableau de bord Container Insights.

Si vous avez déjà configuré CloudWatch Application Insights pour surveiller vos applications conteneurisées, le tableau de bord Application Insights apparaît sous le tableau de bord Container Insights.

Pour plus d'informations sur Application Insight et les applications conteneurisées, consultezActiver la surveillance des ressources Application Insights pour Amazon ECS et Amazon EKS.

Consulter les événements du cycle de vie d'Amazon ECS dans Container Insights

Vous pouvez consulter les événements du cycle de vie d'Amazon ECS dans la console Container Insights. Cela vous aide à corréler vos métriques, journaux et événements de conteneurs en une seule vue pour vous donner une visibilité opérationnelle plus complète.

Les événements comprennent des événements de changement d'état d'instance de conteneur, des événements de changement d'état de tâche et des événements d'action de service. Ils sont automatiquement envoyés par Amazon ECS à Amazon EventBridge et sont également collectés CloudWatch sous forme de journal d'événements. Pour plus d'informations sur ces événements, consultez Événements Amazon ECS.

La tarification standard de Container Insights s'applique aux événements Amazon ECS Lifecycle. Pour en savoir plus, consultez Tarification Amazon CloudWatch.

Pour configurer le tableau des événements du cycle de vie et créer des règles pour un cluster, vous devez disposer des autorisations `events:PutRule`, `events:PutTargets` et `logs:CreateLogGroup`. Vous devez également vous assurer qu'il existe une politique de

ressources qui permet de EventBridge créer le flux de journaux et d'envoyer les CloudWatch journaux à Logs. Si cette politique de ressources n'existe pas, vous pouvez saisir la commande suivante pour la créer :

```
aws --region region logs put-resource-policy --policy-name 'EventBridgeCloudWatchLogs'
 --policy-document '{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
            "Resource": "arn:aws:logs:region:account-id:log-group:/aws/events/ecs/
containerinsights/*:*",
            "Sid": "TrustEventBridgeToStoreECSLifecycleLogEvents"
        }
    ],
    "Version": "2012-10-17"
}'
```

Vous pouvez utiliser la commande suivante pour vérifier si vous disposez déjà de cette politique et pour confirmer que son rattachement a fonctionné correctement.

```
aws logs describe-resource-policies --region region --output json
```

Pour afficher le tableau des événements du cycle de vie, vous devez disposer des autorisations `events:DescribeRule`, `events:ListTargetsByRule` et `logs:DescribeLogGroups`.

Pour consulter les événements du cycle de vie d'Amazon ECS dans la console CloudWatch Container Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Choisissez Insights (Informations), Container Insights.
3. Choisissez Afficher les tableaux de bord des performances.
4. Dans la liste déroulante suivante, choisissez soit ECS Clusters (Clusters ECS), soit ECS Services (Services ECS), soit ECS Tasks (Tâches ECS).

5. Si vous avez choisi ECS Services (Services ECS) ou ECS Tasks (Tâches ECS) à l'étape précédente, choisissez l'onglet Lifecycle events (Événements du cycle de vie).
6. Au bas de la page, si vous voyez Configurer les événements du cycle de vie, choisissez-le pour créer des EventBridge règles pour votre cluster.

   Les événements sont affichés sous les volets Container Insights et au-dessus de la section Application Insights. Pour exécuter des analyses supplémentaires et créer des visualisations additionnelles sur ces événements, choisissez View in Logs Insights (Afficher dans Logs Insights) dans le tableau Lifecycle Events (Événements du cycle de vie).

Résolution des problèmes liés à Container Insights

Les sections suivantes peuvent aider si vous rencontrez des problèmes avec Container Insights.

Échec du déploiement sur Amazon EKS ou Kubernetes

Si l'agent ne se déploie pas correctement sur un cluster Kubernetes, essayez ce qui suit :

- Exécutez la commande suivante pour obtenir la liste des pods.

```
kubectl get pods -n amazon-cloudwatch
```

- Exécutez la commande suivante et vérifiez les événements au bas de la sortie.

```
kubectl describe pod pod-name -n amazon-cloudwatch
```

- Exécutez la commande suivante pour vérifier les journaux.

```
kubectl logs pod-name -n amazon-cloudwatch
```

Panique non autorisée : Impossible de récupérer les données cadvisor à partir de kubelet

Si votre déploiement échoue avec l'erreur `Unauthorized panic: Cannot retrieve cadvisor data from kubelet`, il se peut que le mode d'autorisation Webhook ne soit pas activé pour votre Kubelet. Ce mode est obligatoire pour Container Insights. Pour plus d'informations, consultez Vérifiez les conditions préalables.

Déploiement de Container Insights sur un cluster supprimé puis recréé sur Amazon ECS

Si vous supprimez un cluster Amazon ECS existant sur lequel Container Insights n'est pas activé et que vous le recréez avec le même nom, vous ne pouvez pas activer Container Insights sur ce nouveau cluster au moment où vous le recréez. Vous pouvez l'activer en le recréant, puis en entrant la commande suivante :

```
aws ecs update-cluster-settings --cluster myCICluster --settings
 name=containerInsights,value=enabled
```

Erreur de point de terminaison non valide

Si un message d'erreur semblable au suivant s'affiche, vérifiez que vous avez remplacé tous les espaces réservés tels que *cluster-name* et *region-name* dans les commandes que vous utilisez, par les informations correctes pour votre déploiement.

```
"log": "2020-04-02T08:36:16Z E! cloudwatchlogs: code: InvalidEndpointURL, message:
 invalid endpoint uri, original error: &url.Error{Op:\"parse\", URL:\"https://
logs.{{region_name}}.amazonaws.com/\", Err:\"{\"}, &awserr.baseError{code:
\"InvalidEndpointURL\", message:\"invalid endpoint uri\", errs:[]error{(*url.Error)
(0xc0008723c0)}}\n",
```

Les métriques ne s'affichent pas dans la console

Si vous ne voyez aucune métrique de Container Insights dans le AWS Management Console, assurez-vous d'avoir terminé la configuration de Container Insights. Les métriques n'apparaissent pas avant que Container Insights n'ait été configuré complètement. Pour plus d'informations, consultez . Configuration de Container Insights.

Métriques de pod manquantes sur Amazon EKS ou Kubernetes après la mise à niveau du cluster

Cette section peut être utile si toutes les métriques du pod ou certaines d'entre elles sont manquantes après le déploiement de l' CloudWatch agent en tant que daemonset sur un cluster nouveau ou mis à niveau, ou si un journal des erreurs s'affiche avec le message. `W! No pod metric collected`

Ces erreurs peuvent être causées par des changements dans l'exécution du conteneur, tels que containerd ou le pilote docker systemd cgroup. Vous pouvez généralement résoudre ce problème en

mettant à jour votre manifeste de déploiement afin que le socket containerd de l'hôte soit monté dans le conteneur. Consultez l’exemple suivant:

```
# For full example see https://github.com/aws-samples/amazon-cloudwatch-container-
insights/blob/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/
container-insights-monitoring/cwagent/cwagent-daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: cloudwatch-agent
  namespace: amazon-cloudwatch
spec:
  template:
    spec:
      containers:
        - name: cloudwatch-agent
# ...
          # Don't change the mountPath
          volumeMounts:
# ...
            - name: dockersock
              mountPath: /var/run/docker.sock
              readOnly: true
            - name: varlibdocker
              mountPath: /var/lib/docker
              readOnly: true
            - name: containerdsock # NEW mount
              mountPath: /run/containerd/containerd.sock
              readOnly: true
# ...
      volumes:
# ...
        - name: dockersock
          hostPath:
            path: /var/run/docker.sock
        - name: varlibdocker
          hostPath:
            path: /var/lib/docker
        - name: containerdsock # NEW volume
          hostPath:
            path: /run/containerd/containerd.sock
```

Aucune métrique de pod lors de l'utilisation de Bottlerocket pour Amazon EKS

Bottlerocket est un système d'exploitation open source basé sur Linux qui est spécialement conçu par AWS pour exécuter des conteneurs.

Bottlerocket utilise un chemin `containerd` sur l'hôte, vous devez donc modifier les volumes à son emplacement. Dans le cas contraire, une erreur s'affiche dans les journaux qui incluent `W! No pod metric collected`. Consultez l'exemple suivant.

```
volumes:
  # ...
    - name: containerdsock
      hostPath:
        # path: /run/containerd/containerd.sock
        # bottlerocket does not mount containerd sock at normal place
        # https://github.com/bottlerocket-os/bottlerocket/
commit/91810c85b83ff4c3660b496e243ef8b55df0973b
        path: /run/dockershim.sock
```

Aucune métrique de FileSystem de conteneur lors de l'utilisation de l'exécution containerd pour Amazon EKS ou Kubernetes

C'est un problème connu et les contributeurs de la communauté travaillent sur cette question. Pour plus d'informations, consultez les sections Métrique d'utilisation du disque pour les containers et Les métriques du système de fichiers conteneur ne sont pas prises en charge par Cadvisor pour containerd on. GitHub

Augmentation inattendue du volume de logs due à l' CloudWatch agent lors de la collecte des métriques Prometheus

Il s'agit d'une régression introduite dans la version 1.247347.6b250880 de l'agent. CloudWatch Cette régression a déjà été corrigée dans les versions plus récentes de l'agent. Son impact s'est limité aux scénarios dans lesquels les clients collectaient les journaux de l' CloudWatch agent lui-même et utilisaient également Prometheus. Pour plus d'informations, voir [prometheus] L'agent imprime toutes les métriques supprimées lors de la connexion. GitHub

Dernière image docker mentionnée dans les notes de mise à jour introuvables depuis Dockerhub

Nous mettons à jour la note de mise à jour et l'identification sur Github avant de démarrer la version réelle en interne. 1 à 2 semaines sont généralement nécessaires pour voir la dernière image docker sur les registres après avoir élevé le numéro de version sur Github. Il n'y a pas de publication nocturne pour l'image du conteneur de l' CloudWatch agent. Vous pouvez créer l'image directement à partir de la source à l'emplacement suivant : https://github.com/aws/ amazon-cloudwatch-agent /tree/ main//amazon-cloudwatch-container-insightscloudwatch-agent-dockerfile

CrashLoopBackoff erreur sur l' CloudWatch agent

Si un `CrashLoopBackOff` message d'erreur s'affiche pour l' CloudWatch agent, assurez-vous que vos autorisations IAM sont correctement définies. Pour plus d'informations, consultez Vérifiez les conditions préalables.

CloudWatch agent ou module Fluentd bloqué en attente

Si un CloudWatch agent ou un pod Fluentd est bloqué `Pending` ou s'il présente une `FailedScheduling` erreur, déterminez si vos nœuds disposent de suffisamment de ressources de calcul en fonction du nombre de cœurs et de la quantité de RAM requis par les agents. Entrez la commande suivante pour décrire le pod :

```
kubectl describe pod cloudwatch-agent-85ppg -n amazon-cloudwatch
```

Création de votre propre image Docker d' CloudWatch agent

Vous pouvez créer votre propre image Docker d' CloudWatch agent en vous référant au Dockerfile situé à l'adresse https://github.com/aws-samples/ amazon-cloudwatch-container-insights /blob/ latest/ /Dockerfile. cloudwatch-agent-dockerfile

Le fichier Docker prend en charge la création d'images multi-architectures directement à l'aide de `docker buildx`.

Déploiement d'autres fonctionnalités d' CloudWatch agent dans vos conteneurs

Vous pouvez déployer des fonctionnalités de surveillance supplémentaires dans vos conteneurs à l'aide de l' CloudWatch agent. Les principales fonctions sont notamment :

- Format de métriques intégré – Pour en savoir plus, consultez Intégration de métriques dans les journaux.
- StatsD – Pour en savoir plus, consultez Récupération de métriques personnalisées avec StatsD .

Les instructions et les fichiers nécessaires se trouvent GitHub aux emplacements suivants :

- Pour les conteneurs Amazon ECS, consultez Exemples de définitions de tâches Amazon ECS en fonction des modes de déploiement.
- Pour les conteneurs Amazon EKS et Kubernetes, consultez Exemples de fichiers YAML Kubernetes en fonction des modes de déploiement.

Aperçu Lambda

CloudWatch Lambda Insights est une solution de surveillance et de dépannage pour les applications sans serveur exécutées sur. AWS Lambda La solution collecte, agrège et résume les métriques de niveau système, notamment le temps de CPU, la mémoire, le disque et le réseau. Elle collecte, agrège et résume également des informations de diagnostic telles que des démarrages à froid et des arrêts de rôle de travail Lambda pour vous aider à circonscrire des problèmes liés à vos fonctions Lambda, ainsi qu'à les résoudre rapidement.

Lambda Insights utilise une nouvelle extension CloudWatch Lambda, fournie sous forme de couche Lambda. Lorsque vous installez cette extension sur une fonction Lambda, elle collecte des métriques au niveau du système et émet un seul événement de journal des performances pour chaque appel de cette fonction Lambda. CloudWatch utilise le formatage des métriques intégré pour extraire les métriques des événements du journal.

Pour plus d'informations sur les extensions Lambda, consultez la section Utilisation AWS Lambda des extensions. Pour plus d'informations sur le format de métrique intégrée, consultez Intégration de métriques dans les journaux.

Vous pouvez utiliser Lambda Insights avec n'importe quelle fonction Lambda qui utilise un runtime Lambda prenant en charge les extensions Lambda. Pour obtenir la liste de ces runtimes, consultez API Extensions Lambda.

Tarification

Pour chaque fonction Lambda activée pour Lambda Insights, vous ne payez que votre utilisation des métriques et des journaux. Pour un exemple de tarification, consultez Amazon CloudWatch Pricing.

Vous êtes facturé pour le temps d'exécution consommé par l'extension Lambda, par incréments de 1 ms. Pour plus d'informations sur la tarification Lambda, consultez Tarification AWS Lambda.

Mise en route avec Lambda Insights

Pour activer Lambda Insights sur une fonction Lambda, vous pouvez utiliser la touche à bascule en un clic dans la console Lambda. Vous pouvez également utiliser le AWS CLI, AWS CloudFormation, la AWS Serverless Application Model CLI ou le AWS Cloud Development Kit (AWS CDK).

Les sections suivantes fournissent des instructions détaillées pour la réalisation de ces étapes.

Rubriques

Versions disponibles de l'extension Lambda Insights

Cette section répertorie les versions de l'extension Lambda Insights et les ARN à utiliser pour ces extensions dans chaque région. AWS

Rubriques

Plateformes x86-64

Cette section répertorie les versions de l'extension Lambda Insights pour les plateformes x86-64, ainsi que les ARN à utiliser pour ces extensions dans chaque région. AWS

> **Important**
>
> Les extensions Lambda Insights 1.0.317.0 et versions ultérieures ne sont pas compatibles avec Amazon Linux 1.

1,0317,0

La version 1.0.317.0 inclut la suppression du support pour la plate-forme Amazon Linux 1 et des corrections de bugs. Il inclut également le soutien aux AWS GovCloud (US) régions.

ARN pour la version 1.0.317.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:52`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:52`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:52`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:52`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:43`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:43`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:25`

Région	ARN
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:29`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:20`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:50`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:33`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:51`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:52`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:52`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:79`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:51`
Canada Ouest (Calgary)	`arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:12`
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:42`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:42`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:52`

Région	ARN
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:52`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:52`
Europe (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:43`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:51`
Europe (Espagne)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:27`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:49`
Europe (Zurich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:26`
Israël (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:20`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:43`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:26`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:51`
AWS GovCloud (USA Est)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension:19`
AWS GovCloud (US-Ouest)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension:19`

1.0.295.0

La version 1.0.295.0 inclut des mises à jour des dépendances pour tous les environnements d'exécution compatibles.

ARN pour la version 1.0.295.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:51`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:51`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:51`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:51`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:42`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:42`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:24`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:28`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:19`

Région	ARN
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:49`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:32`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:50`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:51`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:51`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:78`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:50`
Canada Ouest (Calgary)	`arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:11`
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:41`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:41`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:51`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:51`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:51`

Région	ARN
Europe (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:42`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:50`
Europe (Espagne)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:26`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:48`
Europe (Zurich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:25`
Israël (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:19`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:42`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:25`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:50`

1.0.275.0

La version 1.0.275.0 inclut d'importantes mises à jour de dépendances pour tous les environnements d'exécution compatibles.

ARN pour la version 1.0.275.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:49`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:49`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:49`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:49`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:40`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:40`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:22`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:26`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:17`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:47`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:30`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:48`

Région	ARN
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:49
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:49
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:76
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:48
Canada Ouest (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:9
Chine (Beijing)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:39
Chine (Ningxia) ;	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:39
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:49
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:49
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:49
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:40
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:48
Europe (Espagne)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:24

Région	ARN
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:46`
Europe (Zurich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:23`
Israël (Tel Aviv)	`arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:17`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:40`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:23`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:48`

1.0.273.0

La version 1.0.273.0 inclut des corrections de bogues importantes pour tous les environnements d'exécution compatibles et ajoute le support pour Canada West (Calgary).

ARN pour la version 1.0.273.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:45`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:45`

Région	ARN
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:45`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:45`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:35`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:35`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:17`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:21`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:12`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:43`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:26`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:44`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:45`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:45`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:72`

Région	ARN
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:44
Canada Ouest (Calgary)	arn:aws:lambda:ca-west-1:946466191631:layer:LambdaInsightsExtension:4
Chine (Beijing)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:36
Chine (Ningxia) ;	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:36
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:45
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:45
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:45
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:35
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:44
Europe (Espagne)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:19
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:42
Europe (Zurich)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:17
Israël (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:12

Région	ARN
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:35`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:18`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:44`

1,0,229,0

La version 1.0.229.0 inclut des correctifs de bogues importants pour toutes les exécutions compatibles et prend désormais en charge la région d'Israël (Tel Aviv).

ARN pour la version 1.0.229.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:38`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:38`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:38`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:38`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:28`

Région	ARN
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:28`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:10`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:14`
Asie-Pacifique (Melbourne)	`arn:aws:lambda:ap-southeast-4:158895979263:layer:LambdaInsightsExtension:5`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:36`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:19`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:37`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:38`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:38`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:60`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:37`
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:29`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:29`

Région	ARN
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:38
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:38
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:38
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:28
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:37
Europe (Espagne)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:12
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:35
Europe (Zurich)	arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:11
Israël (Tel Aviv)	arn:aws:lambda:il-central-1:459530977127:layer:LambdaInsightsExtension:5
Moyen-Orient (Bahreïn)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:28
Moyen-Orient (EAU)	arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:11
Amérique du Sud (São Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:37

1.0.178.0

La version 1.0.178.0 ajoute le support pour les régions suivantes AWS .

- Asie-Pacifique (Hyderabad)
- Asie-Pacifique (Jakarta)
- Europe (Espagne)
- Europe (Zurich)
- Moyen-Orient (EAU)

ARN pour la version 1.0.178.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:35`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:33`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:33`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:33`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:25`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:25`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension:8`

Région	ARN
Asie-Pacifique (Jakarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension:11
Asie-Pacifique (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:31
Asie-Pacifique (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:32
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:33
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:33
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:50
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:32
Chine (Beijing)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:26
Chine (Ningxia) ;	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:26
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:35
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:33
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:33

Région	ARN
Europe (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:25`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:32`
Europe (Espagne)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension:10`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:30`
Europe (Zurich)	`arn:aws:lambda:eu-central-2:033019950311:layer:LambdaInsightsExtension:7`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:25`
Moyen-Orient (EAU)	`arn:aws:lambda:me-central-1:732604637566:layer:LambdaInsightsExtension:9`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:32`

1.0.143.0

La version 1.0.143.0 inclut des corrections de bogues compatibles avec Python 3.7 et Go 1.x. Le moteur d'exécution Lambda Python 3.6 est obsolète. Pour plus d'informations, veuillez consulter la rubrique Environnement d'exécution Lambda.

ARN pour la version 1.0.143.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:21`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:21`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:20`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:21`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:13`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:13`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:21`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:2`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:20`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:21`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:21`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:32`

Région	ARN
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:20`
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:14`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:14`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:21`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:21`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:21`
Europe (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:13`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:20`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:20`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:13`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:20`

1.0.135.0

La version 1.0.135.0 inclut des corrections de bogues concernant la façon dont Lambda Insights collecte et signale l'utilisation du disque et du descripteur de fichier. Dans les versions précédentes

de l'extension, la métrique `tmp_free` indiquait l'espace libre maximal dans le répertoire `/tmp` pendant l'exécution d'une fonction. Cette version modifie la métrique pour signaler la valeur minimale à la place, ce qui la rend plus utile lors de l'évaluation de l'utilisation du disque. Pour plus d'informations sur les quotas de stockage du répertoire `tmp`, consultez Quotas Lambda.

La version 1.0.135.0 signale également l'utilisation du descripteur de fichier (`fd_use` et `fd_max`) en tant que valeur maximale entre les processus plutôt que de générer des rapports au niveau du système d'exploitation.

ARN pour la version 1.0.135.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:18`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:18`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:18`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:18`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:11`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:11`
Asia Pacific (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:18`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension:1`

Région	ARN
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:18`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:18`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:18`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:25`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:18`
Chine (Beijing)	`arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:11`
Chine (Ningxia) ;	`arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:11`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:18`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:18`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:18`
Europe (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:11`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:18`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:18`

Région	ARN
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:11`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:18`

1,0119,0

ARN pour la version 1.0.119.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:16`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:16`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:16`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:16`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:9`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:9`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:16`

Région	ARN
Asie-Pacifique (Séoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:16
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:16
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:16
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:23
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:16
Chine (Beijing)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:9
Chine (Ningxia) ;	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:9
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:16
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:16
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:16
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:9
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:16
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:16

Région	ARN
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:9`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:16`

1,098.0

Cette version supprime la journalisation inutile et résout également un problème lié aux appels locaux de la AWS Serverless Application Model CLI. Pour plus d'informations sur ce problème, voir Ajouter des LambdaInsightsExtension résultats dans le délai d'expiration avec « sam local invoke ».

ARN pour la version 1.0.98.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:14`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:14`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:14`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension:8`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension:8`

Région	ARN
Asie-Pacifique (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:14
Asie-Pacifique (Séoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:14
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:14
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:14
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:14
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:14
Chine (Beijing)	arn:aws-cn:lambda:cn-north-1:488211338238:layer:LambdaInsightsExtension:8
Chine (Ningxia) ;	arn:aws-cn:lambda:cn-northwest-1:488211338238:layer:LambdaInsightsExtension:8
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:14
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:14
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:14
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension:8
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:14

Région	ARN
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:14`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension:8`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:14`

1,089.0

Cette version corrige l'horodatage de l'événement de performance pour toujours représenter le début de l'invocation de la fonction.

ARN pour la version 1.0.89.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:12`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:12`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:12`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:12`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:12`

Région	ARN
Asie-Pacifique (Séoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:12`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:12`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:12`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:12`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:12`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:12`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:12`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:12`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:12`

1.0.86.0

Avec la version 1.0.54.0 de l'extension, il arrivait que les métriques de mémoire soient indiquées de manière incorrecte et soient supérieures à 100 %. La version 1.0.86.0 corrige le problème

de mesure de la mémoire en utilisant les mêmes données d'événement que les métriques de la plateforme Lambda. Cela signifie que vous pouvez voir un changement spectaculaire dans les valeurs enregistrées de métrique de la mémoire. Ceci est réalisé en utilisant la nouvelle API Logs Lambda. Elle fournit une mesure plus précise de l'utilisation de la mémoire de l'environnement de test Lambda. Cependant, il faut savoir que l'API Logs Lambda ne peut pas fournir des événements de rapport de plateforme si un environnement de test de fonction expire et est ralenti. Dans ce cas, Lambda Insights ne peut pas enregistrer les métriques d'invocation. Pour plus d'informations sur l'API Logs Lambda, consultez API Logs AWS Lambda.

Nouvelles fonctions de la version 1.0.86.0

- Utilise l'API Logs Lambda pour corriger la métrique de mémoire. Cela résout le problème précédent où les statistiques de la mémoire étaient supérieures à 100 %.
- Introduit `Init Duration` en tant que nouvelle CloudWatch métrique.
- Utilise l'ARN d'appel pour ajouter une dimension de version pour les alias et les versions invoquées. Si vous utilisez des alias ou des versions Lambda pour réaliser des déploiements incrémentiels (tels que des déploiements bleu/vert), vous pouvez afficher vos métriques selon l'alias appelé. La dimension de version n'est pas appliquée si la fonction n'utilise pas d'alias ou de version. Pour plus d'informations, consultez les alias de fonction Lambda.
- Ajoute un `billed_mb_ms field` aux événements de performance pour afficher le coût par appel. Cela ne prend pas en compte les coûts associés à la simultanéité provisionnée.
- Ajoute des champs `billed_duration` et `duration` aux événements de performance.

ARN pour la version 1.0.86.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:11`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:11`

Région	ARN
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:11`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:11`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:11`
Asie-Pacifique (Séoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:11`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:11`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:11`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:11`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:11`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:11`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:11`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:11`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:11`

Région	ARN
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:11`

1.0.54.0

La version 1.0.54.0 est la première version de l'extension Lambda Insights.

ARN pour la version 1.0.54.0

Le tableau suivant répertorie les ARN à utiliser pour cette version de l'extension dans chaque AWS région où elle est disponible.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension:2`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension:2`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:2`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension:2`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension:2`
Asie-Pacifique (Séoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension:2`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension:2`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension:2`

Région	ARN
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension:2`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension:2`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension:2`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension:2`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension:2`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension:2`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension:2`
Amérique du Sud (São Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension:2`

Plateformes ARM64

Cette section répertorie les versions de l'extension Lambda Insights pour les plateformes ARM64, ainsi que les ARN à utiliser pour ces extensions dans chaque région. AWS

> **Important**
>
> Les extensions Lambda Insights 1.0.317.0 et versions ultérieures ne sont pas compatibles avec Amazon Linux 1.

1,0317,0

La version 1.0.317.0 inclut la suppression du support pour la plate-forme Amazon Linux 1 et des corrections de bugs. Il inclut également le soutien aux AWS GovCloud (US) régions.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:19`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:21`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:17`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:17`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:17`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:5`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:17`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:21`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:16`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`

Région	ARN
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:19
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:30
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:17
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:19
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:17
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:17
Europe (Espagne)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:5
Europe (Stockholm)	arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:17
Moyen-Orient (Bahreïn)	arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:17
Amérique du Sud (Sao Paulo)	arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:17

Région	ARN
AWS GovCloud (USA Est)	`arn:aws-us-gov:lambda:us-gov-east-1:122132214140:layer:LambdaInsightsExtension-Arm64:1`
AWS GovCloud (US-Ouest)	`arn:aws-us-gov:lambda:us-gov-west-1:751350123760:layer:LambdaInsightsExtension-Arm64:1`

1.0.295.0

La version 1.0.295.0 inclut des mises à jour des dépendances pour tous les environnements d'exécution compatibles.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:18`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:20`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:16`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:16`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:4`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:16`

Région	ARN
Asie-Pacifique (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:20
Asie-Pacifique (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:15
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:17
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:29
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:18
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:16
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europe (Espagne)	arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:4

Région	ARN
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:16`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`

1.0.275.0

La version 1.0.275.0 inclut des corrections de bogues pour tous les environnements d'exécution compatibles et un support pour les régions Europe (Espagne) et Asie-Pacifique (Hyderabad).

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:16`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:18`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:14`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:14`
Asie-Pacifique (Hyderabad)	`arn:aws:lambda:ap-south-2:891564319516:layer:LambdaInsightsExtension-Arm64:2`

Région	ARN
Asie-Pacifique (Jakarta)	arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:14
Asie-Pacifique (Mumbai)	arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:18
Asie-Pacifique (Osaka)	arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:13
Asia Pacific (Seoul)	arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:15
Asie-Pacifique (Singapour)	arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asie-Pacifique (Sydney)	arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Asie-Pacifique (Tokyo)	arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:27
Canada (Centre)	arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:14
Europe (Francfort)	arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europe (Irlande)	arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europe (Londres)	arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:16
Europe (Milan)	arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:14
Europe (Paris)	arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:14

Région	ARN
Europe (Espagne)	`arn:aws:lambda:eu-south-2:352183217350:layer:LambdaInsightsExtension-Arm64:2`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:14`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`

1.0.273.0

La version 1.0.273.0 inclut des corrections de bugs pour tous les environnements d'exécution compatibles.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:14`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:9`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:9`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:9`

Région	ARN
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:9`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:14`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:9`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:11`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:23`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:12`
Europe (Milan)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:9`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:10`

Région	ARN
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:9`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:10`

1,0,229,0

La version 1.0.229.0 inclut des correctifs de bogues pour toutes les exécutions compatibles. En outre, elle prend désormais en charge les régions suivantes :

- USA Ouest (Californie du Nord)
- Afrique (Le Cap)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Osaka)
- Asie-Pacifique (Séoul)
- Canada (Centre)
- Europe (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Moyen-Orient (Bahreïn)
- Amérique du Sud (Sao Paulo)

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`

Région	ARN
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:7`
USA Ouest (Californie du Nord)	`arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Afrique (Le Cap)	`arn:aws:lambda:af-south-1:012438385374:layer:LambdaInsightsExtension-Arm64:2`
Asie-Pacifique (Hong Kong)	`arn:aws:lambda:ap-east-1:519774774795:layer:LambdaInsightsExtension-Arm64:2`
Asie-Pacifique (Jakarta)	`arn:aws:lambda:ap-southeast-3:439286490199:layer:LambdaInsightsExtension-Arm64:2`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:7`
Asie-Pacifique (Osaka)	`arn:aws:lambda:ap-northeast-3:194566237122:layer:LambdaInsightsExtension-Arm64:2`
Asia Pacific (Seoul)	`arn:aws:lambda:ap-northeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:4`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:11`
Canada (Centre)	`arn:aws:lambda:ca-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

Région	ARN
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:5`
Europe (Espagne)	`arn:aws:lambda:eu-south-1:339249233099:layer:LambdaInsightsExtension-Arm64:2`
Europe (Paris)	`arn:aws:lambda:eu-west-3:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`
Moyen-Orient (Bahreïn)	`arn:aws:lambda:me-south-1:285320876703:layer:LambdaInsightsExtension-Arm64:2`
Amérique du Sud (Sao Paulo)	`arn:aws:lambda:sa-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:3`

1.0.135.0

La version 1.0.135.0 inclut des corrections de bogues concernant la façon dont Lambda Insights collecte et signale l'utilisation du disque et du descripteur de fichier. Dans les versions précédentes de l'extension, la métrique `tmp_free` indiquait l'espace libre maximal dans le répertoire `/tmp` pendant l'exécution d'une fonction. Cette version modifie la métrique pour signaler la valeur minimale à la place, ce qui la rend plus utile lors de l'évaluation de l'utilisation du disque. Pour plus d'informations sur les quotas de stockage du répertoire `tmp`, consultez Quotas Lambda.

La version 1.0.135.0 signale également l'utilisation du descripteur de fichier (`fd_use` et `fd_max`) en tant que valeur maximale entre les processus plutôt que de générer des rapports au niveau du système d'exploitation.

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:2`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:2`

1,0119,0

Région	ARN
USA Est (Virginie du Nord)	`arn:aws:lambda:us-east-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Région	ARN
USA Est (Ohio)	`arn:aws:lambda:us-east-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
USA Ouest (Oregon)	`arn:aws:lambda:us-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asie-Pacifique (Mumbai)	`arn:aws:lambda:ap-south-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asie-Pacifique (Singapour)	`arn:aws:lambda:ap-southeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asie-Pacifique (Sydney)	`arn:aws:lambda:ap-southeast-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Asie-Pacifique (Tokyo)	`arn:aws:lambda:ap-northeast-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europe (Francfort)	`arn:aws:lambda:eu-central-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europe (Irlande)	`arn:aws:lambda:eu-west-1:580247275435:layer:LambdaInsightsExtension-Arm64:1`
Europe (Londres)	`arn:aws:lambda:eu-west-2:580247275435:layer:LambdaInsightsExtension-Arm64:1`

Utilisation de la console pour activer Lambda Insights sur une fonction Lambda existante

Effectuez les étapes suivantes dans la console Lambda afin d'activer Lambda Insights sur une fonction Lambda existante.

Pour activer Lambda Insights sur une fonction Lambda

1. Ouvrez la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/.
2. Choisissez le nom d'une fonction, puis sélectionnez l'onglet Configuration sur l'écran suivant.

3. Sous l'onglet Configuration, choisissez Outils de surveillance et d'exploitation dans le menu de navigation de gauche, puis sélectionnez Modifier.

   Vous êtes redirigé vers un écran où vous pouvez modifier les outils de surveillance.
4. Dans le cadre de la surveillance améliorée de Lambda Insights, choisissez Modifier.
5. Sous CloudWatch Lambda Insights, activez la surveillance améliorée, puis choisissez Enregistrer.

Utilisation du AWS CLI pour activer Lambda Insights sur une fonction Lambda existante

Suivez ces étapes pour utiliser Lambda Insights AWS CLI pour activer Lambda Insights sur une fonction Lambda existante.

Étape 1 : mise à jour des autorisations de fonction

Pour mettre à jour des autorisations de fonction

- Associez la politique IAM CloudWatchLambdaInsightsExecutionRolePolicygérée au rôle d'exécution de la fonction en saisissant la commande suivante.

```
aws iam attach-role-policy \
--role-name function-execution-role \
--policy-arn "arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy"
```

Étape 2 : installation de l'extension Lambda

Installez l'extension Lambda en saisissant la commande suivante. Remplacez la valeur ARN pour le paramètre `layers` avec l'ARN qui correspond à votre région et à la version d'extension que vous souhaitez utiliser. Pour plus d'informations, consultez Versions disponibles de l'extension Lambda Insights.

```
aws lambda update-function-configuration \
   --function-name function-name \
   --layers "arn:aws:lambda:us-west-1:580247275435:layer:LambdaInsightsExtension:14"
```

Étape 3 : activer le point de CloudWatch terminaison VPC Logs

Cette étape n'est nécessaire que pour les fonctions exécutées dans un sous-réseau privé sans accès à Internet, et si vous n'avez pas encore configuré de point de terminaison VPC (Virtual Private Cloud) CloudWatch Logs.

Si vous devez effectuer cette étape, saisissez la commande suivante en remplaçant les espaces réservés par des informations pour votre VPC.

Pour plus d'informations, consultez la section Utilisation CloudWatch des journaux avec les points de terminaison VPC de l'interface.

```
aws ec2 create-vpc-endpoint \
--vpc-id vpcId \
--vpc-endpoint-type Interface \
--service-name com.amazonaws.region.logs \
--subnet-id subnetId
--security-group-id securitygroupId
```

Utilisation de la AWS SAM CLI pour activer Lambda Insights sur une fonction Lambda existante

Suivez ces étapes pour utiliser Lambda Insights AWS SAM AWS CLI pour activer Lambda Insights sur une fonction Lambda existante.

Si la dernière version de la AWS SAM CLI n'est pas encore installée, vous devez d'abord l'installer ou la mettre à niveau. Pour plus d'informations, consultez la section Installation de la AWS SAM CLI.

Étape 1 : installation de la couche

Pour rendre l'extension Lambda Insights disponible pour toutes vos fonctions Lambda, ajoutez une propriété `Layers` à la section `Globals` de votre modèle SAM avec l'ARN de la couche Lambda Insights. L'exemple ci-dessous utilise la couche pour la version initiale de Lambda Insights. Pour obtenir la dernière version de la couche d'extension Lambda Insights, consultez Versions disponibles de l'extension Lambda Insights.

```
Globals:
  Function:
    Layers:
      - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Pour activer cette couche pour une seule fonction, ajoutez la propriété `Layers` à la fonction, comme illustré dans cet exemple.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Étape 2 : ajout de la politique gérée

Pour chaque fonction, ajoutez la politique CloudWatchLambdaInsightsExecutionRolePolicyIAM.

AWS SAM ne prend pas en charge les politiques globales, vous devez donc les activer individuellement pour chaque fonction, comme indiqué dans cet exemple. Pour plus d'informations sur les variables globales, consultez Section Globals.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
    Properties:
      Policies:
        - CloudWatchLambdaInsightsExecutionRolePolicy
```

Appel local

La AWS SAM CLI prend en charge les extensions Lambda. Cependant, chaque invocation exécutée localement réinitialise l'environnement d'exécution. Les données Lambda Insights ne seront pas disponibles à partir d'invocations locales, car le runtime est redémarré sans événement d'arrêt. Pour plus d'informations, voir Version 1.6.0 - Ajout de la prise en charge des tests locaux des AWS Lambda extensions.

Dépannage

Pour dépanner votre installation Lambda Insights, ajoutez la variable d'environnement suivante à votre fonction Lambda afin d'activer la journalisation du débogage.

```
Resources:
  MyFunction:
    Type: AWS::Serverless::Function
```

```
    Properties:
      Environment:
        Variables:
          LAMBDA_INSIGHTS_LOG_LEVEL: info
```

Utilisation AWS CloudFormation pour activer Lambda Insights sur une fonction Lambda existante

Suivez ces étapes AWS CloudFormation pour activer Lambda Insights sur une fonction Lambda existante.

Étape 1 : installation de la couche

Ajoutez la couche Lambda Insights à la propriété `Layers` dans l'ARN de couche Lambda Insights. L'exemple ci-dessous utilise la couche pour la version initiale de Lambda Insights. Pour obtenir la dernière version de la couche d'extension Lambda Insights, consultez Versions disponibles de l'extension Lambda Insights.

```
Resources:
  MyFunction:
    Type: AWS::Lambda::Function
    Properties:
      Layers:
        - !Sub "arn:aws:lambda:
${AWS::Region}:580247275435:layer:LambdaInsightsExtension:14"
```

Étape 2 : ajout de la politique gérée

Ajoutez la politique CloudWatchLambdaInsightsExecutionRolePolicyIAM à votre rôle d'exécution de fonction.

```
Resources:
  MyFunctionExecutionRole:
    Type: 'AWS::IAM::Role'
    Properties:
      ManagedPolicyArns:
        - 'arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy'
```

Étape 3 : (facultatif) ajout d'un point de terminaison d'un VPC

Cette étape n'est nécessaire que pour les fonctions exécutées dans un sous-réseau privé sans accès à Internet, et si vous n'avez pas encore configuré de point de terminaison VPC (Virtual Private Cloud)

CloudWatch Logs. Pour plus d'informations, consultez la section Utilisation CloudWatch des journaux avec les points de terminaison VPC de l'interface.

```
Resources:
  CloudWatchLogsVpcPrivateEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      PrivateDnsEnabled: 'true'
      VpcEndpointType: Interface
      VpcId: !Ref: VPC
      ServiceName: !Sub com.amazonaws.${AWS::Region}.logs
      SecurityGroupIds:
        - !Ref InterfaceVpcEndpointSecurityGroup
      SubnetIds:
        - !Ref PublicSubnet01
        - !Ref PublicSubnet02
        - !Ref PublicSubnet03
```

Utilisation du AWS CDK pour activer Lambda Insights sur une fonction Lambda existante

Suivez ces étapes pour utiliser Lambda Insights AWS CDK pour activer Lambda Insights sur une fonction Lambda existante. Pour suivre ces étapes, vous devez déjà utiliser le AWS CDK pour gérer vos ressources.

Les commandes de cette section sont disponibles TypeScript.

Tout d'abord, mettez à jour les autorisations de fonction.

```
executionRole.addManagedPolicy(
 ManagedPolicy.fromAwsManagedPolicyName('CloudWatchLambdaInsightsExecutionRolePolicy')
);
```

Ensuite, installez l'extension sur la fonction Lambda. Remplacez la valeur ARN pour le paramètre `layerArn` avec l'ARN qui correspond à votre région et à la version d'extension que vous souhaitez utiliser. Pour plus d'informations, consultez Versions disponibles de l'extension Lambda Insights.

```
import lambda = require('@aws-cdk/aws-lambda');
const layerArn = 'arn:aws:lambda:us-
west-1:580247275435:layer:LambdaInsightsExtension:14';
const layer = lambda.LayerVersion.fromLayerVersionArn(this, 'LayerFromArn', layerArn);
```

Si nécessaire, activez le point de terminaison du cloud privé virtuel (VPC) pour les CloudWatch journaux. Cette étape n'est nécessaire que pour les fonctions exécutées dans un sous-réseau privé sans accès à Internet, et si vous n'avez pas encore configuré de point de terminaison CloudWatch Logs VPC.

```
const cloudWatchLogsEndpoint = vpc.addInterfaceEndpoint('cwl-gateway', {
  service: InterfaceVpcEndpointAwsService.CLOUDWATCH_LOGS,
});

cloudWatchLogsEndpoint.connections.allowDefaultPortFromAnyIpv4();
```

Utilisation de l'infrastructure sans serveur pour activer Lambda Insights sur une fonction Lambda existante

Suivez ces étapes pour utiliser l'infrastructure sans serveur et activer Lambda Insights sur une fonction Lambda existante. Pour plus d'informations sur l'infrastructure sans serveur, consultez serverless.com.

Pour ce faire, un plugin Lambda Insights pour infrastructure sans serveur est nécessaire. Pour plus d'informations, consultez serverless-plugin-lambda-insights.

Si la dernière version de l'interface de ligne de commande sans serveur n'est pas déjà installée, vous devez d'abord l'installer ou la mettre à niveau. Pour plus d'informations, voir Commencer avec Serverless Framework Open Source & AWS.

Pour utiliser l'infrastructure sans serveur et activer Lambda Insights sur une fonction Lambda

1. Installez le plugin sans serveur pour Lambda Insights en exécutant la commande suivante dans votre répertoire sans serveur :

   ```
   npm install --save-dev serverless-plugin-lambda-insights
   ```

2. Dans votre fichier `serverless.yml`, ajoutez le plugin dans la section `plugins` comme indiqué :

   ```
   provider:
     name: aws
   plugins:
     - serverless-plugin-lambda-insights
   ```

3. Activez Lambda Insights.
   - Vous pouvez activer Lambda Insights individuellement par fonction en ajoutant la propriété suivante au fichier serverless.yml.

     ```
     functions:
       myLambdaFunction:
         handler: src/app/index.handler
         lambdaInsights: true #enables Lambda Insights for this function
     ```

   - Vous pouvez activer Lambda Insights pour toutes les fonctions du fichier `serverless.yml` en ajoutant la section personnalisée suivante :

     ```
     custom:
       lambdaInsights:
         defaultLambdaInsights: true #enables Lambda Insights for all functions
     ```

4. Redéployez le service sans serveur en saisissant la commande suivante :

   ```
   serverless deploy
   ```

   Toutes les fonctions sont redéployées et Lambda Insights est activé pour les fonctions que vous avez spécifiées. Cette commande active Lambda Insights en ajoutant la couche Lambda Insights et en attachant les autorisations nécessaires à l'aide de la stratégie IAM `arn:aws:iam::aws:policy/CloudWatchLambdaInsightsExecutionRolePolicy`.

Activation de Lambda Insights sur un déploiement de l'image de conteneur Lambda

Pour activer Lambda Insights sur une fonction Lambda déployée en tant qu'image de conteneur, ajoutez des lignes dans votre Dockerfile. Ces lignes installent l'agent Lambda Insights comme une extension dans votre image de conteneur. Les lignes à ajouter sont différentes pour les conteneurs x86-64 et les conteneurs ARM64.

> **Note**
>
> L'agent Lambda Insights est pris en charge uniquement sur les exécutions Lambda utilisant Amazon Linux 2.

Rubriques

- Déploiement d'images de conteneur x86-64
- Déploiement d'images de conteneur ARM64

Déploiement d'images de conteneur x86-64

Pour activer Lambda Insights sur une fonction Lambda déployée en tant qu'image de conteneur exécutée sur un conteneur x86-64, ajoutez les lignes suivantes dans votre Dockerfile. Ces lignes installent l'agent Lambda Insights comme une extension dans votre image de conteneur.

```
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm
```

Après avoir créé votre fonction Lambda, attribuez la politique CloudWatchLambdaInsightsExecutionRolePolicyIAM au rôle d'exécution de la fonction, et Lambda Insights est activé sur la fonction Lambda basée sur l'image du conteneur.

> **Note**
>
> Pour utiliser une ancienne version de l'extension Lambda Insights, remplacez l'URL des commandes précédentes par cette URL : `https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension.1.0.111.0.rpm`. Seules les versions 1.0.111.0 et ultérieures de Lambda Insights sont actuellement disponibles. Pour plus d'informations, consultez Versions disponibles de l'extension Lambda Insights.

Pour vérifier la signature du package d'agent Lambda Insights sur un serveur Linux

1. Saisissez la commande suivante pour télécharger la clé publique.

   ```
   shell$ wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
   lambda-insights-extension.gpg
   ```

2. Saisissez la commande suivante pour importer la clé publique dans votre porte-clés.

   ```
   shell$ gpg --import lambda-insights-extension.gpg
   ```

La sortie est similaire à ce qui suit. Notez la valeur `key`, car vous en aurez besoin lors de l'étape suivante. Dans cet exemple de sortie, la valeur clé est `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Vérifiez l'empreinte digitale en saisissant la commande suivante. Remplacez `key-value` avec la valeur de la clé de l'étape précédente.

```
shell$  gpg --fingerprint key-value
```

La chaîne d'empreinte digitale dans la sortie de cette commande doit être équivalente à `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Si la chaîne ne correspond pas, n'installez pas l'agent et contactez AWS.

4. Après avoir vérifié l'empreinte digitale, vous pouvez l'utiliser pour vérifier le package d'agent Lambda Insights. Téléchargez le fichier signature de package en saisissant la commande suivante.

```
shell$  wget https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm.sig
```

5. Vérifiez la signature en saisissant la commande suivante.

```
shell$ gpg --verify lambda-insights-extension.rpm.sig lambda-insights-extension.rpm
```

Le résultat doit ressembler à ce qui suit :

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

La sortie attendue peut être composée d'un avertissement concernant une signature approuvée. Une clé est uniquement approuvée si vous ou une personne de confiance l'a signée. Cela ne signifie pas que la signature n'est pas valide, mais seulement que vous n'avez pas vérifié la clé publique.

Si la sortie contient `BAD signature`, vérifiez si vous avez effectué les étapes correctement. Si vous recevez toujours une `BAD signature` réponse, contactez le fichier téléchargé AWS et évitez de l'utiliser.

Exemple x86-64

Cette section inclut un exemple d'activation de Lambda Insights sur une fonction Python Lambda basée sur une image de conteneur.

Exemple d'activation de Lambda Insights sur une image de conteneur Lambda

1. Créez un fichier Dockerfile semblable à ce qui suit :

```
FROM public.ecr.aws/lambda/python:3.8

// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension.rpm && \
    rpm -U lambda-insights-extension.rpm && \
    rm -f lambda-insights-extension.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Créez un fichier Python nommé `index.py`, qui est semblable à ce qui suit :

```
def handler(event, context):
  return {
    'message': 'Hello World!'
  }
```

3. Placez le fichier Dockerfile et `index.py` dans le même répertoire. Ensuite, exécutez les étapes suivantes dans ce répertoire pour créer l'image Docker et la télécharger sur Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
```

```
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Utilisez cette image Amazon ECR que vous venez de créer pour créer la fonction Lambda.
5. Assignez la politique CloudWatchLambdaInsightsExecutionRolePolicyIAM au rôle d'exécution de la fonction.

Déploiement d'images de conteneur ARM64

Pour activer Lambda Insights sur une fonction Lambda déployée en tant qu'image de conteneur exécutée sur un conteneur AL2_aarch64 (qui utilise une architecture ARM64), ajoutez les lignes suivantes dans votre Dockerfile. Ces lignes installent l'agent Lambda Insights comme une extension dans votre image de conteneur.

```
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/
amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm
```

Après avoir créé votre fonction Lambda, attribuez la politique CloudWatchLambdaInsightsExecutionRolePolicyIAM au rôle d'exécution de la fonction, et Lambda Insights est activé sur la fonction Lambda basée sur l'image du conteneur.

> **Note**
>
> Pour utiliser une ancienne version de l'extension Lambda Insights, remplacez l'URL des commandes précédentes par cette URL : `https://lambda-insights-extension-arm64.s3-ap-northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.1.0.229.0.rpm`. Seules les versions 1.0.229.0 et ultérieures de Lambda Insights sont actuellement disponibles. Pour plus d'informations, consultez Versions disponibles de l'extension Lambda Insights.

Pour vérifier la signature du package d'agent Lambda Insights sur un serveur Linux

1. Saisissez la commande suivante pour télécharger la clé publique.

```
shell$ wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/lambda-insights-extension.gpg
```

2. Saisissez la commande suivante pour importer la clé publique dans votre porte-clés.

```
shell$ gpg --import lambda-insights-extension.gpg
```

   La sortie est similaire à ce qui suit. Notez la valeur `key`, car vous en aurez besoin lors de l'étape suivante. Dans cet exemple de sortie, la valeur clé est `848ABDC8`.

```
gpg: key 848ABDC8: public key "Amazon Lambda Insights Extension" imported
gpg: Total number processed: 1
gpg: imported: 1  (RSA: 1)
```

3. Vérifiez l'empreinte digitale en saisissant la commande suivante. Remplacez `key-value` avec la valeur de la clé de l'étape précédente.

```
shell$  gpg --fingerprint key-value
```

   La chaîne d'empreinte digitale dans la sortie de cette commande doit être équivalente à `E0AF FA11 FFF3 5BD7 349E E222 479C 97A1 848A BDC8`. Si la chaîne ne correspond pas, n'installez pas l'agent et contactez AWS.

4. Après avoir vérifié l'empreinte digitale, vous pouvez l'utiliser pour vérifier le package d'agent Lambda Insights. Téléchargez le fichier signature de package en saisissant la commande suivante.

```
shell$  wget https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm.sig
```

5. Vérifiez la signature en saisissant la commande suivante.

```
shell$ gpg --verify lambda-insights-extension-arm64.rpm.sig lambda-insights-
extension-arm64.rpm
```

   Le résultat doit ressembler à ce qui suit :

```
gpg: Signature made Thu 08 Apr 2021 06:41:00 PM UTC using RSA key ID 848ABDC8
gpg: Good signature from "Amazon Lambda Insights Extension"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: E0AF FA11 FFF3 5BD7 349E  E222 479C 97A1 848A BDC8
```

La sortie attendue peut être composée d'un avertissement concernant une signature approuvée. Une clé est uniquement approuvée si vous ou une personne de confiance l'a signée. Cela ne signifie pas que la signature n'est pas valide, mais seulement que vous n'avez pas vérifié la clé publique.

Si la sortie contient `BAD signature`, vérifiez si vous avez effectué les étapes correctement. Si vous recevez toujours une `BAD signature` réponse, contactez le fichier téléchargé AWS et évitez de l'utiliser.

Exemple ARM64

Cette section inclut un exemple d'activation de Lambda Insights sur une fonction Python Lambda basée sur une image de conteneur.

Exemple d'activation de Lambda Insights sur une image de conteneur Lambda

1. Créez un fichier Dockerfile semblable à ce qui suit :

```
FROM public.ecr.aws/lambda/python:3.8
// extra lines to install the agent here
RUN curl -O https://lambda-insights-extension-arm64.s3-ap-
northeast-1.amazonaws.com/amazon_linux/lambda-insights-extension-arm64.rpm && \
    rpm -U lambda-insights-extension-arm64.rpm && \
    rm -f lambda-insights-extension-arm64.rpm

COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Créez un fichier Python nommé `index.py`, qui est semblable à ce qui suit :

```
def handler(event, context):
  return {
    'message': 'Hello World!'
```

```
  }
```

3. Placez le fichier Dockerfile et `index.py` dans le même répertoire. Ensuite, exécutez les étapes suivantes dans ce répertoire pour créer l'image Docker et la télécharger sur Amazon ECR.

```
// create an ECR repository
aws ecr create-repository --repository-name test-repository
// build the docker image
docker build -t test-image .
// sign in to AWS
aws ecr get-login-password | docker login --username AWS --password-stdin
 "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com
// tag the image
docker tag test-image:latest "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/
test-repository:latest
// push the image to ECR
docker push "${ACCOUNT_ID}".dkr.ecr."${REGION}".amazonaws.com/test-
repository:latest
```

4. Utilisez cette image Amazon ECR que vous venez de créer pour créer la fonction Lambda.
5. Assignez la politique CloudWatchLambdaInsightsExecutionRolePolicyIAM au rôle d'exécution de la fonction.

Affichage de vos métriques Lambda Insights

Après avoir installé l'extension Lambda Insights sur une fonction Lambda qui a été invoquée, vous pouvez utiliser la CloudWatch console pour consulter vos métriques. Vous pouvez bénéficier d'une vue d'ensemble multifonctions, ou vous concentrer sur une seule fonction.

Pour obtenir la liste des métriques Lambda Insights, consultez Métriques collectées par Lambda Insights.

Pour consulter la vue d'ensemble multifonctions de vos métriques Lambda Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, sous Lambda Insights, choisissez Multi-fonction (Multifonctions).

La partie supérieure de la page affiche des graphiques contenant des métriques agrégées de toutes vos fonctions Lambda dans la région pour lesquelles Lambda Insights est activé. Un tableau répertoriant les fonctions est repris plus bas sur la page.

3. Pour filtrer par nom de fonction et réduire le nombre de fonctions affichées, saisissez une partie du nom de la fonction dans la zone de texte située en haut de la page.
4. Pour ajouter cette vue à un tableau de bord en tant que widget, choisissez Add to dashboard (Ajouter au tableau de bord).

Pour afficher les métriques d'une fonction unique

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation de gauche, sous Lambda Insights, choisissez Single-function (Fonction unique).

   La partie supérieure de la page affiche des graphiques contenant des métriques pour la fonction sélectionnée.
3. Si vous avez activé la fonction X-Ray, vous pouvez sélectionner un seul ID de suivi. Cette action ouvre la page de la carte de suivi X-Ray pour cette invocation. De là, vous pouvez faire un zoom arrière pour voir le suivi distribué et les autres services impliqués dans le traitement de cette transaction spécifique. Pour plus d'informations sur la carte de suivi X-Ray, veuillez consulter la rubrique Using the X-Ray Trace Map.
4. Pour ouvrir CloudWatch Logs Insights et zoomer sur une erreur spécifique, choisissez Afficher les journaux dans le tableau au bas de la page.
5. Pour ajouter cette vue à un tableau de bord en tant que widget, choisissez Add to dashboard (Ajouter au tableau de bord).

Intégration à Application Insights

Amazon CloudWatch Application Insights vous aide à surveiller vos applications, à identifier et à configurer les indicateurs clés, les journaux et les alarmes pour l'ensemble de vos ressources applicatives et de votre infrastructure technologique. Pour plus d'informations, consultez Informations sur les CloudWatch applications Amazon.

Vous pouvez activer Application Insights pour recueillir des données supplémentaires à partir de vos fonctions Lambda. Si vous ne l'avez pas déjà fait, vous pouvez l'activer en sélectionnant Auto-

configure Application Insights (Configuration automatique d'Application Insights) dans l'onglet Application Insights, sous la vue des performances dans le tableau de bord Lambda Insights.

Si vous avez déjà configuré CloudWatch Application Insights pour surveiller vos fonctions Lambda, le tableau de bord Application Insights apparaît sous le tableau de bord Lambda Insights, dans l'onglet Application Insights.

Métriques collectées par Lambda Insights

Lambda Insights collecte plusieurs métriques à partir des fonctions Lambda où il est installé. Certaines de ces mesures sont disponibles sous forme de données agrégées de séries chronologiques dans CloudWatch Metrics. Les autres mesures ne sont pas agrégées sous forme de séries chronologiques, mais peuvent être trouvées dans les entrées de journal au format métrique intégré à l'aide de CloudWatch Logs Insights.

Les métriques suivantes sont disponibles sous forme de données agrégées de séries chronologiques dans CloudWatch Metrics in the `LambdaInsights` namespace.

Nom de la métrique	Dimensions	Description
`cpu_total_time`	function_name function_name, version	Somme de `cpu_system_time` et `cpu_user_time` . Unité : millisecondes
`init_duration`	function_name function_name, version	Temps consacré à la phase `init` du cycle de vie de l'environ nement d'exécution Lambda. Unité : millisecondes
`memory_ut ilization`	function_name function_name, version	Mémoire maximale mesurée en pourcentage de la mémoire allouée à la fonction.

Nom de la métrique	Dimensions	Description
		Unité : pourcentage
`rx_bytes`	function_name function_name, version	Nombre d'octets reçus par la fonction. Unité : octets
`tmp_used`		Quantité d'espace utilisée dans le répertoire `/tmp`. Unité : octets
`tx_bytes`	function_name function_name, version	Nombre d'octets envoyés par la fonction. Unité : octets
`total_memory`	function_name function_name, version	Quantité de mémoire allouée à votre fonction Lambda. Elle équivaut à la taille de la mémoire de votre fonction. Unité : mégaoctets

Nom de la métrique	Dimensions	Description
`total_network`	function_name function_name, version	Somme de `rx_bytes` et `tx_bytes`. Même pour les fonctions qui n'effectuent pas de tâches d'I/O, cette valeur est généralement supérieure à zéro en raison des appels réseau effectués par le runtime Lambda. Unité : octets
`used_memory_max`	function_name function_name, version	Mémoire mesurée de l'environnement de test de la fonction. Unité : mégaoctets

Les métriques suivantes peuvent être trouvées dans les entrées de journal au format métrique intégré à l'aide de CloudWatch Logs Insights. Pour plus d'informations sur CloudWatch Logs Insights, consultez Analyser les données des CloudWatch journaux avec Logs Insights.

Pour plus d'informations sur le format de métrique intégrée, consultez Intégration de métriques dans les journaux.

Nom des métriques	Description	
`cpu_system_time`	Temps passé par le CPU à exécuter le code du noyau. Unité : millisecondes	
`cpu_total_time`	Somme de `cpu_system_time` et `cpu_user_time` .	

Nom des métriques	Description
	Unité : millisecondes
`cpu_user_time`	Temps passé par le CPU à exécuter le code utilisateur. Unité : millisecondes
`fd_max`	Nombre maximal de descripteurs de fichiers disponibles. Unité : nombre
`fd_use`	Nombre maximal de descripteurs de fichiers en cours d'utilisation. Unité : nombre
`memory_utilization`	Mémoire maximale mesurée en pourcentage de la mémoire allouée à la fonction. Unité : pourcentage
`rx_bytes`	Nombre d'octets reçus par la fonction. Unité : octets
`tx_bytes`	Nombre d'octets envoyés par la fonction. Unité : octets
`threads_max`	Nombre de threads en cours d'utilisation dans le processus de la fonction. En tant qu'auteur de fonction, vous ne contrôlez pas le nombre initial de threads créés par le runtime. Unité : nombre

Nom des métriques	Description
`tmp_max`	Quantité d'espace disponible dans le répertoire `/tmp`. Unité : octets
`total_memory`	Quantité de mémoire allouée à votre fonction Lambda. Elle équivaut à la taille de la mémoire de votre fonction. Unité : mégaoctets
`total_network`	Somme de `rx_bytes` et `tx_bytes`. Même pour les fonctions qui n'effectuent pas de tâches d'I/O, cette valeur est généralement supérieure à zéro en raison des appels réseau effectués par le runtime Lambda. Unité : octets
`used_memory_max`	Mémoire mesurée de l'environnement de test de la fonction. Unité : octets

Résolution des problèmes et problèmes connus

La première étape à privilégier pour résoudre les problèmes consiste à activer la journalisation de débogage sur l'extension Lambda Insights. Pour ce faire, définissez la variable d'environnement suivante sur votre fonction Lambda :`LAMBDA_INSIGHTS_LOG_LEVEL=info`. Pour plus d'informations, consultez Utilisation des variables d'environnement AWS Lambda.

L'extension émet des journaux dans le même groupe de journaux que votre fonction (`/aws/lambda/`*`function-name`*`)`. Passez en revue ces journaux pour déterminer si l'erreur peut être liée à un problème de configuration.

Je ne vois aucune métrique de Lambda Insights

Si vous ne voyez pas les métriques Lambda Insights que vous vous attendez à voir, vérifiez les possibilités suivantes :

- Il se peut que les métriques soient simplement retardées. Si la fonction n'a pas encore été invoquée ou si les données n'ont pas encore été vidées, vous ne verrez pas les métriques dedans. CloudWatch Pour plus d'informations, consultez Problèmes connus plus loin dans cette section.
- Vérifiez que la fonction Lambda dispose des autorisations appropriées : assurez-vous que la politique CloudWatchLambdaInsightsExecutionRolePolicyIAM est affectée au rôle d'exécution de la fonction.
- Vérifiez le runtime Lambda — Lambda Insights ne prend en charge que certains runtimes Lambda. Pour obtenir une liste des runtimes pris en charge, consultez Aperçu Lambda.

  Par exemple, pour utiliser Lambda Insights sur Java 8, vous devez utiliser le runtime `java8.al2`, et non le runtime `java8`.
- Vérifier l'accès au réseau : la fonction Lambda se trouve peut-être sur un sous-réseau privé VPC sans accès à Internet et aucun point de terminaison VPC n'est configuré pour les journaux. CloudWatch Pour résoudre ce problème, vous pouvez définir la variable d'environnement `LAMBDA_INSIGHTS_LOG_LEVEL=info`.

Problèmes connus

Le délai des données peut atteindre 20 minutes. Lorsqu'un gestionnaire de fonctions se termine, Lambda gèle l'environnement de test (sandbox), ce qui bloque également l'extension Lambda Insights. Lorsque la fonction est en cours d'exécution, nous utilisons une stratégie de traitement par lots adaptative basée sur la fonction TPS pour produire des données. Toutefois, si la fonction cesse d'être appelée pendant une période prolongée et qu'il y a toujours des données d'événement dans la mémoire tampon, ces données peuvent être retardées jusqu'à ce que Lambda arrête l'environnement de test inactif. Lorsque Lambda arrête l'environnement de test, nous vidons les données conservées dans le tampon.

Exemple d'événement de télémétrie

Chaque invocation d'une fonction Lambda sur laquelle Lambda Insights est activé écrit un événement de journal unique dans le groupe de journaux `/aws/lambda-insights`. Chaque événement de

journal contient des métriques au format de métrique intégrée. Pour plus d'informations sur le format de métrique intégrée, consultez Intégration de métriques dans les journaux.

Pour analyser ces événements de journal, vous pouvez utiliser les méthodes suivantes :

- La section Lambda Insights de la CloudWatch console, comme expliqué dans. Affichage de vos métriques Lambda Insights
- Enregistrez les requêtes d'événements à l'aide de CloudWatch Logs Insights. Pour plus d'informations, consultez la section Analyse des données de journal avec CloudWatch Logs Insights.
- Métriques collectées dans l'espace de `LambdaInsights` noms, que vous pouvez représenter graphiquement à l'aide de CloudWatch métriques.

Voici un exemple d'événement de journal Lambda Insights avec format de métrique intégrée.

```
{
    "_aws": {
        "Timestamp": 1605034324256,
        "CloudWatchMetrics": [
            {
                "Namespace": "LambdaInsights",
                "Dimensions": [
                    [ "function_name" ],
                    [ "function_name", "version" ]
                ],
                "Metrics": [
                    { "Name": "memory_utilization", "Unit": "Percent" },
                    { "Name": "total_memory", "Unit": "Megabytes" },
                    { "Name": "used_memory_max", "Unit": "Megabytes" },
                    { "Name": "cpu_total_time", "Unit": "Milliseconds" },
                    { "Name": "tx_bytes", "Unit": "Bytes" },
                    { "Name": "rx_bytes", "Unit": "Bytes" },
                    { "Name": "total_network", "Unit": "Bytes" },
                    { "Name": "init_duration", "Unit": "Milliseconds" }
                ]
            }
        ],
        "LambdaInsights": {
            "ShareTelemetry": true
        }
    },
```

```
    "event_type": "performance",
    "function_name": "cpu-intensive",
    "version": "Blue",
    "request_id": "12345678-8bcc-42f7-b1de-123456789012",
    "trace_id": "1-5faae118-12345678901234567890",
    "duration": 45191,
    "billed_duration": 45200,
    "billed_mb_ms": 11571200,
    "cold_start": true,
    "init_duration": 130,
    "tmp_free": 538329088,
    "tmp_max": 551346176,
    "threads_max": 11,
    "used_memory_max": 63,
    "total_memory": 256,
    "memory_utilization": 24,
    "cpu_user_time": 6640,
    "cpu_system_time": 50,
    "cpu_total_time": 6690,
    "fd_use": 416,
    "fd_max": 32642,
    "tx_bytes": 4434,
    "rx_bytes": 6911,
    "timeout": true,
    "shutdown_reason": "Timeout",
    "total_network": 11345,
    "agent_version": "1.0.72.0",
    "agent_memory_avg": 10,
    "agent_memory_max": 10
}
```


Utilisez Contributor Insights pour analyser les données à haute cardinalité

Vous pouvez utiliser Contributor Insights pour analyser les données des journaux et 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. Cela vous aide à trouver les principaux intervenants et à comprendre qui ou ce qui a un impact sur les performances du système. Par exemple, vous pouvez trouver des hôtes défectueux, identifier les utilisateurs réseau les plus lourds ou trouver les URL qui génèrent le plus d'erreurs.

Vous pouvez créer vos règles à partir de zéro, et lorsque vous les utilisez, AWS Management Console vous pouvez également utiliser les exemples de règles qui ont AWS été créés. Les règles définissent les champs de journal que vous souhaitez utiliser pour définir des contributeurs, tels qu'`IpAddress`. Vous pouvez également filtrer les données du journal pour rechercher et analyser le comportement des contributeurs individuels.

CloudWatch fournit également des règles intégrées que vous pouvez utiliser pour analyser les métriques d'autres AWS services.

Toutes les règles analysent les données entrantes en temps réel.

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 créer des règles Contributor Insights dans ce compte de surveillance afin d'analyser les groupes de journaux dans les comptes sources et dans le compte de surveillance. Vous pouvez également créer une règle unique qui analyse les groupes de journaux dans plusieurs comptes. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

> **Note**
>
> Si vous utilisez Contributor Insights, vous êtes facturé pour chaque occurrence d'un événement de journal correspondant à une règle. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

Rubriques

Création d'une règle Contributor Insights

Vous pouvez créer des règles pour analyser les données du journal. Tous les journaux au format JSON ou CLF (Common Log Format) peuvent être évalués. Cela inclut vos journaux personnalisés

qui suivent l'un de ces formats et les journaux provenant de AWS services tels que les journaux de flux Amazon VPC, les journaux de requêtes DNS Amazon Route 53, les journaux de conteneurs Amazon ECS et les journaux provenant d' AWS CloudTrail Amazon, d' SageMakerAmazon RDS et d'API AWS AppSync Gateway.

Dans une règle, lorsque vous spécifiez des noms ou des valeurs de champ, toutes les correspondances sont sensibles à la casse.

Vous pouvez utiliser des exemples de règles intégrés lorsque vous créez une règle ou vous pouvez créer votre propre règle à partir de zéro. Contributor Insights inclut des exemples de règles pour les types de journaux suivants :

- Journaux Amazon API Gateway
- Journaux de requêtes DNS publics Amazon Route 53
- Journaux de requête Amazon Route 53 Resolver
- CloudWatch journaux de Container Insights
- Journaux de flux VPC

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 créer des règles Contributor Insights pour les groupes de journaux dans les comptes sources liés à ce compte de surveillance, en plus de créer des règles pour les groupes de journaux dans le compte de surveillance. Vous pouvez également configurer une règle unique qui surveille les groupes de journaux dans différents comptes. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

> **Important**
>
> Lorsque vous accordez l'`cloudwatch:PutInsightRule`autorisation à un utilisateur, celui-ci peut par défaut créer une règle qui évalue n'importe quel groupe de CloudWatch journaux dans Logs. Vous pouvez ajouter des conditions de politique IAM qui limitent ces autorisations pour qu'un utilisateur inclue et exclue des groupes de journaux spécifiques. Pour plus d'informations, consultez Utilisation de clés de condition pour limiter l'accès des utilisateurs Contributor Insights aux groupes de journaux.

Pour créer une règle à l'aide d'un exemple de règle intégré

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation, choisissez Insights, puis choisissez Contributor Insights.
3. Choisissez Créer une règle.
4. Pour Select log group(s) (Sélectionner le(s) groupe(s) de journaux), sélectionnez le ou les groupes de journaux que la règle doit surveiller. Vous pouvez sélectionner jusqu'à 20 groupes de journaux. Si vous êtes connecté à un compte de surveillance configuré pour l'observabilité CloudWatch entre comptes, vous pouvez sélectionner des groupes de journaux dans les comptes sources, et vous pouvez également définir une règle unique pour analyser les groupes de journaux dans différents comptes.

    - (Facultatif) Pour sélectionner tous les groupes de journaux dont le nom commence par une chaîne spécifique, choisissez Select by prefix match (Sélectionner par correspondance du préfixe) dans la liste déroulante, puis saisissez le préfixe. S'il s'agit d'un compte de surveillance, vous pouvez éventuellement sélectionner les comptes dans lesquels effectuer la recherche, sinon tous les comptes sont sélectionnés.

    > **Note**
    >
    > Vous encourez des frais pour chaque événement de journal correspondant à votre règle. Si vous choisissez le Select by prefix match (Sélectionner par correspondance du préfixe) dans la liste déroulante, soyez attentif au nombre de groupes de journaux auxquels le préfixe peut correspondre. Si vous recherchez accidentellement plus de groupes de journaux que vous ne le souhaitez, des frais inattendus peuvent vous être facturés. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

5. Pour Rule type (Type de règle), choisissez Sample rule (Exemple de règle). Ensuite, choisissez Select sample rule (Sélectionner un exemple de règle) et sélectionnez la règle.
6. L'exemple de règle a rempli les champs Log format (Format du journal), Contribution, Filters (Filtres) et Aggregate on (Regrouper par). Vous pouvez ajuster ces valeurs, si vous le souhaitez.
7. Choisissez Suivant.
8. Pour Rule name (Nom de la règle), entrez un nom. Les caractères valides sont A-Z, a-z, 0-9, (trait d'union), (trait de soulignement) et (point).
9. Indiquez si vous souhaitez créer la règle dans un état désactivé ou activé. Si vous choisissez de l'activer, la règle commence immédiatement à analyser vos données. Vous encourez des frais lorsque vous exécutez des règles activées. Pour en savoir plus, consultez Tarification Amazon CloudWatch.

Contributor Insights analyse uniquement les nouveaux événements de journal après la création d'une règle. Une règle ne peut pas traiter les événements des journaux précédemment traités par CloudWatch Logs.

10. (Facultatif) Pour Tags (Balises), ajoutez une ou plusieurs paires clé/valeur comme balises pour cette règle. Les balises peuvent vous aider à identifier et à organiser vos AWS ressources et à suivre vos AWS coûts. Pour plus d'informations, consultez Marquer vos ressources Amazon CloudWatch .
11. Sélectionnez Créer.

Pour créer une règle à partir de zéro

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Contributor Insights.
3. Choisissez Create rule (Créer une règle).

4. Pour Select log group(s) (Sélectionner le(s) groupe(s) de journaux), sélectionnez le ou les groupes de journaux que la règle doit surveiller. Vous pouvez sélectionner jusqu'à 20 groupes de journaux. Si vous êtes connecté à un compte de surveillance configuré pour l'observabilité CloudWatch entre comptes, vous pouvez sélectionner des groupes de journaux dans les comptes sources, et vous pouvez également définir une règle unique pour analyser les groupes de journaux dans différents comptes.

    - (Facultatif) Pour sélectionner tous les groupes de journaux dont le nom commence par une chaîne spécifique, choisissez Select by prefix match (Sélectionner par correspondance du préfixe) dans la liste déroulante, puis saisissez le préfixe.

> **Note**
>
> Vous encourez des frais pour chaque événement de journal correspondant à votre règle. Si vous choisissez le Select by prefix match (Sélectionner par correspondance du préfixe) dans la liste déroulante, soyez attentif au nombre de groupes de journaux auxquels le préfixe peut correspondre. Si vous recherchez accidentellement plus de groupes de journaux que vous ne le souhaitez, des frais inattendus peuvent vous être facturés. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

5. Pour Rule type (Type de règle), choisissez Custom rule (Règle personnalisée).
6. Pour le Format de journal, choisissez JSON ou CLF.
7. Vous pouvez terminer la création de la règle à l'aide de l'Assistant ou en choisissant l'onglet Syntaxe et en spécifiant manuellement la syntaxe de la règle.

   Pour continuer à utiliser l'Assistant, procédez comme suit :

   a. Pour Contribution, Clé, entrez un type de contributeur sur lequel vous souhaitez créer un rapport. Le rapport affiche les valeurs top-N pour ce type de contributeur.

      Les entrées valides sont tout champ de journal qui a des valeurs. Exemples : **`requestId`**, **`sourceIPaddress`** et **`containerID`**.

      Pour obtenir des informations sur la recherche des noms de champs de journal pour les journaux d'un groupe de journaux spécifique, consultez la rubrique Recherche de champs de journal.

      Les clés supérieures à 1 Ko sont tronquées à 1 Ko.

   b. (Facultatif) Choisissez Add new key (Ajouter une nouvelle clé) pour ajouter d'autres clés. Vous pouvez inclure jusqu'à quatre clés dans une règle. Si vous entrez plusieurs clés, les contributeurs du rapport sont définis par des combinaisons de valeurs uniques des clés. Par exemple, si vous spécifiez trois clés, chaque combinaison unique de valeurs pour les trois clés est comptée comme un contributeur unique.

   c. (Facultatif) Si vous souhaitez ajouter un filtre qui réduit la portée de vos résultats, choisissez Add filter (Ajouter un filtre). Pour Match (Correspondance), saisissez nom du champ de journal que vous souhaitez filtrer. Pour Condition, choisissez un opérateur de comparaison et saisissez une valeur que vous souhaitez filtrer.

      Vous pouvez ajouter jusqu'à quatre filtres dans une règle. Plusieurs filtres sont joints par la logique AND, de sorte que seuls les événements de journal qui correspondent à tous ces filtres sont évalués.

      > **Note**
      >
      > Les tableaux qui suivent des opérateurs de comparaison, tels que `In`, `NotIn`, ou `StartsWith`, peuvent inclure jusqu'à 10 valeurs de chaîne. Pour plus

d'informations sur la syntaxe des règles de Contributor Insights, veuillez consulter Syntaxe des règles Contributor Insights.

d. Pour Aggregate on (Regrouper dans), choisissez Count (Nombre) ou Sum (Somme). Si vous choisissez Count (Nombre), le classement des contributeurs est basé sur le nombre d'occurrences. Si vous choisissez Sum (Somme), le classement est basé sur la somme agrégée des valeurs du champ que vous spécifiez pour Contribution, Value (Valeur).

8. Pour entrer votre règle en tant qu'objet JSON au lieu d'utiliser l'assistant, procédez comme suit :

   a. Choisissez l'onglet Syntaxe.

   b. Dans Corps de la règle, entrez l'objet JSON de votre règle. Pour de plus amples informations sur la syntaxe des règles, veuillez consulter Syntaxe des règles Contributor Insights.

9. Choisissez Suivant.

10. Pour Rule name (Nom de la règle), entrez un nom. Les caractères valides sont A-Z, a-z, 0-9, « - », « _ » et « . ».

11. Indiquez si vous souhaitez créer la règle dans un état désactivé ou activé. Si vous choisissez de l'activer, la règle commence immédiatement à analyser vos données. Vous encourez des frais lorsque vous exécutez des règles activées. Pour en savoir plus, consultez Tarification Amazon CloudWatch.

    Contributor Insights analyse uniquement les nouveaux événements de journal après la création d'une règle. Une règle ne peut pas traiter les événements des journaux précédemment traités par CloudWatch Logs.

12. (Facultatif) Pour Tags (Balises), ajoutez une ou plusieurs paires clé/valeur comme balises pour cette règle. Les balises peuvent vous aider à identifier et à organiser vos AWS ressources et à suivre vos AWS coûts. Pour plus d'informations, consultez Marquer vos ressources Amazon CloudWatch .

13. Choisissez Suivant.

14. Confirmez les paramètres que vous avez saisis, puis sélectionnez Create rule (Créer la règle).

Vous pouvez désactiver, activer ou supprimer les règles que vous avez créées.

Pour activer, désactiver ou supprimer une règle dans Contributor Insights

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation, choisissez Contributor Insights.
3. Dans la liste des règles, cochez la case en regard d'une règle unique.

   Les règles intégrées sont créées par AWS les services et ne peuvent pas être modifiées, désactivées ou supprimées.
4. Choisissez Actions, puis choisissez l'option souhaitée.

Recherche de champs de journal

Lorsque vous créez une règle, vous devez connaître les noms des champs dans les entrées de journal d'un groupe de journaux.

Pour rechercher les champs de journal dans un groupe de journaux

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sous Logs (Journaux), choisissez Insights.
3. Sélectionnez un ou plusieurs groupes de journaux à interroger, au-dessus de l'éditeur de requête.

   Lorsque vous sélectionnez un groupe de CloudWatch journaux, Logs Insights détecte automatiquement les champs des données du groupe de journaux et les affiche dans le volet droit de la section Champs découverts.

Syntaxe des règles Contributor Insights

Cette section explique la syntaxe des règles de Contributor Insights. Utilisez cette syntaxe uniquement lorsque vous créez une règle en entrant un bloc JSON. Si vous utilisez l'Assistant pour créer une règle, vous n'avez pas besoin de connaître la syntaxe. Pour de plus amples informations sur la création de règles à l'aide de l'Assistant, veuillez consulter Création d'une règle Contributor Insights.

Toutes les correspondances entre les règles et les noms et valeurs des champs d'événements de journal sont sensibles à la casse.

L'exemple suivant illustre la syntaxe des journaux JSON.

```
{
    "Schema": {
```

```
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*",
        "Log-group-name2"
    ],
    "LogFormat": "JSON",
    "Contribution": {
        "Keys": [
            "$.ip"
        ],
        "ValueOf": "$.requestBytes",
        "Filters": [
            {
                "Match": "$.httpMethod",
                "In": [
                    "PUT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Champs dans les règles de Contributor Insights

Schema

La valeur de `Schema` pour une règle qui analyse les données des CloudWatch journaux doit toujours être `{"Name": "CloudWatchLogRule", "Version": 1}`

LogGroupNames

Tableau de chaînes. Pour chaque élément du tableau, vous pouvez éventuellement utiliser * à la fin d'une chaîne pour inclure tous les groupes de journaux dont les noms commencent par ce préfixe.

Veillez à utiliser des caractères génériques avec des noms de groupes de journaux. Vous encourez des frais pour chaque événement de journal correspondant à une règle. Si vous recherchez accidentellement plus de groupes de journaux que vous ne le souhaitez, des frais inattendus peuvent vous être facturés. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

LogGroupARN

Si vous créez cette règle dans un CloudWatch compte de surveillance d'observabilité entre comptes, vous pouvez l'utiliser `LogGroupARNs` pour spécifier des groupes de journaux dans les comptes sources liés au compte de surveillance, et pour spécifier des groupes de journaux dans le compte de surveillance lui-même. Vous devez spécifier `LogGroupNames` ou `LogGroupARNs` dans votre règle, mais pas les deux.

`LogGroupARNs` est un tableau de chaînes de caractères. Pour chaque élément du tableau, vous pouvez utiliser * comme caractère générique dans certaines situations. Par exemple, vous pouvez spécifier `arn:aws:logs:us-west-1:*:log-group/MyLogGroupName2` pour spécifier les groupes de journaux nommés `MyLogGroupName2` dans tous les comptes sources et dans le compte de surveillance, dans la région USA Ouest (Californie du Nord). Vous pouvez également spécifier `arn:aws:logs:us-west-1:111122223333:log-group/GroupNamePrefix*` pour spécifier tous les groupes de journaux dans la région USA Ouest (Californie du Nord) en 111122223333 dont le nom commence par `GroupNamePrefix`.

Vous ne pouvez pas spécifier un identifiant de AWS compte partiel comme préfixe avec un joker.

Faites attention à l'utilisation de caractères génériques avec les ARN de groupes de journaux. Vous encourez des frais pour chaque événement de journal correspondant à une règle. Si vous recherchez accidentellement plus de groupes de journaux que vous ne le souhaitez, des frais inattendus peuvent vous être facturés. Pour plus d'informations, consultez Amazon CloudWatch Pricing.

LogFormat

Les valeurs valides sont `JSON` et `CLF`.

Contribution

Cet objet comprend un tableau `Keys` avec jusqu'à quatre membres, éventuellement un seul `ValueOf`, et éventuellement un tableau comportant jusqu'à quatre `Filters`.

Clés

Tableau comportant jusqu'à quatre champs de journal utilisés comme dimensions pour classer les contributeurs. Si vous entrez plusieurs clés, chaque combinaison unique de valeurs pour les clés est comptabilisée comme un contributeur unique. Les champs doivent être spécifiés à l'aide de la notation de format de propriété JSON.

ValueOf

(Facultatif) Spécifiez cette valeur uniquement lorsque vous spécifiez `Sum` comme valeur de `AggregateOn`. `ValueOf` spécifie un champ de journal avec des valeurs numériques. Dans ce type de règle, les contributeurs sont classés en fonction de leur somme de la valeur de ce champ, au lieu de leur nombre d'occurrences dans les entrées du journal. Par exemple, si vous souhaitez trier les contributeurs en fonction de leur `BytesSent` total sur une période, vous devez définir `ValueOf` sur `BytesSent` et spécifier `Sum` pour `AggregateOn`.

Filtres

(Facultatif) Spécifie un tableau pouvant comporter jusqu'à quatre filtres pour affiner les événements de journal inclus dans le rapport. Si vous spécifiez plusieurs filtres, Contributor Insights les évalue avec un opérateur AND logique. Vous pouvez utiliser cela pour filtrer les événements de journal non pertinents dans votre recherche ou vous pouvez l'utiliser pour sélectionner un seul contributeur afin d'analyser son comportement.

Chaque membre du tableau doit inclure un champ `Match` et un champ indiquant le type d'opérateur correspondant à utiliser.

Le champ `Match` spécifie un champ journal à évaluer dans le filtre. Le champ journal est spécifié à l'aide de la notation de format de propriété JSON.

Le champ opérateur correspondant doit être l'un des éléments suivants : `In`, `NotIn`, `StartsWith`, `GreaterThan`, `LessThan`, `EqualTo`, `NotEqualTo` ou `IsPresent`. Si le champ opérateur est `In`, `NotIn`, ou `StartsWith`, il est suivi d'un tableau de valeurs de chaîne à vérifier. Contributor Insights évalue le tableau de valeurs de chaîne avec un opérateur OR. Le tableau peut inclure jusqu'à 10 valeurs de chaîne.

Si le champ opérateur est `GreaterThan`, `LessThan`, `EqualTo` ou `NotEqualTo`, il est suivi d'une seule valeur numérique à comparer.

Si le champ opérateur est `IsPresent`, il est suivi de `true` ou de `false`. Cet opérateur met en correspondance les événements de journal selon que le champ de journal spécifié est présent ou non dans l'événement de journal. Le `isPresent` fonctionne uniquement avec des valeurs dans le nœud feuille des propriétés JSON. Par exemple, un filtre qui recherche des correspondances de `c-count` n'évalue pas un événement de journal avec une valeur de `details.c-count.c1`.

Consultez les quatre exemples de filtres suivants :

```
{"Match": "$.httpMethod", "In": [ "PUT", ] }
```

```
{"Match": "$.StatusCode", "EqualTo": 200 }
{"Match": "$.BytesReceived", "GreaterThan": 10000}
{"Match": "$.eventSource", "StartsWith": [ "ec2", "ecs" ] }
```

AggregateOn

Les valeurs valides sont `Count` et `Sum`. Spécifie si le rapport doit être agrégé en fonction d'un nombre d'occurrences ou d'une somme des valeurs du champ spécifié dans le champ `ValueOf`.

Notation de format de propriété JSON

Les champs `Keys`, `ValueOf` et `Match` suivent le format de propriété JSON avec notation point, où $ représente la racine de l'objet JSON. Ceci est suivi d'un point, puis d'une chaîne alphanumérique avec le nom de la sous-propriété. Plusieurs niveaux de propriétés sont pris en charge.

Le premier caractère de la chaîne doit être une lettre majuscule ou minuscule. Les caractères suivants de la chaîne peuvent être des lettres majuscules, minuscules ou des chiffres de 0 à 9.

La liste suivante illustre des exemples valides de format de propriété JSON :

```
$.userAgent
$.endpoints[0]
$.users[1].name
$.requestParameters.instanceId
```

Champ supplémentaire dans les règles des journaux CLF

Les événements du journal Common Log Format (CLF) n'ont pas de noms pour les champs comme le fait JSON. Pour fournir les champs à utiliser pour les règles Contributor Insights, un événement de journal CLF peut être traité comme un tableau avec un index commençant à partir de 1. Vous pouvez spécifier le premier champ comme **"1"**, le second champ comme **"2"**, etc.

Pour faciliter la lecture d'une règle pour un journal CLF, vous pouvez utiliser `Fields`. Cela vous permet de fournir un alias de dénomination pour les emplacements de champ CLF. Par exemple, vous pouvez spécifier que l'emplacement « 4 » est une adresse IP. Une fois spécifié, `IpAddress` peut être utilisé comme propriété dans `Keys`, `ValueOf` et `Filters` dans la règle.

Voici un exemple de règle pour un journal CLF qui utilise le champ `Fields`.

```
{
    "Schema": {
```

```
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "API-Gateway-Access-Logs*"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "IpAddress",
        "7": "StatusCode"
    },
    "Contribution": {
        "Keys": [
            "IpAddress"
        ],
        "Filters": [
            {
                "Match": "StatusCode",
                "EqualTo": 200
            }
        ]
    },
    "AggregateOn": "Count"
}
```

Exemples de règles de Contributor Insights

Cette section contient des exemples illustrant les cas d'utilisation des règles de Contributor Insights.

Journaux de flux VPC : transfert d'octets par adresse IP source et destination

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "4": "srcaddr",
        "5": "dstaddr",
```

```
        "10": "bytes"
    },
    "Contribution": {
        "Keys": [
            "srcaddr",
            "dstaddr"
        ],
        "ValueOf": "bytes",
        "Filters": []
    },
    "AggregateOn": "Sum"
}
```

Journaux de flux VPC : nombre le plus élevé de requêtes HTTPS

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "5": "destination address",
        "7": "destination port",
        "9": "packet count"
    },
    "Contribution": {
        "Keys": [
            "destination address"
        ],
        "ValueOf": "packet count",
        "Filters": [
            {
                "Match": "destination port",
                "EqualTo": 443
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Journaux de flux VPC : connexions TCP rejetées

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Réponses NXDomain Route 53 par adresse source

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
```

```
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.rcode",
                "StartsWith": [
                    "NXDOMAIN"
                ]
            }
        ],
        "Keys": [
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Requêtes Route 53 Resolver par nom de domaine

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_name"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Requêtes Route 53 Resolver par type de requête et adresse source

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [],
        "Keys": [
            "$.query_type",
            "$.srcaddr"
        ]
    },
    "LogFormat": "JSON",
    "LogGroupNames": [
        "<loggroupname>"
    ]
}
```

Affichage des rapports de Contributor Insights

Pour afficher des graphiques des données de rapport et une liste classée des contributeurs trouvés par vos règles, procédez comme suit.

Pour afficher vos rapports de règles

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Contributor Insights.
3. Dans la liste des règles, choisissez le nom d'une règle.

   Le graphique affiche les résultats de la règle au cours des trois dernières heures. Le tableau sous le graphique montre les 10 principaux contributeurs.
4. Pour modifier le nombre de contributeurs indiqué dans le tableau, choisissez 10 contributeurs principaux en haut du graphique.
5. Pour filtrer le graphique afin d'afficher uniquement les résultats d'un seul contributeur, choisissez ce contributeur dans la légende du tableau. Pour afficher à nouveau tous les contributeurs, choisissez à nouveau le même contributeur dans la légende.

6. Pour modifier la plage de temps affichée dans le rapport, choisissez 15m (15 min), 30m (30 min), 1h (1 h), 2h (2 h), 3h (3 h) ou custom (personnalisé) en haut du graphique.

   La plage de temps maximale pour l'état est de 24 heures, mais vous pouvez choisir une fenêtre de 24 heures qui s'est produite il y a 15 jours. Pour choisir une fenêtre horaire dans le passé, choisissez personnalisée, absolue, puis spécifiez votre fenêtre horaire.
7. Pour modifier la durée de la période utilisée pour l'agrégation et le classement des contributeurs, choisissez la période en haut du graphique. L'affichage d'une période plus longue montre généralement un rapport plus fluide avec peu de pointes. Si vous choisissez une période plus courte, vous êtes plus susceptible de voir des pics.
8. Pour ajouter ce graphique à un CloudWatch tableau de bord, choisissez Ajouter au tableau de bord.
9. Pour ouvrir la fenêtre de requête CloudWatch Logs Insights, les groupes de journaux de ce rapport étant déjà chargés dans la zone de requête, choisissez Afficher les journaux.
10. Pour exporter les données du rapport vers votre Presse-papiers ou un fichier CSV, choisissez Exporter.

Graphique des métriques générées par les règles

Contributor Insights fournit une fonction mathématique métrique, `INSIGHT_RULE_METRIC`. Vous pouvez utiliser cette fonction pour ajouter les données d'un rapport Contributor Insights à un graphique dans l'onglet Metrics de la CloudWatch console. Vous pouvez également définir une alerte en fonction de cette fonction mathématique. Pour de plus amples informations sur les fonctions mathématiques de métrique, consultez Utilisation des mathématiques appliquées aux métriques.

Pour utiliser cette fonction mathématique de métrique, vous devez être connecté à un compte disposant à la fois des autorisations `cloudwatch:GetMetricData` et `cloudwatch:GetInsightRuleReport`.

La syntaxe est `INSIGHT_RULE_METRIC(`*`ruleName`*`, `*`metricName`*`)`. *`ruleName`* est le nom d'une règle de Contributor Insights. *`metricName`* est l'une des valeurs de la liste suivante. La valeur de *`metricName`* détermine le type de données renvoyé par la fonction mathématique.

- `UniqueContributors` — le nombre de contributeurs uniques pour chaque point de données.
- `MaxContributorValue` — la valeur du contributeur le plus important pour chaque point de données. L'identité du contributeur peut changer pour chaque point de données du graphique.

Si cette règle est agrégée par `Count`, le contributeur le plus important pour chaque point de données est celui qui a le plus d'occurrences au cours de cette période. Si la règle est agrégée par `Sum`, le contributeur principal est le contributeur dont la somme est la plus élevée dans le champ de journal spécifié par la valeur `Value` de la règle pendant cette période.

- `SampleCount` — le nombre de points de données correspondant à la règle.
- `Sum` — la somme des valeurs de tous les contributeurs pendant la période représentée par ce point de données.
- `Minimum` — la valeur minimale d'une observation unique au cours de la période représentée par ce point de données.
- `Maximum` — la valeur maximale d'une observation unique pendant la période représentée par ce point de données.
- `Average` — la valeur moyenne de tous les contributeurs au cours de la période représentée par ce point de données.

Définition d'une alerte sur les données de métrique de Contributor Insights

À l'aide de la fonction `INSIGHT_RULE_METRIC`, vous pouvez définir des alertes sur les métriques générées par Contributor Insights. Par exemple, vous pouvez créer une alerte en fonction du pourcentage de connexions de Transmission Control Protocol (TCP) rejetées. Pour commencer avec ce type d'alerte, vous pouvez créer des règles comme celles présentées dans les deux exemples suivants :

Exemple de règle : « RejectedConnectionsRule »

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
```

```
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            },
            {
                "Match": "action",
                "In": [
                    "REJECT"
                ]
            }
        ]
    },
    "AggregateOn": "Sum"
}
```

Exemple de règle : « TotalConnectionsRule »

```
{
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupNames": [
        "/aws/containerinsights/sample-cluster-name/flowlogs"
    ],
    "LogFormat": "CLF",
    "Fields": {
        "3": "interfaceID",
        "4": "sourceAddress",
        "8": "protocol",
        "13": "action"
    },
    "Contribution": {
        "Keys": [
            "interfaceID",
```

```
            "sourceAddress"
        ],
        "Filters": [
            {
                "Match": "protocol",
                "EqualTo": 6
            }
    "AggregateOn": "Sum"
}
```

Après avoir créé vos règles, vous pouvez sélectionner l'onglet Mesures dans la CloudWatch console, dans lequel vous pouvez utiliser les exemples d'expressions mathématiques métriques suivants pour représenter graphiquement les données rapportées par Contributor Insights :

Exemple : expressions mathématiques appliquées aux métriques

```
e1 INSIGHT_RULE_METRIC("RejectedConnectionsRule", "Sum")
e2 INSIGHT_RULE_METRIC("TotalConnectionsRule", "Sum")
e3 (e1/e2)*100
```

Dans cet exemple, l'expression mathématique appliquée à une métrique `e3` renvoie toutes les connexions TCP rejetées. Si vous souhaitez être averti lorsque 20 % des connexions TCP sont rejetées, vous pouvez modifier l'expression en changeant le seuil de `100` pour `20`.

> **Note**
>
> Vous pouvez définir une alerte sur une métrique que vous surveillez à partir de la section Metrics (Métriques). Pendant que vous êtes sur l'onglet Graphed metrics (Métriques sous forme de graphique), vous pouvez sélectionner l'icône Create alarm (Créer une alerte) sous la colonne Actions. L'icône Create alarm Créer une alerte) ressemble à une cloche.

Pour de plus amples informations sur la représentation graphique des métriques et l'utilisation des fonctions mathématiques des métriques, veuillez consulter la section suivante : Ajouter une expression mathématique à un CloudWatch graphique.

Utilisation des règles intégrées de Contributor Insights

Vous pouvez utiliser les règles intégrées de Contributor Insights pour analyser les statistiques d'autres AWS services. Les services suivants prennent en charge les règles intégrées :

- Contributor Insights pour Amazon DynamoDB dans le Guide du développeur Amazon DynamoDB.
- Utilisation des règles intégrées de Contributor Insights dans le Guide AWS PrivateLink .

Informations sur les CloudWatch applications Amazon

Amazon CloudWatch Application Insights facilite l'observabilité de vos applications et des AWS ressources sous-jacentes. Cette solution vous aide à configurer les meilleures surveillances pour vos ressources d'application, afin d'analyser en continu les données à la recherche de problèmes liés à vos applications. Application Insights, qui s'appuie sur d'SageMakerautres AWS technologies, fournit des tableaux de bord automatisés qui indiquent les problèmes potentiels liés aux applications surveillées, ce qui vous aide à isoler rapidement les problèmes récurrents liés à vos applications et à votre infrastructure. La visibilité accrue fournie par Application Insights sur l'état de vos applications réduit le temps moyen de réparation (MTTR) pour résoudre vos problèmes d'applications.

Lorsque vous ajoutez vos applications à Amazon CloudWatch Application Insights, celui-ci analyse les ressources des applications, recommande et configure les métriques et les connexions CloudWatchpour les composants de l'application. Des exemples de composants d'application peuvent inclure des bases de données backend SQL Server et des niveaux Microsoft IIS/Web. Application Insights analyse les modèles de métriques à l'aide de données d'historique pour identifier les anomalies et détecte en continu les erreurs et les exceptions à partir de vos journaux d'applications, de systèmes d'exploitation et d'infrastructure. La solution met en corrélation ces observations à l'aide d'une combinaison d'algorithmes de classification et de règles préintégrées. Elle crée ensuite automatiquement des tableaux de bord qui affichent les observations pertinentes et les informations relatives à la gravité du problème afin de vous aider à hiérarchiser vos actions. Pour les problèmes courants liés aux piles d'applications .NET et SQL, tels que la latence des applications, les échecs de sauvegarde SQL Server, les fuites de mémoire, les requêtes HTTP volumineuses et les opérations d'I/O annulées, elle fournit des informations supplémentaires vous indiquant la cause potentielle et les étapes menant à la résolution. L'intégration intégrée à AWS SSM vous OpsCenter permet de résoudre les problèmes en exécutant le document Systems Manager Automation correspondant.

Sections

- Qu'est-ce qu'Amazon CloudWatch Application Insights ?
- Comment fonctionne Amazon CloudWatch Application Insights
- Commencez avec Amazon CloudWatch Application Insights
- Observabilité inter-comptes Application Insights

- Utilisation de configurations de composants
- Création et configuration de la surveillance CloudWatch d'Application Insights à l'aide CloudFormation de modèles
- Didacticiel : Configurer la surveillance pour SAP ASE
- Didacticiel : configurer la surveillance pour SAP HANA
- Tutoriel : Configuration de la surveillance pour SAP NetWeaver
- Afficher et résoudre les problèmes détectés par Amazon CloudWatch Application Insights
- Logs et statistiques pris en charge par Amazon CloudWatch Application Insights

Qu'est-ce qu'Amazon CloudWatch Application Insights ?

CloudWatch Application Insights vous aide à surveiller vos applications qui utilisent des instances Amazon EC2 ainsi que d'autres ressources applicatives. Cette solution identifie et configure des métriques, des journaux et des alertes clés sur vos ressources d'application et votre pile technologique (par exemple, votre base de données Microsoft SQL Server, les serveurs web (IIS) et d'applications, le système d'exploitation, les équilibreurs de charge et les files d'attente). Elle surveille en permanence les métriques et les journaux afin de détecter et de corréler les anomalies et les erreurs. Lorsque des erreurs et des anomalies sont détectées, Application Insights génère CloudWatch des événements que vous pouvez utiliser pour configurer des notifications ou prendre des mesures. Afin de faciliter le dépannage, elle crée des tableaux de bord automatisés pour les problèmes détectés, qui incluent les anomalies de métriques corrélées, les erreurs de journalisation, ainsi que des informations supplémentaires vous indiquant la cause racine potentielle. Les tableaux de bord automatisés vous aident à lancer des actions correctives pour maintenir l'intégrité de vos applications et pour empêcher tout impact sur les utilisateurs finaux de vos applications. Il crée également OpsItems pour que vous puissiez résoudre les problèmes à l'aide de AWS SSM. OpsCenter

Vous pouvez configurer des compteurs importants, tels que la transaction d'écriture en miroir, la longueur de la file d'attente de restauration et le délai de transaction, ainsi que l'ouverture de journaux d'événements Windows. CloudWatch Lorsqu'un événement ou un problème de basculement survient avec votre charge de travail SQL HA, tel qu'un accès restreint pour interroger une base de données cible, CloudWatch Application Insights fournit des informations automatisées.

CloudWatch Application Insights s'intègre AWS Launch Wizardpour fournir une expérience de configuration de surveillance en un clic pour le déploiement de charges de travail SQL Server HA sur. AWS Lorsque vous sélectionnez l'option permettant de configurer la surveillance et les informations

avec Application Insights sur la console Launch Wizard, CloudWatch Application Insights configure automatiquement les métriques, les journaux et les alarmes pertinents CloudWatch, et commence à surveiller les charges de travail nouvellement déployées. Vous pouvez consulter les informations automatisées et les problèmes détectés, ainsi que l'état de vos charges de travail SQL Server HA, sur la CloudWatch console.

Table des matières

Fonctionnalités

Application Insights fournit les fonctions ci-dessous.

Configuration automatique des moniteurs pour les ressources d'application

CloudWatch Application Insights réduit le temps nécessaire à la configuration de la surveillance de vos applications. Pour ce faire, il analyse les ressources de votre application, fournit une liste personnalisable de métriques et de journaux recommandés, et les configure CloudWatch pour fournir la visibilité nécessaire sur les ressources de vos applications, telles qu'Amazon EC2 et Elastic Load Balancers (ELB). Cette solution configure également des alertes dynamiques sur des métriques surveillées. Les alertes sont automatiquement mises à jour en fonction des anomalies détectées au cours des deux dernières semaines.

Détection et notification des problèmes

CloudWatch Application Insights détecte les signes de problèmes potentiels liés à votre application, tels que les anomalies métriques et les erreurs de journal. La solution met en corrélation ces observations pour faire apparaître les problèmes potentiels de votre application. Il génère ensuite CloudWatch des événements, qui peuvent être configurés pour recevoir des notifications ou prendre des mesures. Vous n'avez donc plus besoin de créer des alertes individuelles en fonction des métriques ou des erreurs de journaux.

Résolution des problèmes

CloudWatch Application Insights crée des tableaux de bord CloudWatch automatiques pour les problèmes détectés. Les tableaux de bord affichent des détails sur le problème, notamment les anomalies métriques et les erreurs de journaux associées, afin de faciliter le dépannage. Ils fournissent également des informations supplémentaires qui vous indiquent les causes potentielles des anomalies et des erreurs.

Concepts

Les concepts suivants sont importants pour comprendre comment Application Insights surveille votre application.

Composant

Un groupement automatique, autonome ou personnalisé de ressources similaires qui constituent une application. Pour garantir une surveillance optimale, il est recommandé de regrouper les ressources similaires dans des composants personnalisés.

Observation

Un événement individuel (anomalie métrique, erreur du journal ou exception) qui est détecté dans une application ou une ressource d'application.

Problème

Les problèmes sont détectés par la corrélation, la classification ou le groupement d'observations associées.

Pour les définitions des autres concepts clés d' CloudWatch Application Insights, consultez Amazon CloudWatch Concepts.

Tarification

CloudWatch Application Insights définit les mesures et les journaux recommandés pour les ressources d'application sélectionnées à l'aide de CloudWatch métriques, de journaux et d'événements pour les notifications relatives aux problèmes détectés. Ces fonctionnalités sont facturées sur votre AWS compte conformément à la CloudWatch tarification. Pour les problèmes détectés, des SSM OpsItems sont également créés par Application Insights pour vous informer des problèmes. En outre, Application Insights crée les paramètres du magasin de paramètres SSM pour configurer les CloudWatch agents sur vos instances. Les fonctions d'Amazon EC2 Systems Manager sont facturées selon la tarification SSM. Vous n'êtes pas facturé pour l'aide à la configuration, la surveillance, l'analyse des données ou la détection des problèmes.

Coûts liés à CloudWatch Application Insights

Les coûts pour Amazon EC2 comprennent l'utilisation des fonctions suivantes :

- CloudWatch Agent
  - CloudWatch Groupes de journaux d'agents
  - CloudWatch Métriques relatives aux agents
  - Groupes de journaux Prometheus (pour les charges de travail JMX)

Les coûts pour toutes les ressources comprennent l'utilisation des fonctions suivantes :

- CloudWatch alarmes (majeure partie du coût)
- SSM OpsItems (coût minimal)

Exemple de calcul des coûts

Les coûts dans cet exemple sont considérés selon le scénario suivant.

Vous avez créé un groupe de ressources qui comprend les éléments suivants :

- Une instance Amazon EC2 avec SQL Server installé.
- Un volume Amazon EBS attaché.

Lorsque vous intégrez ce groupe de ressources à CloudWatch Application Insights, la charge de travail SQL Server installée sur l'instance Amazon EC2 est détectée. CloudWatch Application Insights commence à surveiller les indicateurs suivants.

Les métriques suivantes sont surveillées pour l'instance SQL Server :

- CPUUtilization
- StatusCheckFailed
- % d'octets validés de mémoire en cours d'utilisation
- Mo de mémoire disponible
- Total octets interface réseau/sec
- Utilisation en % du fichier de pagination
- Temps disque en % de disque physique

- Temps de traitement en % du processeur
- SQLServer : Taux de réussite du cache du gestionnaire des buffers
- SQLServer : Espérance de vie de la page du gestionnaire des buffers
- SQLServer : Statistiques générales - Processus bloqués
- SQLServer : Statistiques générales - Connexions utilisateurs
- SQLServer : Verrouillages - Nombre de blocages/s
- SQLServer : Statistiques SQL - Requêtes par lots/sec
- Longueur de la file d'attente du processeur système

Les métriques suivantes sont surveillées pour les volumes attachés à l'instance SQL Server :

- VolumeReadBytes
- VolumeWriteBytes
- VolumeReadOps
- VolumeWriteOps
- VolumeTotalReadTime
- VolumeTotalWriteTime
- VolumeIdleTime
- VolumeQueueLength
- VolumeThroughputPercentage
- VolumeConsumedReadWriteOps
- BurstBalance

Pour ce scénario, les coûts sont calculés en fonction de la page de CloudWatch tarification et de la page de tarification SSM :

- Métriques personnalisées

  Pour ce scénario, 13 des métriques ci-dessus sont émises à CloudWatch l'aide de l' CloudWatch agent. Ces métriques sont traitées comme des métriques personnalisées. Le coût de chaque métrique personnalisée est de 0,3 USD/mois. Le coût total de ces métriques personnalisées est de 13 * 0,3 USD = 3,90 USD/mois.

- Alarmes

  Dans ce scénario, CloudWatch Application Insights surveille 26 métriques au total, ce qui crée 26 alarmes. Le coût de chaque alarme est de 0,1 USD/mois. Le coût total des alertes est de 26 * 0,1 USD = 2,60 USD/mois.

- Ingestion de données et journaux d'erreurs

  Le coût de l'ingestion de données est de 0,05 USD/Go et le stockage du journal des erreurs du serveur SQL est de 0,03 USD/Go. Le coût total de l'ingestion de données et du journal des erreurs est de 0,05 USD/Go + 0,03 USD/Go = 0,08 USD/Go.

- Amazon EC2 Systems Manager OpsItems

  Un SSM OpsItem est créé pour chaque problème détecté par CloudWatch Application Insights. Pour un nombre n de problèmes dans votre application, le coût total est de 0,00267 USD * n/mois.

Services connexes

Les services suivants sont utilisés avec CloudWatch Application Insights :

AWS Services connexes

- Amazon CloudWatch fournit une visibilité à l'échelle du système sur l'utilisation des ressources, les performances des applications et la santé opérationnelle. Il collecte et suit les métriques, envoie des notifications d'alarme, met automatiquement à jour les ressources que vous surveillez en fonction des règles que vous définissez et vous permet de surveiller vos propres métriques personnalisées. CloudWatch Application Insights est lancé via, CloudWatch en particulier, les tableaux de bord opérationnels CloudWatch par défaut. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- CloudWatch Container Insights collecte, agrège et résume les métriques et les journaux de vos applications conteneurisées et de vos microservices. Vous pouvez utiliser Container Insights pour surveiller les plates-formes Amazon ECS, Amazon Elastic Kubernetes Service et Kubernetes sur Amazon EC2. Lorsque Application Insights est activé sur les consoles Container Insights ou Application Insights, Application Insights affiche les problèmes détectés sur votre tableau de bord Container Insights. Pour plus d'informations, consultez Container Insights.
- Amazon DynamoDB est un service de base de données NoSQL totalement géré qui vous permet de ne pas avoir à assurer les charges administratives liées au fonctionnement et à la mise à l'échelle d'une base de données distribuée, vous n'avez pas à vous soucier de l'allocation, du

paramétrage, de la configuration et de la réplication du matériel, ainsi que des correctifs logiciels ou de la mise à l'échelle des clusters. DynamoDB offre également le chiffrement au repos, qui élimine la lourdeur opérationnelle et la complexité induites par la protection des données sensibles.

- Amazon EC2 fournit une capacité de calcul évolutive dans le AWS cloud. Vous pouvez utiliser Amazon EC2 pour lancer autant de serveurs virtuels que nécessaire, configurer la sécurité et la mise en réseau, et gérer le stockage. Vous pouvez monter ou descendre en puissance afin de gérer les modifications en termes d'exigences ou de pics de popularité, et réduire ainsi le besoin de prévoir le trafic. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon EC2 pour les instances Linux ou le Guide de l'utilisateur Amazon EC2 pour les instances Windows.
- Amazon Elastic Block Store (Amazon EBS) fournit des volumes de stockage niveau bloc à utiliser avec les instances Amazon EC2. Les volumes d'Amazon EBS se comportent comme des périphériques de stockage en mode blocs bruts non formatés. Vous pouvez monter ces volumes en tant qu'appareils sur vos instances. Les volumes d'Amazon EBS qui sont attachés à une instance sont exposés en tant que volumes de stockage qui sont conservés indépendamment du cycle de vie de l'instance. Vous pouvez créer un système de fichiers au-dessus de ces volumes ou les utiliser comme vous utiliseriez un périphérique de stockage en mode bloc (comme un disque dur). Vous pouvez modifier dynamiquement la configuration d'un volume attaché à une instance. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon EBS.
- Amazon EC2 Auto Scaling permet de vous assurer que vous disposez du bon nombre d'instances EC2 disponibles pour gérer la charge de l'application. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon EC2 Auto Scaling.
- Elastic Load Balancing répartit le trafic réseau ou le trafic applicatif entrant sur plusieurs cibles (par exemple, des instances EC2, des conteneurs et des adresses IP) dans plusieurs zones de disponibilité. Pour plus d'informations, consultez le Guide de l'utilisateur Elastic Load Balancing.
- 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 AWS ressources (authentification), ainsi que pour contrôler les ressources qu'ils peuvent utiliser et comment ils peuvent les utiliser (autorisation). Pour plus d'informations, consultez Authentification et contrôle d'accès pour Amazon CloudWatch.
- AWS Lambdavous permet de créer des applications sans serveur composées de fonctions déclenchées par des événements et de les déployer automatiquement à l'aide de CodePipeline et AWS CodeBuild. Pour plus d'informations, consultez AWS Lambda Applications.
- AWS Launch Wizard for SQL Server réduit le temps nécessaire au déploiement de la solution de haute disponibilité SQL Server dans le cloud. Vous saisissez les exigences de votre application, notamment les performances, le nombre de nœuds et la connectivité sur la console de service, et

vous AWS Launch Wizard identifiez les AWS ressources appropriées pour déployer et exécuter votre application SQL Server Always On.

- AWS Resource Groups vous aide à organiser les ressources qui constituent votre application. L'outil Resource Groups vous permet de gérer et d'automatiser des tâches sur un grand nombre de ressources à la fois. Un seul Resource Group peut être enregistré pour une seule application. Pour plus d'informations, consultez le Guide de l'utilisateur sur les groupes de ressources AWS .
- Amazon SQS offre une file d'attente hébergée sécurisée, durable et disponible qui vous permet d'intégrer et de découpler les systèmes et les composants de logiciels distribués. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon SQS.
- AWS Step Functionsest un compositeur de fonctions sans serveur qui vous permet de séquencer une variété de AWS services et de ressources, y compris des AWS Lambda fonctions, dans des flux de travail visuels structurés. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Step Functions.
- AWS Le SSM OpsCenter agrège et normalise l' OpsItems ensemble des services tout en fournissant des données d'investigation contextuelles sur chacun des services OpsItem, ainsi que sur les ressources connexes et connexes OpsItems. OpsCenter fournit également des documents d'automatisation de Systems Manager (runbooks) que vous pouvez utiliser pour résoudre rapidement les problèmes. Vous pouvez spécifier des données personnalisées consultables pour chacune OpsItem d'entre elles. Vous pouvez également consulter des rapports de synthèse générés automatiquement OpsItems par statut et par source. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Systems Manager .
- Amazon API Gateway est un AWS service de création, de publication, de maintenance, de surveillance et de sécurisation des API REST, HTTP et des WebSocket API à n'importe quelle échelle. Les développeurs d'API peuvent créer des API qui accèdent à AWS d'autres services Web, ainsi qu'à des données stockées dans le AWS cloud. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon API Gateway.

  > **Note**
  >
  > Application Insights prend uniquement en charge les protocoles API REST (v1 du service API Gateway).

- Amazon Elastic Container Service (Amazon ECS) est un service d'orchestration de conteneurs entièrement géré. Vous pouvez utiliser Amazon ECS pour exécuter vos applications les plus sensibles et stratégiques. Pour plus d'informations, consultez le Guide du développeur Amazon Elastic Container Service.

- Amazon Elastic Kubernetes Service (Amazon EKS) est un service géré que vous pouvez utiliser pour exécuter AWS Kubernetes sans avoir à installer, exploiter et gérer votre propre plan de contrôle ou vos propres nœuds Kubernetes. Kubernetes est un système open source destiné à l'automatisation du déploiement, la mise à l'échelle et la gestion d'applications conteneurisées. Pour plus d'informations, consultez le Guide de l'utilisateur Amazon EKS.
- Kubernetes sur Amazon EC2. Kubernetes est un logiciel open source qui vous aide à déployer et à gérer des applications conteneurisées à grande échelle. Kubernetes gère des clusters d'instances de calcul Amazon EC2 et exécute des conteneurs sur ces instances avec des processus de déploiement, de maintenance et de mise à l'échelle. Avec Kubernetes, vous pouvez exécuter n'importe quel type d'application conteneurisée avec le même jeu d'outils sur site et dans le cloud. Pour plus d'informations, consultez Kubernetes Documentation: Getting started.
- Amazon FSx vous aide à lancer et à exécuter des systèmes de fichiers populaires entièrement gérés par AWS. Amazon FSx vous permet de tirer parti des ensembles de fonctions et des performances des systèmes de fichiers open source courants et sous licence commerciale pour éviter des tâches administratives fastidieuses. Pour plus d'informations, consultez la documentation d'Amazon FSx.
- Amazon Simple Notification Service (SNS) est un service de messagerie entièrement géré à la fois pour les communications. application-to-application application-to-person Vous pouvez configurer Amazon SNS pour la surveillance par Application Insights. Lorsque Amazon SNS est configuré comme ressource de surveillance, Application Insights suit les métriques SNS pour déterminer les raisons pour lesquelles les messages SNS pourraient rencontrer des problèmes ou échouer.
- Amazon Elastic File System (Amazon EFS) est un système de fichiers NFS élastique entièrement géré destiné à être utilisé avec des AWS Cloud services et des ressources sur site. Il est conçu pour atteindre des pétaoctets à la demande sans perturber les applications. Il augmente ou diminue automatiquement au fil de vos ajouts et suppressions de fichiers, ce qui élimine le besoin de provisionner et gérer la capacité nécessaire pour répondre à la croissance. Pour en savoir plus, consultez la documentation d'Amazon Elastic File System.

Services tiers associés

- Pour certaines charges de travail et applications surveillées dans Application Insights, l'exportateur JMX Prometheus est installé à l' AWS Systems Manager aide de Distributor afin CloudWatch qu'Application Insights puisse récupérer des métriques spécifiques à Java. Lorsque vous sélectionnez de surveiller une application Java, Application Insights installe automatiquement Prometheus JMX Exporter pour vous.

Composants d'application pris en charge

CloudWatch Application Insights analyse votre groupe de ressources pour identifier les composants de l'application. Les composants peuvent être autonomes, automatiquement regroupés (par exemple, des instances dans un groupe Auto Scaling ou derrière un équilibreur de charge), ou personnalisés (en regroupant des instances Amazon EC2).

Les composants suivants sont pris en charge par CloudWatch Application Insights :

AWS composants

- Amazon EC2
- Amazon EBS
- Amazon RDS
- Elastic Load Balancing : Application Load Balancer et Classic Load Balancer (toutes les instances cibles de ces programmes d'équilibrage de charge sont identifiées et configurées).
- Groupes Amazon EC2 Auto Scaling : AWS Auto Scaling (les groupes Auto Scaling sont configurés dynamiquement pour toutes les instances cibles ; si votre application évolue, CloudWatch Application Insights configure automatiquement les nouvelles instances). Les groupes Auto Scaling ne sont pas pris en charge pour les groupes de ressources CloudFormation basés sur des piles.
- AWS Lambda
- Amazon Simple Queue Service (Amazon SQS)
- Table Amazon DynamoDB
- Métriques de compartiment Amazon S3
- AWS Step Functions
- Étapes d'API REST Amazon API Gateway
- Amazon Elastic Container Service (Amazon ECS) : cluster, service et tâche
- Amazon Elastic Kubernetes Service (Amazon EKS) : cluster
- Kubernetes sur Amazon EC2 : cluster Kubernetes fonctionnant sur EC2
- Rubrique Amazon SNS

Les autres ressources de type de composant ne sont actuellement pas suivies par CloudWatch Application Insights. Si un type de composant qui est pris en charge n'apparaît pas dans votre

solution Application Insights, le composant peut déjà être enregistré et géré par une autre de vos applications, supervisée par Application Insights.

Piles technologiques prises en charge

Vous pouvez utiliser CloudWatch Application Insights pour surveiller vos applications exécutées sur les systèmes d'exploitation Windows Server et Linux en sélectionnant l'option du menu déroulant au niveau de l'application pour l'une des technologies suivantes :

- Front-end : Serveur web Microsoft Internet Information Services (IIS)
- Niveau opérateur :
  - .NET Framework
  - .NET Core
- Applications :
  - Java
  - Déploiements SAP NetWeaver standard, distribués et à haute disponibilité
- Active Directory
- SharePoint
- Bases de données :
  - Microsoft SQL Server s'exécutant sur Amazon RDS ou Amazon EC2 (y compris les configurations haute disponibilité SQL Server. Veuillez consulter Exemples de configuration de composants).
  - MySQL exécuté sur Amazon RDS, Amazon Aurora ou Amazon EC2
  - PostgreSQL exécuté sur Amazon RDS ou Amazon EC2
  - Table Amazon DynamoDB
  - Oracle exécuté sur Amazon RDS ou Amazon EC2
  - Base de données SAP HANA sur une instance Amazon EC2 unique et plusieurs instances EC2
  - Configuration de la haute disponibilité de la base de données SAP HANA Cross-AZ
  - Base de données SAP Sybase ASE sur une seule instance Amazon EC2
  - Configuration de la haute disponibilité de la base de données SAP Sybase ASE Cross-AZ

Si aucune des piles technologiques répertoriées ci-dessus ne s'applique à vos ressources d'application, vous pouvez surveiller votre pile d'applications en sélectionnant Custom (Personnalisé).

dans le menu déroulant du niveau de l'application de la page Manage monitoring (Gérer la surveillance).

Comment fonctionne Amazon CloudWatch Application Insights

Cette section contient des informations sur le fonctionnement CloudWatch d'Application Insights, notamment :

- Comment Application Insights surveille les applications
- Conservation des données
- Quotas
- AWS Packages Systems Manager (SSM) utilisés par CloudWatch Application Insights
- AWS Documents Systems Manager (SSM) utilisés par CloudWatch Application Insights

Comment Application Insights surveille les applications

Application Insights surveille les applications comme suit.

Détection et configuration d'applications

La première fois qu'une application est ajoutée à CloudWatch Application Insights, elle analyse les composants de l'application pour recommander des indicateurs clés, des journaux et d'autres sources de données à surveiller pour votre application. Vous pouvez ensuite configurer votre application en fonction de ces recommandations.

Prétraitement des données

CloudWatch Application Insights analyse en permanence les sources de données surveillées dans les ressources de l'application afin de détecter les anomalies métriques et de consigner les erreurs (observations).

Détection intelligente des problèmes

Le moteur CloudWatch Application Insights détecte les problèmes de votre application en corrélant les observations à l'aide d'algorithmes de classification et de règles intégrées. Pour faciliter le dépannage, il crée des CloudWatch tableaux de bord automatisés, qui incluent des informations contextuelles sur les problèmes.

Alerte et action

Lorsqu' CloudWatch Application Insights détecte un problème avec votre application, il génère CloudWatch des événements pour vous en informer. Pour plus d'informations sur la configuration de ces événements, consultez Application Insights CloudWatch Événements et notifications en cas de problèmes détectés.

Exemple de scénario

Vous avez une application .NET ASP basée sur une base de données SQL Server. Soudain, votre base de données commence à présenter un dysfonctionnement en raison d'une haute sollicitation de la mémoire. Cela conduit à la dégradation des performances de l'application et éventuellement à des erreurs HTTP 500 au niveau de vos serveurs web et de votre équilibreur de charge.

Grâce à CloudWatch Application Insights et à ses analyses intelligentes, vous pouvez identifier la couche applicative à l'origine du problème en consultant le tableau de bord créé dynamiquement qui affiche les métriques associées et les extraits de fichier journal. Dans ce cas, le problème peut se situer sur la couche de base de données SQL.

Conservation des données

CloudWatch Application Insights conserve les problèmes pendant 55 jours et les observations pendant 60 jours.

Quotas

Pour connaître les quotas par défaut pour CloudWatch Application Insights, consultez la section Points de terminaison et quotas Amazon CloudWatch Application Insights. Sauf indication contraire, chaque quota est établi par AWS région. Veuillez contacter Support AWS pour demander une augmentation du quota de votre service. De nombreux services contiennent des quotas qui ne peuvent pas être augmentés. Pour plus d'informations sur les quotas d'un service spécifique, consultez la documentation de ce service.

AWS Packages Systems Manager (SSM) utilisés par CloudWatch Application Insights

Les packages répertoriés dans cette section sont utilisés par Application Insights et peuvent être gérés et déployés indépendamment avec AWS Systems Manager Distributor. Pour plus d'informations sur le distributeur SSM, consultez AWS Distributeur Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.

Packages :

- AWSObservabilityExporter-JMXExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure
- AWSObservabilityExporter-HAClusterExporterInstallAndConfigure
- AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure
- AWSObservabilityExporter-SQLExporterInstallAndConfigure

`AWSObservabilityExporter-JMXExporterInstallAndConfigure`

Vous pouvez récupérer des métriques Java spécifiques à une charge de travail à partir de Prometheus JMX Exporter pour Application Insights pour configurer et surveiller les alertes. Dans la console Application Insights, dans la page Gérer la surveillance, sélectionnez Application JAVA à partir du menu déroulant Niveau Application. Ensuite, sous Configuration de l'exportateur JAVA Prometheus, sélectionnez votre Méthode de collecte et votre Numéro de port JMX.

Pour utiliser AWS Systems Manager Distributor pour empaqueter, installer et configurer le package d'exportation Prometheus JMX AWS fourni indépendamment d'Application Insights, procédez comme suit.

Conditions préalables à l'utilisation du package SSM de Prometheus JMX Exporter

- Agent SSM Agent version 2.3.1550.0 ou ultérieure installée
- La variable d'environnement JAVA_HOME est définie

Installation et configuration du package **`AWSObservabilityExporter-JMXExporterInstallAndConfigure`**

Le package `AWSObservabilityExporter-JMXExporterInstallAndConfigure` est un package Distributeur SSM que vous pouvez utiliser pour installer et configurer Prometheus JMX Exporter. Lorsque des métriques Java sont envoyées par l'exportateur JMX Prometheus, CloudWatch l'agent peut être configuré pour récupérer les métriques du service. CloudWatch

1. Selon vos préférences, préparez le fichier de configuration YAML de l'exportateur JMX Prometheus situé dans le référentiel Prometheus. GitHub Utilisez l'exemple de descriptions de configuration et d'options pour vous guider.
2. Copiez le fichier de configuration YAML de Prometheus JMX Exporter codé en Base64 vers un nouveau paramètre SSM dans le stockage de paramètres SSM.

3. Accédez à la console du Distributeur SSM et ouvrez l'onglet Owned by Amazon (Propriété d'Amazon. Sélectionnez AWSObservabilityExporter-JMX ExporterInstallAndConfigure et choisissez Installer une fois.
4. Mettez à jour le paramètre SSM que vous avez créé dans la première étape en remplaçant « Arguments supplémentaires » par ce qui suit :

```
{
  "SSM_EXPORTER_CONFIGURATION": "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
  "SSM_EXPOSITION_PORT": "9404"
}
```

> **Note**
>
> Le port 9404 est le port par défaut utilisé pour envoyer des métriques Prometheus JMX. Vous pouvez mettre à jour ce port.

Exemple : configurer CloudWatch l'agent pour récupérer les métriques Java

1. Installez le Prometheus JMX Exporter, comme décrit dans la procédure précédente. Vérifiez ensuite qu'il est correctement installé sur votre instance en vérifiant l'état du port.

   Exemple d'installation réussie sur une instance Windows

   ```
   PS C:\> curl http://localhost:9404 (http://localhost:9404/)
   StatusCode : 200
   StatusDescription : OK
   Content : # HELP jvm_info JVM version info
   ```

   Exemple d'installation réussie sur une instance Linux

   ```
   $ curl localhost:9404
   # HELP jmx_config_reload_failure_total Number of times configuration have failed to
    be reloaded.
   # TYPE jmx_config_reload_failure_total counter
   jmx_config_reload_failure_total 0.0
   ```

2. Créez le fichier YAML de découverte de service Prometheus. L'exemple de fichier de découverte de service suivant effectue les opérations suivantes :

- Spécifie le port hôte de Prometheus JMX Exporter comme `localhost: 9404`.
- Attache des étiquettes (`ApplicationComponentName`,, et`InstanceId`) aux métriques, qui peuvent être définies comme des dimensions CloudWatch métriques.

```
$ cat prometheus_sd_jmx.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    Application: myApp
    ComponentName: arn:aws:elasticloadbalancing:us-
east-1:123456789012:loadbalancer/app/sampl-Appli-MMZW8E3GH4H2/aac36d7fea2a6e5b
    InstanceId: i-12345678901234567
```

3. Créez le fichier de configuration YAML de Prometheus JMX Exporter. L'exemple de fichier de configuration suivant spécifie les éléments suivants :

   - Intervalle de travail d'extraction des métriques et délai d'expiration.
   - Les travaux de récupération de métriques (`jmx` et `sap`), qui comprennent le nom de la tâche, la série chronologique maximale renvoyée à la fois et le chemin du fichier de découverte de service.

```
$ cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_jmx.yaml"]
  - job_name: sap
    sample_limit: 10000
    file_sd_configs:
      - files: ["/tmp/prometheus_sd_sap.yaml"]
```

4. Vérifiez que l' CloudWatch agent est installé sur votre instance Amazon EC2 et que la version est 1.247346.1b249759 ou ultérieure. Pour installer l' CloudWatchagent sur votre instance EC2,

consultez la section Installation de l' CloudWatch agent. Pour vérifier la version, consultez la section Recherche d'informations sur les versions des CloudWatch agents.

5. Configurez l' CloudWatch agent. Pour plus d'informations sur la configuration du fichier de configuration de l' CloudWatch agent, voir Création ou modification manuelle du fichier de configuration de l' CloudWatch agent. L'exemple de fichier de configuration d' CloudWatch agent suivant effectue les opérations suivantes :

   - Spécifie le chemin de fichier de configuration YAML de Prometheus JMX Exporter.
   - Spécifie le groupe de journaux cible dans lequel publier les journaux de mesures EMF.
   - Spécifie deux ensembles de dimensions pour chaque nom de métrique.
   - Envoie 8 métriques (4 noms de métriques * 2 ensembles de dimensions par nom de métrique) CloudWatch métriques.

```
{
   "logs":{
      "logs_collected":{
         ....
      },
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-test-cluster",
            "log_group_name":"prometheus-test",
            "prometheus_config_path":"/tmp/prometheus.yaml",
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent",
               "metric_unit":{
                  "jvm_threads_current":"Count",
                  "jvm_gc_collection_seconds_sum":"Second",
                  "jvm_memory_bytes_used":"Bytes"
               },
               "metric_declaration":[
                  {
                     "source_labels":[
                        "job"
                     ],
                     "label_matcher":"^jmx$",
                     "dimensions":[
                        [
                           "InstanceId",
```

```
                        "ComponentName"
                     ],
                     [
                        "ComponentName"
                     ]
                  ],
                  "metric_selectors":[
                     "^java_lang_threading_threadcount$",
                     "^java_lang_memory_heapmemoryusage_used$",
                     "^java_lang_memory_heapmemoryusage_committed$"
                  ]
               }
            ]
         }
      }
   },
   "metrics":{
      ....
   }
}
```

AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure

Vous pouvez récupérer des métriques SAP HANA spécifiques à une charge de travail depuis l'exportateur de base de données Prometheus HANA pour Application Insights pour configurer et surveiller les alertes. Pour plus d'informations, consultez Configurer votre base de données SAP HANA pour la surveillance dans ce guide.

Pour utiliser AWS Systems Manager Distributor pour empaqueter, installer et configurer le package d'exportation de base de données Prometheus HANA AWS fourni indépendamment d'Application Insights, procédez comme suit.

Conditions préalables à l'utilisation du package SSM d'exportateur de base de données Prometheus HANA

- Agent SSM Agent version 2.3.1550.0 ou version ultérieure installée
- Base de données SAP HANA
- Système d'exploitation Linux (SUSE Linux, RedHat Linux)

- Un secret avec les informations d'identification de surveillance de base de données SAP HANA AWS Secrets Manager. Créez un secret en utilisant le format de paires clé/valeur, spécifiez le nom d'utilisateur de la clé et saisissez l'utilisateur de la base de données correspondant à la valeur. Ajoutez un second mot de passe, puis pour Value (Valeur), saisissez le mot de passe. Pour plus d'informations sur la création de secrets, consultez la rubrique Création d'un secret dans le Guide de l'utilisateur AWS Secrets Manager . Le secret doit présenter le format suivant :

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

Installation et configuration du package **AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure**

Le package `AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure` est un package Distributeur SSM que vous pouvez utiliser pour installer et configurer l'exportateur de base de données Prometheus HANA. Lorsque les métriques de base de données HANA sont envoyées par l'exportateur de base de données Prometheus HANA, CloudWatch l'agent peut être configuré pour récupérer les métriques du service. CloudWatch

1. Création d'un paramètre SSM dans le stockage de paramètres SSM pour stocker les configurations de l'exportateur. Voici un exemple de valeur de paramètre.

```
{\"exposition_port\":9668,\"multi_tenant\":true,\"timeout\":600,\"hana\":{\"host\":
\"localhost\",\"port\":30013,\"aws_secret_name\":\"HANA_DB_CREDS\",\"scale_out_mode
\":true}}
```

> **Note**
>
> Dans cet exemple, l'exportation s'exécute uniquement sur l'instance Amazon EC2 à l'aide de l'option active `SYSTEM` et elle restera inactive sur les autres instances EC2 afin d'éviter les doublons de métriques. L'exportateur peut extraire toutes les informations du locataire de la base de données à partir de `SYSTEM` la base de données.

2. Créer un paramètre SSM dans Stockage de paramètres SSM pour stocker les requêtes de métriques de l'exportateur. Le package peut accepter plusieurs paramètres de métriques. Chaque paramètre doit avoir un format d'objet JSON valide. Voici un exemple de valeur de paramètre :

```
{\"SELECT MAX(TIMESTAMP) TIMESTAMP, HOST, MEASURED_ELEMENT_NAME CORE,
 SUM(MAP(CAPTION, 'User Time', TO_NUMBER(VALUE), 0)) USER_PCT, SUM(MAP(CAPTION,
 'System Time', TO_NUMBER(VALUE), 0)) SYSTEM_PCT, SUM(MAP(CAPTION, 'Wait
 Time', TO_NUMBER(VALUE), 0)) WAITIO_PCT, SUM(MAP(CAPTION, 'Idle Time', 0,
 TO_NUMBER(VALUE))) BUSY_PCT, SUM(MAP(CAPTION, 'Idle Time', TO_NUMBER(VALUE), 0))
 IDLE_PCT FROM sys.M_HOST_AGENT_METRICS WHERE MEASURED_ELEMENT_TYPE = 'Processor'
 GROUP BY HOST, MEASURED_ELEMENT_NAME;\":{\"enabled\":true,\"metrics\":[{\"name\":
\"hanadb_cpu_user\",\"description\":\"Percentage of CPU time spent by HANA DB in user
 space, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"USER_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_system
\",\"description\":\"Percentage of CPU time spent by HANA DB in Kernel space,
 over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":
\"SYSTEM_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_waitio
\",\"description\":\"Percentage of CPU time spent by HANA DB in IO mode, over the
 last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE\"],\"value\":\"WAITIO_PCT\",
\"unit\":\"percent\",\"type\":\"gauge\"},{\"name\":\"hanadb_cpu_busy\",\"description
\":\"Percentage of CPU time spent by HANA DB, over the last minute (in seconds)\",
\"labels\":[\"HOST\",\"CORE\"],\"value\":\"BUSY_PCT\",\"unit\":\"percent\",\"type\":
\"gauge\"},{\"name\":\"hanadb_cpu_idle\",\"description\":\"Percentage of CPU time not
 spent by HANA DB, over the last minute (in seconds)\",\"labels\":[\"HOST\",\"CORE
\"],\"value\":\"IDLE_PCT\",\"unit\":\"percent\",\"type\":\"gauge\"}]}}
```

Pour plus d'informations sur les requêtes de métriques, consultez le SUSE / hanadb_exporterdépôt sur GitHub.

3. Accédez à la console du Distributeur SSM et ouvrez l'onglet Owned by Amazon (Propriété d'Amazon. Sélectionnez AWSObservabilityExporterExporterInstallAndConfigure-SAP-HANADB* et choisissez Installer une fois.
4. Mettez à jour le paramètre SSM que vous avez créé dans la première étape en remplaçant « Arguments supplémentaires » par ce qui suit :

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_EXPORTER_METRICS_1": "{{ssm:<SSM_FIRST_METRICS_PARAMETER_STORE_NAME>}}",
  "SSM_EXPORTER_METRICS_2": "{{ssm:<SSM_SECOND_METRICS_PARAMETER_STORE_NAME>}}"
}
```

5. Sélectionnez les instances Amazon EC2 avec base de données SAP HANA, puis sélectionnez Run (Exécuter).

`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`

Vous pouvez récupérer des métriques de cluster haute disponibilité (HA) spécifiques à la charge de travail à partir de l'exportateur de cluster Prometheus HANA pour Application Insights afin de configurer et de surveiller les alertes pour une configuration de haute disponibilité de base de données SAP HANA. Pour plus d'informations, consultez Configurer votre base de données SAP HANA pour la surveillance dans ce guide.

Pour utiliser AWS Systems Manager Distributor pour empaqueter, installer et configurer le package d'exportation de clusters Prometheus HA AWS fourni indépendamment d'Application Insights, procédez comme suit.

Conditions préalables à l'utilisation du package SSM de l'exportateur de cluster Prometheus HA

- Agent SSM Agent version 2.3.1550.0 ou version ultérieure installée
- Cluster HA pour Pacemaker, Corosync, SBD et DRBD
- Système d'exploitation Linux (SUSE Linux, RedHat Linux)

Installation et configuration du package `AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`

Le package `AWSObservabilityExporter-HAClusterExporterInstallAndConfigure` est un package de distributeur SSM que vous pouvez utiliser pour installer et configurer l'exportateur de cluster Prometheus HA. Lorsque les métriques du cluster sont envoyées par l'exportateur de base de données Prometheus HANA, CloudWatch l'agent peut être configuré pour récupérer les métriques du service. CloudWatch

1. Créez un paramètre SSM dans le stockage de paramètres SSM pour stocker les configurations de l'exportateur au format JSON. Voici un exemple de valeur de paramètre.

   ```
   {\"port\":\"9664\",\"address\":\"0.0.0.0\",\"log-level\":\"info\",\"crm-mon-path
   \":\"/usr/sbin/crm_mon\",\"cibadmin-path\":\"/usr/sbin/cibadmin\",\"corosync-
   cfgtoolpath-path\":\"/usr/sbin/corosync-cfgtool\",\"corosync-quorumtool-path\":\"/
   usr/sbin/corosync-quorumtool\",\"sbd-path\":\"/usr/sbin/sbd\",\"sbd-config-path\":
   \"/etc/sysconfig/sbd\",\"drbdsetup-path\":\"/sbin/drbdsetup\",\"enable-timestamps
   \":false}
   ```

   Pour plus d'informations sur les configurations de l'exportateur, consultez le `ClusterLabs / ha_cluster_exporter`dépôt sur GitHub.

2. Accédez à la console du Distributeur SSM et ouvrez l'onglet Owned by Amazon (Propriété d'Amazon. Sélectionnez AWSObservabilityExporter-HA ClusterExporterInstallAndConfigure * et choisissez Installer une fois.
3. Mettez à jour le paramètre SSM que vous avez créé dans la première étape en remplaçant « Arguments supplémentaires » par ce qui suit :

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<*SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>*}}"
}
```

4. Sélectionnez les instances Amazon EC2 avec base de données SAP HANA, puis sélectionnez Run (Exécuter).

`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`

Vous pouvez récupérer des métriques SAP spécifiques à la charge de travail NetWeaver depuis Prometheus SAP Host Exporter for Application Insights afin de configurer et de surveiller les alarmes pour les déploiements SAP Distributed et High Availability. NetWeaver Pour plus d'informations, consultez Commencez avec Amazon CloudWatch Application Insights.

Pour utiliser le Distributeur AWS Systems Manager pour packager, installer et configurer le package exportateur d'hôte SAP indépendamment d'Application Insights, effectuez les étapes suivantes.

Conditions préalables à l'utilisation du package SSM de l'exportateur d'hôte SAP de Prometheus

- Agent SSM Agent version 2.3.1550.0 ou version ultérieure installée
- Serveurs NetWeaver d'applications SAP
- Système d'exploitation Linux (SUSE Linux, RedHat Linux)

Installation et configuration du package **`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`**

Le `AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure` package est un package SSM Distributor que vous pouvez utiliser pour installer et configurer l'exportateur de métriques SAP NetWeaver Prometheus. Lorsque NetWeaver les métriques SAP sont envoyées par l'exportateur Prometheus, CloudWatch l'agent peut être configuré pour récupérer les métriques du service. CloudWatch

1. Créez un paramètre SSM dans le stockage de paramètres SSM pour stocker les configurations de l'exportateur au format JSON. Voici un exemple de valeur de paramètre.

```
{\"address\":\"0.0.0.0\",\"port\":\"9680\",\"log-level\":\"info\",\"is-HA\":false}
```

   - address

     L'adresse cible à laquelle envoyer les métriques Prometheus. La valeur par défaut est `localhost`.

   - port

     Le port cible vers lequel envoyer les métriques Prometheus. La valeur par défaut est `9680`.

   - is-HA

     `true`pour les déploiements SAP NetWeaver High Availability. Pour tous les autres déploiements, la valeur est `false`.

2. Accédez à la console du Distributeur SSM et ouvrez l'onglet Owned by Amazon (Propriété d'Amazon. Sélectionnez AWSObservabilityExporter-SAP-SAP HostExporterInstallAndConfigure et choisissez Installer une fois.

3. Mettez à jour le paramètre SSM que vous avez créé dans la première étape en remplaçant « Arguments supplémentaires » par ce qui suit :

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:<SSM_CONFIGURATIONS_PARAMETER_STORE_NAME>}}",
  "SSM_SID": "<SAP_DATABASE_SID>",
  "SSM_INSTANCES_NUM": "<instances_number seperated by comma>"
}
```

   Exemple

```
{
  "SSM_EXPORTER_CONFIG": "{{ssm:exporter_config_paramter}}",
  "SSM_INSTANCES_NUM": "11,12,10",
  "SSM_SID": "PR1"
}
```

4. Sélectionnez les instances Amazon EC2 avec les NetWeaver applications SAP, puis choisissez Exécuter.

> **Note**
>
> L'exportateur Prometheus gère les métriques NetWeaver SAP sur un point de terminaison local. Seuls les utilisateurs du système d'exploitation de l'instance Amazon EC2 peuvent accéder au point de terminaison local. Par conséquent, après l'installation du package de l'exportateur, les métriques sont disponibles pour tous les utilisateurs du système d'exploitation. Le point de terminaison local par défaut est `localhost:9680/metrics`.

`AWSObservabilityExporter-SQLExporterInstallAndConfigure`

Vous pouvez récupérer des métriques SQL Server spécifiques à une charge de travail à partir de Prometheus SQL Exporter pour Application Insights pour surveiller des métriques clés.

Pour utiliser le Distributeur AWS Systems Manager pour packager, installer et configurer le package exportateur de SQL indépendamment d'Application Insights, effectuez les étapes suivantes.

Conditions préalables à l'utilisation du package SSM de Prometheus SQL Exporter

- Agent SSM Agent version 2.3.1550.0 ou version ultérieure installée
- Instance Amazon EC2 exécutant SQL Server sous Windows avec authentification utilisateur SQL Server activée.
- Un utilisateur de SQL Server disposant des autorisations suivantes :

```
GRANT VIEW ANY DEFINITION TO
```

```
GRANT VIEW SERVER STATE TO
```

- Un secret contenant la chaîne de connexion à la base de données utilisant AWS Secrets Manager. Pour plus d'informations sur la création de secrets, consultez la rubrique Création d'un secret dans le Guide de l'utilisateur AWS Secrets Manager . Le secret doit présenter le format suivant :

```
{
  "data_source_name":"sqlserver://<username>:<password>@localhost:1433"
}
```

> **Note**
>
> Si le mot de passe ou le nom d'utilisateur contient des caractères spéciaux, vous devez les encoder en pourcentage pour garantir une connexion réussie à la base de données.

Installation et configuration du package **AWSObservabilityExporter-SQLExporterInstallAndConfigure**

Le package `AWSObservabilityExporter-SQLExporterInstallAndConfigure` est un package Distributeur SSM que vous pouvez utiliser pour installer et configurer l'exportateur de métriques SQL Prometheus. Lorsque les métriques sont envoyées par l'exportateur Prometheus, CloudWatch l'agent peut être configuré pour récupérer les métriques du service. CloudWatch

1. En fonction de vos préférences, préparez la configuration YAML de SQL Exporter. L'exemple de configuration suivant comporte une seule métrique configurée. Utilisez l'exemple de configuration pour mettre à jour la configuration avec des métriques supplémentaires ou pour créer votre propre configuration.

```
---
global:
  scrape_timeout_offset: 500ms
  min_interval: 0s
  max_connections: 3
  max_idle_connections: 3
target:
  aws_secret_name: <SECRET_NAME>
  collectors:
    - mssql_standard
collectors:
  - collector_name: mssql_standard
    metrics:
    - metric_name: mssql_batch_requests
      type: counter
      help: 'Number of command batches received.'
      values: [cntr_value]
      query: |
        SELECT cntr_value
        FROM sys.dm_os_performance_counters WITH (NOLOCK)
        WHERE counter_name = 'Batch Requests/sec'
```

2. Copiez le fichier de configuration YAML de Prometheus SQL Exporter codé en Base64 vers un nouveau paramètre SSM dans le stockage de paramètres SSM.
3. Accédez à la console du Distributeur SSM et ouvrez l'onglet Owned by Amazon (Propriété d'Amazon. Sélectionnez AWSObservabilityExporter-SQL, ExporterInstallAndConfigure puis choisissez Installer une fois.
4. Remplacez les « arguments supplémentaires » par les informations suivantes. `SSM_PARAMETER_NAME` est le nom du paramètre que vous avez créé à l'étape 2.

```
{
  "SSM_EXPORTER_CONFIGURATION":
    "{{ssm:<SSM_PARAMETER_STORE_NAME>}}",
    "SSM_PROMETHEUS_PORT": "9399",
    "SSM_WORKLOAD_NAME": "SQL"
}
```

5. Sélectionnez l'instance Amazon EC2 avec la base de données SQL Server, puis choisissez Exécuter.

AWS Documents Systems Manager (SSM) utilisés par CloudWatch Application Insights

Application Insights utilise les documents SSM répertoriés dans cette section pour définir les actions qu' AWS Systems Manager effectue sur vos instances gérées. Ces documents utilisent les fonctionnalités `Run Command` de Systems Manager pour automatiser les tâches nécessaires à la mise en œuvre des fonctionnalités de surveillance d'Application Insights. Les plannings d'exécution de ces documents sont gérés par Application Insights et ne peuvent pas être modifiés.

Pour plus d'informations sur les documents SSM, veuillez consulter la rubrique Documents AWS Systems Manager dans le Guide de l'utilisateur AWS Systems Manager .

Documents gérés par CloudWatch Application Insights

Le tableau suivant répertorie les documents SSM gérés par Application Insights.

Nom du document	Description	Calendrier d’exécution
`AWSEC2-DetectWorkload`	Détecte automatiquement les applications exécutées dans votre environnement d’application pouvant être configurées pour être surveillées par Application Insights.	Ce document est diffusé toutes les heures dans votre environnement d'application pour obtenir des informations détaillées sur up-to-date l'application.
`AWSEC2-CheckPerformanceCounterSets`	Vérifie si les espaces de noms Performance Counter sont activés sur vos instances Windows Amazon EC2.	Ce document s’exécute toutes les heures dans votre environnement d’application et ne surveille les métriques de Performance Counter que si les espaces de noms correspondants sont activés.
`AWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure`	Installe et configure l' CloudWatch agent en fonction de la configuration de surveillance des composants de votre application.	Ce document est exécuté toutes les 30 minutes pour garantir que la configuration de l' CloudWatch agent est toujours précise et up-to-date. Le document s’exécute également immédiatement après une modification apportée à votre configuration de surveillance des applications, telle que l’ajout ou la suppression de métriques ou la mise à jour des configurations des journaux.

Documents gérés par AWS Systems Manager

Les documents suivants sont utilisés par CloudWatch Application Insights et gérés par Systems Manager.

AWS-ConfigureAWSPackage

Application Insights utilise ce document pour installer et désinstaller les packages de distribution Prometheus Exporter, pour collecter des métriques spécifiques à la charge de travail et pour permettre une surveillance complète des charges de travail sur les instances Amazon EC2 du client. CloudWatch Application Insights installe les packages de distribution Prometheus Exporter uniquement si la charge de travail cible corrélée est exécutée sur votre instance.

Le tableau suivant répertorie les packages du distributeur Prometheus Exporter et les charges de travail cibles corrélées.

Nom du package du distributeur Prometheus Exporter	Charge de travail cible
`AWSObservabilityExporter-HAClusterExporterInstallAndConfigure`	SAP HANA HA
`AWSObservabilityExporter-JMXExporterInstallAndConfigure`	Java/JMX
`AWSObservabilityExporter-SAP-HANADBExporterInstallAndConfigure`	SAP HANA
`AWSObservabilityExporter-SAP-SAPHostExporterInstallAndConfigure`	NetWeaver
`AWSObservabilityExporter-SQLExporterInstallAndConfigure`	SQL Server (Windows) et SAP ASE (Linux)

AmazonCloudWatch-ManageAgent

Application Insights utilise ce document pour gérer le statut et la configuration de l' CloudWatch agent sur vos instances et pour collecter des métriques internes au niveau du système et des journaux à partir des instances Amazon EC2 sur tous les systèmes d'exploitation.

Commencez avec Amazon CloudWatch Application Insights

Pour commencer à utiliser CloudWatch Application Insights, vérifiez que vous remplissez les conditions préalables suivantes et que vous avez créé une politique IAM. Vous pouvez ensuite commencer à utiliser le lien de la console pour activer CloudWatch Application Insights. Pour configurer vos ressources d'application, suivez les étapes indiquées à la rubrique Configurer et gérer votre application pour la surveillance.

Table des matières

Accédez à des informations sur les CloudWatch applications

Vous pouvez accéder à CloudWatch Application Insights et le gérer via l'une des interfaces suivantes :

- CloudWatch console. Pour ajouter des moniteurs à votre application, choisissez Application Insights sous Insights dans le volet de navigation gauche de la CloudWatch console. Une fois votre application configurée, vous pouvez utiliser la CloudWatch console pour afficher et analyser les problèmes détectés.
- AWS Interface de ligne de commande (AWS CLI). Vous pouvez utiliser le AWS CLI pour accéder aux opérations de AWS l'API. Pour plus d'informations, consultez la section Installation de l'interface de ligne de AWS commande dans le guide de l'utilisateur de l'interface de ligne de AWS commande. Pour obtenir des informations sur l'API Application Insights, consultez le manuel Amazon CloudWatch Application Insights API Reference.

Prérequis

Vous devez remplir les conditions préalables suivantes pour configurer une application avec CloudWatch Application Insights :

- AWS Systems Manager activation — Installez l'agent Systems Manager (agent SSM) sur vos instances Amazon EC2 et activez les instances pour SSM. Pour plus d'informations sur l'installation de l'Agent SSM, veuillez consulter la rubrique Configuration de AWS Systems Manager dans le Guide de l'utilisateur AWS Systems Manager .
- Rôle d'instance EC2 : vous devez attacher les rôles d'instance Amazon EC2 suivants pour activer Systems Manager.
  - Vous devez joindre le rôle `AmazonSSMManagedInstanceCore` pour activer Systems Manager. Pour plus d'informations, consultez AWS Systems Manager Exemples de politiques basées sur l'identité.
  - Vous devez joindre la `CloudWatchAgentServerPolicy` politique pour permettre l'émission des métriques et des journaux de l'instance CloudWatch. Pour plus d'informations, voir Création de rôles et d'utilisateurs IAM à utiliser avec l' CloudWatch agent.
- AWS groupes de ressources : pour intégrer vos applications à CloudWatch Application Insights, créez un groupe de ressources qui inclut toutes les AWS ressources associées utilisées par votre pile d'applications. Cela inclut les équilibreurs de charge d'application, les instances Amazon EC2 exécutant IIS et le web frontal, les niveaux d'exécuteurs .NET et les bases de données SQL Server. Pour plus d'informations sur les composants d'application et les piles technologiques pris en charge par Application Insights, consultezComposants d'application pris en charge. CloudWatch Application Insights inclut automatiquement les groupes Auto Scaling utilisant les mêmes balises ou CloudFormation piles que votre groupe de ressources, car les groupes Auto Scaling ne sont pas pris en charge par les groupes de CloudFormation ressources. Pour plus d'informations, consultez Premiers pas avec les Resource Groups AWS.
- Autorisations IAM : pour les utilisateurs qui n'ont pas d'accès administratif, vous devez créer une politique AWS Identity and Access Management (IAM) qui permet à Application Insights de créer un rôle lié à un service et de l'associer à l'identité de l'utilisateur. Pour plus d'informations sur la création de la politique IAM, veuillez consulter Politique IAM.
- Rôle lié à un service — Application Insights utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est créé pour vous lorsque vous créez votre première application Application Insights dans la console Application Insights. Pour plus d'informations, consultez Utilisation de rôles liés à un service pour Application Insights CloudWatch .
- Prise en charge des métriques de compteur de performances pour les instances EC2 Windows : pour surveiller les métriques de compteur de performances sur vos instances Amazon EC2 Windows, les compteurs de performances doivent être installés sur les instances. Pour les métriques de compteur de performances et les noms d'ensemble de compteurs de performances

correspondants, consultez Métriques de compteur de performances. Pour plus d'informations sur les compteurs de performances, consultez Compteurs de performances.

- Amazon CloudWatch agent — Application Insights installe et configure l' CloudWatch agent. Si CloudWatch l'agent est installé, Application Insights conserve votre configuration. Pour éviter un conflit de fusion, supprimez la configuration des ressources que vous souhaitez utiliser dans Application Insights du fichier de configuration de l' CloudWatch agent existant. Pour plus d'informations, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

Politique IAM

Pour utiliser CloudWatch Application Insights, vous devez créer une politique AWS Identity and Access Management (IAM) et l'associer à votre utilisateur, groupe ou rôle. Pour de plus amples informations sur les utilisateurs, les groupes et les rôles, consultez Identités IAM (utilisateurs, groupes d'utilisateurs et rôles). La politique IAM définit les autorisations des utilisateurs.

Pour créer une politique IAM à l'aide de la console

Pour créer une politique IAM à l'aide de la console IAM, réalisez les étapes suivantes.

1. Accédez à la console IAM. Dans le panneau de navigation, sélectionnez Policies (politiques).
2. En haut de la page, sélectionnez Create policy (Créer une politique).
3. Sélectionnez l'onglet JSON.
4. Copiez et collez le document JSON suivant sous l'onglet JSON.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "applicationinsights:*",
                "iam:CreateServiceLinkedRole",
                "iam:ListRoles",
                "resource-groups:ListGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
```

```
}
```

5. Sélectionnez Review policy (Examiner une politique).
6. Entrez un nom pour la politique, par exemple « »AppInsightsPolicy. Saisissez une Description (en option).
7. Sélectionnez Create Policy (Créer une politique).
8. Dans le panneau de navigation, sélectionnez Groupes d'utilisateurs, Utilisateurs ou Rôles.
9. Sélectionnez le nom correspondant au groupe d'utilisateurs, à l'utilisateur ou au rôle auquel vous souhaitez attacher la politique.
10. Sélectionnez Add permissions (Ajouter des autorisations).
11. Sélectionnez Attach existing policies directly (Attacher directement les politiques existantes).
12. Recherchez la politique que vous venez de créer, puis sélectionnez la case à cocher à gauche du nom de la politique.
13. Ensuite, cliquez sur Next: Review (Suivant : Vérification).
14. Assurez-vous que la bonne politique est répertoriée, puis sélectionnez Add permissions (Ajouter des autorisations).
15. Assurez-vous de vous connecter avec l'utilisateur associé à la politique que vous venez de créer lorsque vous utilisez CloudWatch Application Insights.

Pour créer une politique IAM à l'aide du AWS CLI

Pour créer une politique IAM à l'aide de AWS CLI, exécutez l'opération create-policy depuis la ligne de commande en utilisant le document JSON ci-dessus sous forme de fichier dans votre dossier actuel.

Pour créer une politique IAM à l'aide de AWS Tools for Windows PowerShell

Pour créer une politique IAM à l'aide de AWS Tools for Windows PowerShell, exécutez l'applet de commande New-IAMPolicy en utilisant le document JSON ci-dessus sous forme de fichier dans votre dossier actuel.

Autorisations de rôle IAM pour l'onboarding des applications basées sur un compte.

Si vous souhaitez intégrer toutes les ressources de votre compte, et que vous choisissez de ne pas utiliser la politique gérée Application Insights pour accéder pleinement à la fonction Application Insights, vous devez attacher les autorisations suivantes à votre rôle IAM afin que Application Insights puisse découvrir toutes les ressources de votre compte :

```
"ec2:DescribeInstances"
"ec2:DescribeNatGateways"
"ec2:DescribeVolumes"
"ec2:DescribeVPCs"
"rds:DescribeDBInstances"
"rds:DescribeDBClusters"
"sqs:ListQueues"
"elasticloadbalancing:DescribeLoadBalancers"
"autoscaling:DescribeAutoScalingGroups"
"lambda:ListFunctions"
"dynamodb:ListTables"
"s3:ListAllMyBuckets"
"sns:ListTopics"
"states:ListStateMachines"
"apigateway:GET"
"ecs:ListClusters"
"ecs:DescribeTaskDefinition"
"ecs:ListServices"
"ecs:ListTasks"
"eks:ListClusters"
"eks:ListNodegroups"
"fsx:DescribeFileSystems"
"route53:ListHealthChecks"
"route53:ListHostedZones"
"route53:ListQueryLoggingConfigs"
"route53resolver:ListFirewallRuleGroups"
"route53resolver:ListFirewallRuleGroupAssociations"
"route53resolver:ListResolverEndpoints"
"route53resolver:ListResolverQueryLogConfigs"
"route53resolver:ListResolverQueryLogConfigAssociations"
"logs:DescribeLogGroups"
"resource-explorer:ListResources"
```

Configurer et gérer votre application pour la surveillance

Cette section décrit les étapes à suivre pour configurer, configurer et gérer votre CloudWatch application Application Insights à l'aide de la console AWS CLI, du et AWS Tools for Windows PowerShell.

Rubriques

- Configurez, configurez et gérez votre application à des fins de surveillance depuis la CloudWatch console

- Configurer et gérer votre application pour la surveillance à l'aide de la ligne de commande
- Application Insights CloudWatch Événements et notifications en cas de problèmes détectés

Configurez, configurez et gérez votre application à des fins de surveillance depuis la CloudWatch console

Cette section décrit les étapes à suivre pour configurer, configurer et gérer votre application à des fins de surveillance depuis la CloudWatch console.

Procédures de la console

- Ajout et configuration d'une application
- Activer la surveillance des ressources Application Insights pour Amazon ECS et Amazon EKS
- Désactiver la surveillance d'un composant d'application
- Supprimer une application

Ajout et configuration d'une application

Ajouter et configurer une application depuis la CloudWatch console

Pour commencer à utiliser CloudWatch Application Insights depuis la CloudWatch console, effectuez les étapes suivantes.

1. Démarrer. Ouvrez la page d'accueil de la CloudWatch console. Dans le panneau de navigation de gauche, sélectionnez Application Insights sous Insights. La page qui s'ouvre affiche la liste des applications surveillées avec CloudWatch Application Insights, ainsi que leur état de surveillance.
2. Ajouter une application. Pour configurer la surveillance de votre application, sélectionnez Add an application (Ajouter une application. Lorsque vous sélectionnez Ajouter une application, vous êtes invité à Choisir un type d'application.

    - Application basée sur un Resource Group. Lorsque vous sélectionnez cette option, vous pouvez choisir les Resource Groups de ce compte à surveiller. Pour utiliser plusieurs applications sur un composant, vous devez utiliser la surveillance basée sur les groupes de ressources.
    - Application basée sur un compte. Lorsque vous sélectionnez cette option, vous pouvez surveiller toutes les ressources de ce compte. Si vous souhaitez surveiller toutes les

ressources d'un compte, nous recommandons cette option plutôt que celle basée sur les groupes de ressources, car le processus d'onboarding de l'application est plus rapide.

> **Note**
>
> Vous ne pouvez pas combiner la surveillance basée sur des groupes de ressources avec la surveillance basée sur un compte à l'aide d'Application Insights. Pour modifier le type d'application, vous devez supprimer toutes les applications surveillées, et Choisir le type d'application.

Lorsque vous ajoutez votre première application de surveillance, CloudWatch Application Insights crée un rôle lié à un service dans votre compte, qui autorise Application Insights à appeler d'autres AWS services en votre nom. Pour plus d'informations sur le rôle lié à un service créé dans votre compte par Application Insights, consultez la section Utilisation de rôles liés à un service pour Application Insights CloudWatch .

3. Resource-based application monitoring

    1. Sélectionner un Resource Group. Sur la page Spécifier les détails de l'application, sélectionnez le groupe de AWS ressources contenant les ressources de votre application dans la liste déroulante. Ces ressources incluent des serveurs frontaux, des équilibreurs de charge, des groupes Auto Scaling et des serveurs de base de données.

        Si vous n'avez pas créé de Resource Group pour votre application, vous pouvez le faire en sélectionnant Create new resource group (Créer un nouveau Resource Group). Pour plus d'informations sur Resource Groups, consultez le AWS Resource Groups Guide de l'utilisateur.

    2. Surveillez CloudWatch les événements. Cochez la case pour intégrer la surveillance des applications aux CloudWatch événements afin d'obtenir des informations provenant d'Amazon EBS, d'Amazon EC2 AWS CodeDeploy, d'Amazon ECS, des API et des notifications AWS Health , d'Amazon RDS, d'Amazon S3 et. AWS Step Functions

    3. Intégrez avec AWS Systems Manager OpsCenter. Pour consulter les applications sélectionnées et être averti lorsque des problèmes sont détectés, cochez la case Generate Systems Manager OpsCenter OpsItems pour les actions correctives. Pour suivre les opérations effectuées pour résoudre les éléments de travail opérationnels (OpsItems) liés à vos AWS ressources, fournissez l'ARN de la rubrique SNS.

4. Balises : facultatif. CloudWatch Application Insights prend en charge les groupes de ressources basés CloudFormation sur des balises et des groupes de ressources (à l'exception des groupes Auto Scaling). Pour plus d'informations, consultez Utilisation de Tag Editor.
5. Sélectionnez Suivant.

   Un ARN sera généré pour l'application au format suivant.

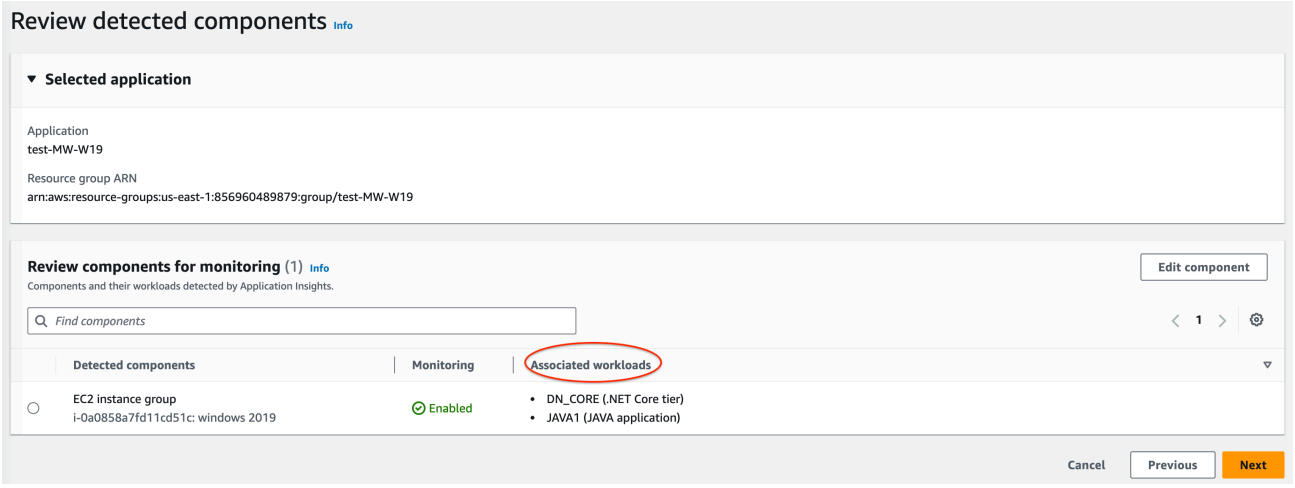
   ```
   arn:partition:applicationinsights:region:account-id:application/resource-group/resource-group-name
   ```

   Exemple

   ```
   arn:aws:applicationinsights:us-east-1:123456789012:application/resource-group/my-resource-group
   ```

6. Dans la page Examiner les composants détectés, sous Vérifier les composants à des fins de surveillance, la table répertorie les composants détectés et les charges de travail détectées associées.

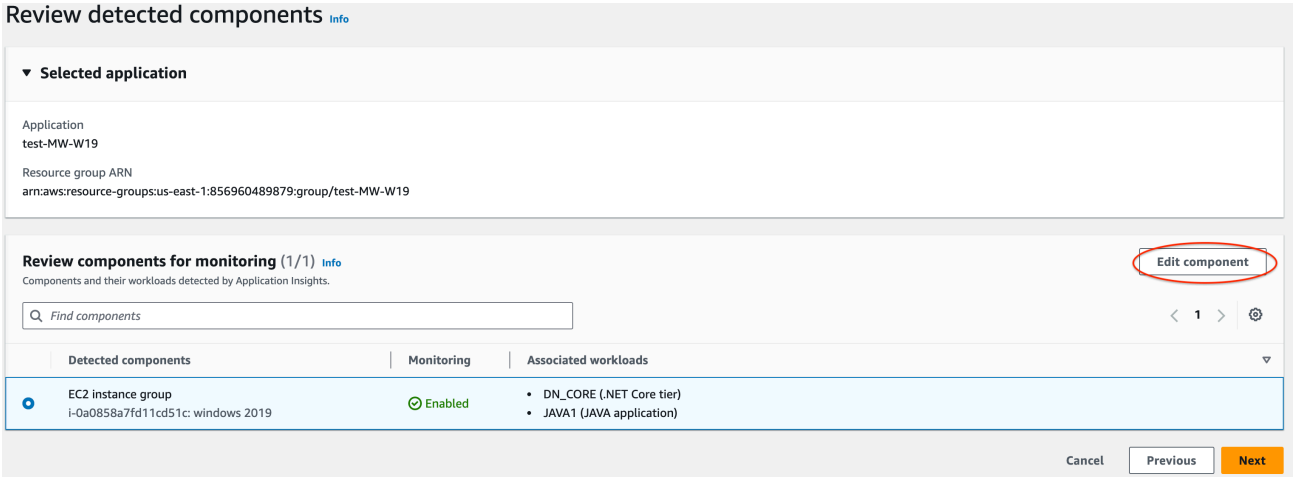
   > **Note**
   >
   > Pour les composants prenant en charge plusieurs charges de travail personnalisées, vous pouvez surveiller jusqu'à cinq charges de travail pour chaque composant. Ces charges de travail seront surveillées séparément du composant.



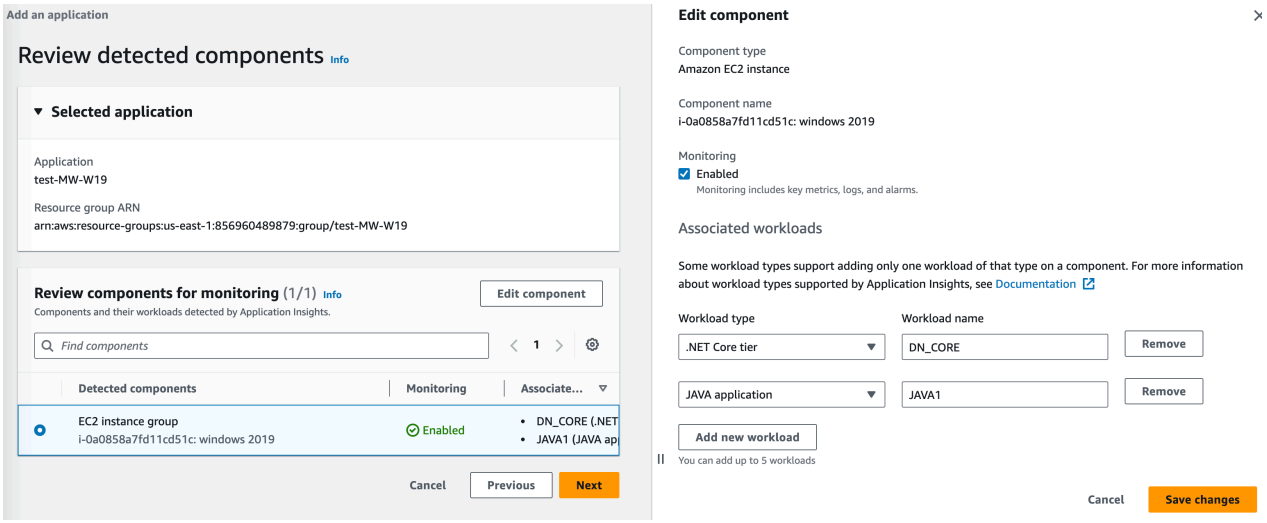
Sous Charges de travail associées, plusieurs messages peuvent s'afficher si une charge de travail n'est pas répertoriée.

- Impossible de détecter les charges de travail : un problème s'est produit lors de la tentative de détection des charges de travail. Assurez-vous d'avoir réalisé les étapes de la rubrique Prérequis. Si vous devez ajouter des charges de travail, choisissez Modifier le composant.
- Aucune charge de travail détectée : nous n'avons détecté aucune charge de travail. Vous devez peut-être ajouter des charges de travail. Pour ce faire, choisissez Modifier le composant.
- Non applicable : le composant ne prend pas en charge les charges de travail personnalisées et sera surveillé à l'aide de métriques, d'alertes et de journaux par défaut. Vous ne pouvez pas ajouter de charges de travail à ces composants.

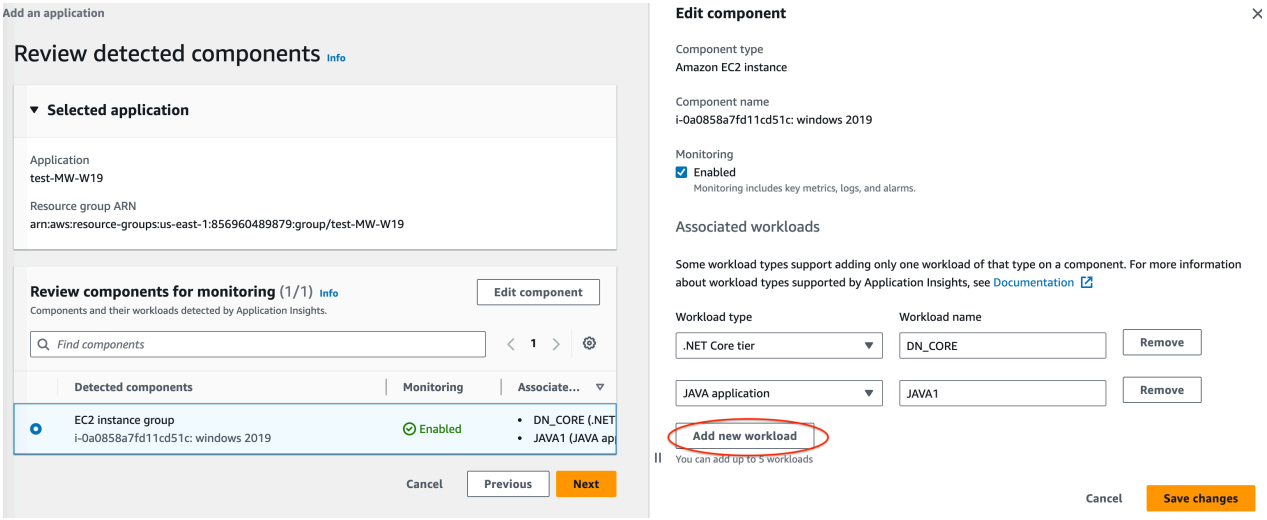
7. Pour modifier un composant, sélectionnez-le, puis choisissez Modifier le composant. Un panneau latéral s'ouvre avec les charges de travail détectées sur le composant. Dans ce panneau, vous pouvez modifier les détails des composants et ajouter des charges de travail.



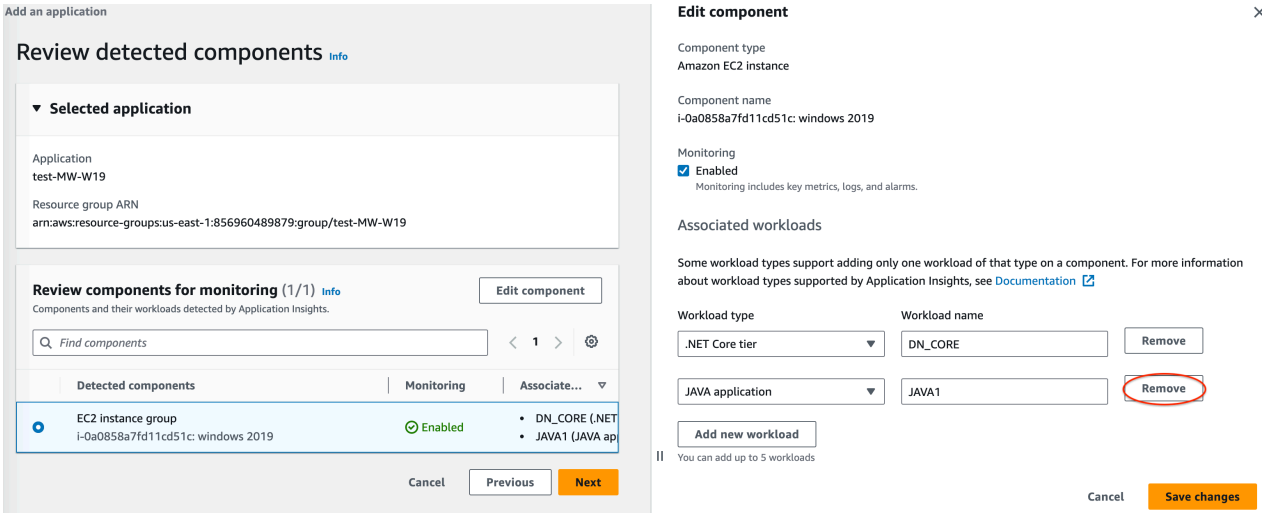
- Pour modifier le type ou le nom de la charge de travail, utilisez le menu déroulant.



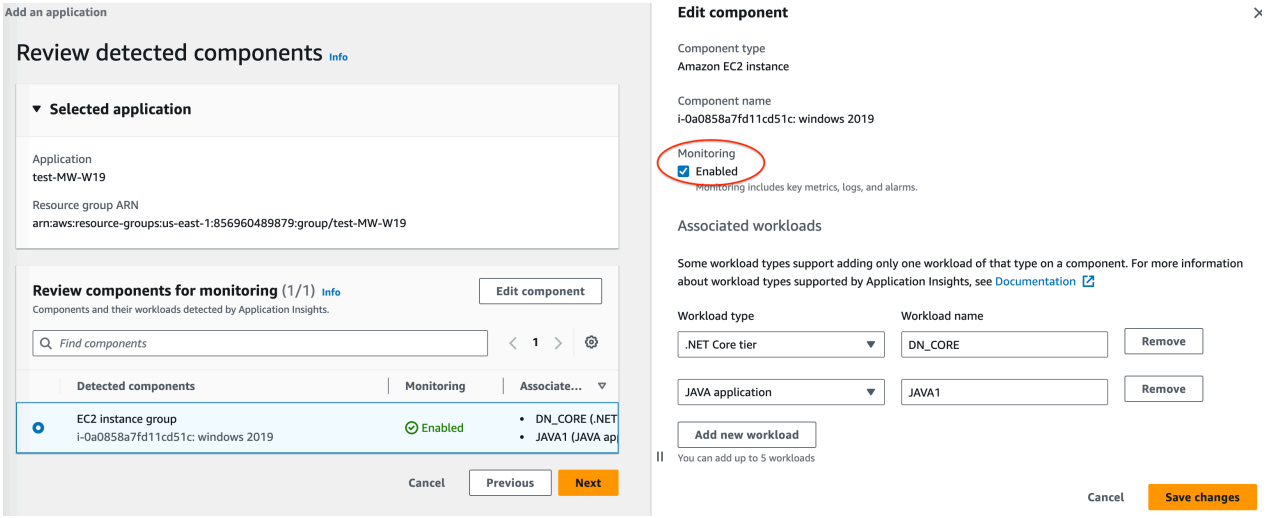
- Pour ajouter une charge de travail au composant, choisissez Ajouter une charge de travail.



  - Si l'option Ajouter une charge de travail ne s'affiche pas, ce composant ne prend pas en charge plusieurs charges de travail.
  - Si l'en-tête Charges de travail associées n'apparaît pas, ce composant ne prend pas en charge les charges de travail personnalisées.
- Pour supprimer une charge de trravail, choisissez Supprimer à côté de la charge de travail que vous souhaitez supprimer de la surveillance.



- Pour désactiver la surveillance de l'ensemble du composant, décochez la case Surveillé.



- Lorsque vous avez terminé de modifier le composant, choisissez Enregistrer les modifications dans le coin inférieur droit. Toute modification apportée aux charges de travail d'un composant est visible dans la table Vérifier les composants à des fins de surveillance sous Charges de travail associées.

8. Sur la page Examiner les composants détectés, choisissez Suivant.
9. La page Spécifier les détails du composant inclut tous les composants avec des charges de travail associées personnalisables de l'étape précédente.

> **Note**
>
> Si l'en-tête d'un composant possède une balise facultative, les détails supplémentaires relatifs aux charges de travail de ce composant sont facultatifs.

Si aucun composant n'apparaît sur cette page, aucun détail supplémentaire ne peut être spécifié au cours de cette étape.

10. Choisissez Suivant.
11. Sur la page Vérifier et soumettre, passez en revue tous les détails des composants et de la charge de travail surveillés.
12. Sélectionnez Envoyer.

Account-based application monitoring

1. Nom de l'application. Saisissez le nom de l'application basée sur un compte.
2. Surveillance automatisée des nouvelles ressources. Par défaut, Application Insights utilise les paramètres recommandés pour configurer la surveillance des composants de ressources ajoutés à votre compte après que vous ayez intégré l'application. Vous pouvez exclure la surveillance des ressources ajoutées après l'onboarding de votre application en décochant la case.
3. Surveillez CloudWatch les événements. Cochez la case pour intégrer la surveillance des applications aux CloudWatch événements afin d'obtenir des informations provenant d'Amazon EBS, d'Amazon EC2 AWS CodeDeploy, d'Amazon ECS, des API et des notifications AWS Health , d'Amazon RDS, d'Amazon S3 et. AWS Step Functions
4. Intégrez avec AWS Systems Manager OpsCenter. Pour consulter les applications sélectionnées et être averti lorsque des problèmes sont détectés, cochez la case Generate Systems Manager OpsCenter OpsItems pour les actions correctives. Pour suivre les opérations effectuées pour résoudre les éléments de travail opérationnels (OpsItems) liés à vos AWS ressources, fournissez l'ARN de la rubrique SNS.
5. Balises : facultatif. CloudWatch Application Insights prend en charge les groupes de ressources basés CloudFormation sur des balises et des groupes de ressources (à l'exception des groupes Auto Scaling). Pour plus d'informations, consultez Utilisation de Tag Editor.
6. Ressources détectées. Toutes les ressources découvertes dans votre compte sont ajoutées à cette liste. Si Application Insights ne peut pas découvrir toutes les ressources de votre compte, un message d'erreur apparaît en haut de la page. Ce message contient un lien vers la documentation concernant l'ajout des autorisations requises.
7. Sélectionnez Next (Suivant).

Un ARN sera généré pour l'application au format suivant.

```
arn:partition:applicationinsights:region:account-id:application/
TBD/application-name
```

Exemple

```
arn:aws:applicationinsights:us-east-1:123456789012:application/TBD/my-
application
```

4. Une fois que vous aurez soumis la configuration de surveillance de votre application, vous serez redirigé vers la page de détails de l'application, où vous pouvez afficher le Résumé de l'application, la liste des Composants surveillés et des Composants non surveillés et, en sélectionnant les onglets à côté de Composants, l'Historique des configurations, Modèles de journaux et n'importe quelle Étiquette que vous avez appliquée.

   Pour afficher les informations relatives à l'application, sélectionnez View Insights (Afficher les analyses).

   Vous pouvez mettre à jour vos sélections pour la surveillance des CloudWatch événements et l'intégration avec AWS Systems Manager OpsCenter en choisissant Modifier.

   Sous Components (Composants), vous pouvez sélectionner le menu Actions pour créer, modifier ou dissocier un groupe d'instances.

   Vous pouvez gérer la surveillance des composants, y compris le niveau d'application, les groupes de journaux, les journaux d'événements, les métriques et les alertes personnalisées, en sélectionnant la puce à côté d'un composant et en sélectionnant Manage monitoring (Gérer la surveillance).

Activer la surveillance des ressources Application Insights pour Amazon ECS et Amazon EKS

Vous pouvez activer Application Insights pour surveiller les applications conteneurisées et les microservices à partir de la console Container Insights. Application Insights prend en charge la surveillance des ressources suivantes :

- Clusters Amazon ECS
- Services Amazon ECS

- Tâches Amazon ECS
- Clusters Amazon EKS

Lorsque Application Insights est activé, il fournit des métriques et des journaux recommandés, détecte les problèmes potentiels, génère des CloudWatch événements et crée des tableaux de bord automatiques pour vos applications conteneurisées et vos microservices.

Vous pouvez activer Application Insights pour les ressources conteneurisées à partir des consoles Container Insights ou Application Insights.

Activer Application Insights à partir de la console Container Insights

Depuis la console Container Insights, sur Container Insights Surveillance des performances Tableau de bord, sélectionnez Auto-configure Application Insights (Configuration automatique de l'application Insights). Lorsque Application Insights est activée, cette rubrique affiche des détails sur les problèmes détectés.

Activer Application Insights à partir de la console Application Insights

Lorsque les clusters ECS apparaissent dans la liste des composants, Application Insights permet automatiquement une surveillance supplémentaire des conteneurs à l'aide de Container Insights.

Pour les clusters EKS, vous pouvez activer une surveillance supplémentaire avec Container Insights pour fournir des informations de diagnostic, telles que les échecs de redémarrage de conteneur, afin de vous aider à isoler et à résoudre les problèmes. Des étapes supplémentaires sont nécessaires pour configurer Container Insights pour EKS. Pour plus d'informations, consultez Configuration de Container Insights sur Amazon EKS et Kubernetes pour savoir comment configurer Container Insights sur EKS.

Une surveillance supplémentaire pour EKS avec Container Insights est prise en charge sur les instances Linux avec EKS.

Pour plus d'informations sur la prise en charge de Container Insights pour les clusters ECS et EKS, consultez Container Insights.

Désactiver la surveillance d'un composant d'application

Pour désactiver la surveillance d'un composant d'application à partir de la page Détails de l'application, sélectionnez le composant pour lequel vous souhaitez désactiver la surveillance. Sélectionnez Actions, puis Remove from monitoring (Supprimer de la surveillance).

Supprimer une application

Pour supprimer une application, dans le CloudWatch tableau de bord, dans le volet de navigation de gauche, choisissez Application Insights sous Insights. Sélectionnez l'application que vous souhaitez supprimer. Sous Actions, sélectionnez Delete application (Supprimer une application). Cela supprime la surveillance ainsi que toutes les surveillances enregistrées pour les composants d'application. Les ressources d'applications ne sont pas supprimées.

Configurer et gérer votre application pour la surveillance à l'aide de la ligne de commande

Cette section décrit les étapes de configuration, de configuration et de gestion de votre application de surveillance à l'aide du AWS CLI et AWS Tools for Windows PowerShell.

Procédures de ligne de commande

- Ajouter et gérer une application
- Gestion et mise à jour du contrôle
- Configurer la surveillance pour les groupes de disponibilité SQL Always On
- Configurer la surveillance pour MySQL RDS
- Configurer la surveillance pour MySQL EC2
- Configurer la surveillance pour PostgreSQL RDS
- Configurer la surveillance pour PostgreSQL EC2
- Configurer la surveillance pour Oracle RDS
- Configurer la surveillance pour Oracle EC2

Ajouter et gérer une application

Vous pouvez ajouter ou obtenir des informations, gérer et configurer votre application Application Insights à l'aide de la ligne de commande.

Rubriques

- Ajout d'une application
- Description d'une application
- Liste des composants d'une application
- Description d'un composant
- Regroupement de ressources similaires dans un composant personnalisé
- Dissociation d'un composant personnalisé

- Mise à jour d'une application
- Mise à jour d'un composant personnalisé

Ajout d'une application

Ajoutez une application à l'aide du AWS CLI

Pour ajouter une application AWS CLI à votre groupe de ressources appelé`my-resource-group`, avec OpsCenter activé pour fournir l'OPSItem créé à l' `arn:aws:sns:us-east-1:123456789012:MyTopic`ARN de la rubrique SNS, utilisez la commande suivante.

```
aws application-insights create-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Ajoutez une application à l'aide de AWS Tools for Windows PowerShell

Pour ajouter une application AWS Tools for Windows PowerShell à votre groupe de ressources appelé `my-resource-group` avec OpsCenter enabled afin de fournir l'OPSItem créé à l'`arn:aws:sns:us-east-1:123456789012:MyTopic`ARN de la rubrique SNS, utilisez la commande suivante.

```
New-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Description d'une application

Décrivez une application utilisant AWS CLI

AWS CLI Pour décrire une application créée sur un groupe de ressources appelé`my-resource-group`, utilisez la commande suivante.

```
aws application-insights describe-application --resource-group-name my-resource-group
```

Décrire une application utilisant AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell Pour décrire une application créée sur un groupe de ressources appelé`my-resource-group`, utilisez la commande suivante.

```
Get-CWAIApplication -ResourceGroupName my-resource-group
```

Liste des composants d'une application

Répertoriez les composants d'une application à l'aide du AWS CLI

Pour AWS CLI répertorier les composants créés sur un groupe de ressources appelé`my-resource-group`, utilisez la commande suivante.

```
aws application-insights list-components --resource-group-name my-resource-group
```

Répertoriez les composants d'une application à l'aide de AWS Tools for Windows PowerShell

Pour AWS Tools for Windows PowerShell répertorier les composants créés sur un groupe de ressources appelé`my-resource-group`, utilisez la commande suivante.

```
Get-CWAIComponentList -ResourceGroupName my-resource-group
```

Description d'un composant

Décrivez un composant à l'aide du AWS CLI

Vous pouvez utiliser la AWS CLI commande suivante pour décrire un composant appelé `my-component` qui appartient à une application créée sur un groupe de ressources appelé`my-resource-group`.

```
aws application-insights describe-component --resource-group-name my-resource-group --
component-name my-component
```

Décrire un composant en utilisant AWS Tools for Windows PowerShell

Vous pouvez utiliser la AWS Tools for Windows PowerShell commande suivante pour décrire un composant appelé `my-component` qui appartient à une application créée sur un groupe de ressources appelé`my-resource-group`.

```
Get-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Regroupement de ressources similaires dans un composant personnalisé

Nous recommandons de regrouper les ressources similaires, telles que les instances de serveur Web .NET, sous forme de composants personnalisés pour faciliter l'onboarding et améliorer la surveillance et les informations. Actuellement, CloudWatch Application Insights prend en charge les groupes personnalisés pour les instances EC2.

Pour regrouper des ressources dans un composant personnalisé à l'aide de l' AWS CLI

AWS CLI Pour regrouper trois instances (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111arn:aws:ec2:us-east-1:123456789012:instance/i-22222`, et `arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) dans un composant personnalisé appelé `my-component` pour une application créée pour le groupe de ressources appelé`my-resource-group`, utilisez la commande suivante.

```
aws application-insights create-component --resource-group-name my-
resource-group --component-name my-component --resource-list arn:aws:ec2:us-
east-1:123456789012:instance/i-11111 arn:aws:ec2:us-east-1:123456789012:instance/
i-22222 arn:aws:ec2:us-east-1:123456789012:instance/i-33333
```

Pour regrouper des ressources dans un composant personnalisé à l'aide des AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell Pour regrouper trois instances (`arn:aws:ec2:us-east-1:123456789012:instance/i-11111`,`arn:aws:ec2:us-east-1:123456789012:instance/i-22222`, et`arn:aws:ec2:us-east-1:123456789012:instance/i-33333`) dans un composant personnalisé appelé`my-component`, pour une application créée pour le groupe de ressources appelé`my-resource-group`, utilisez la commande suivante.

```
New-CWAIComponent -ResourceGroupName my-resource-group -ComponentName my-component
 -ResourceList arn:aws:ec2:us-east-1:123456789012:instance/i-11111,arn:aws:ec2:us-
east-1:123456789012:instance/i-22222,arn:aws:ec2:us-east-1:123456789012:instance/
i-33333
```

Dissociation d'un composant personnalisé

Pour dissocier un composant personnalisé à l'aide du AWS CLI

Pour AWS CLI dissocier un composant personnalisé nommé `my-component` dans une application créée sur le groupe de ressources`my-resource-group`, utilisez la commande suivante.

```
aws application-insights delete-component --resource-group-name my-resource-group --
component-name my-new-component
```

Pour dissocier un composant personnalisé à l'aide de AWS Tools for Windows PowerShell

Pour AWS Tools for Windows PowerShell dissocier un composant personnalisé nommé `my-component` dans une application créée sur le groupe de ressources`my-resource-group`, utilisez la commande suivante.

```
Remove-CWAIComponent -ComponentName my-component -ResourceGroupName my-resource-group
```

Mise à jour d'une application

Mettez à jour une application à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour mettre à jour une application afin de générer AWS Systems Manager OpsCenter OpsItems pour les problèmes détectés avec l'application, et pour associer la création OpsItems à la rubrique SNS `arn:aws:sns:us-east-1:123456789012:MyTopic`, à l'aide de la commande suivante.

```
aws application-insights update-application --resource-group-name my-resource-group --
ops-center-enabled --ops-item-sns-topic-arn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Mettre à jour une application à l'aide AWS des Outils pour Windows PowerShell

Vous pouvez utiliser le AWS Tools for Windows PowerShell pour mettre à jour une application afin de générer un AWS SSM OpsCenter OpsItems pour les problèmes détectés avec l'application, et pour associer le message créé OpsItems au sujet SNS `arn:aws:sns:us-east-1:123456789012:MyTopic`, à l'aide de la commande suivante.

```
Update-CWAIApplication -ResourceGroupName my-resource-group -OpsCenterEnabled true -
OpsItemSNSTopicArn arn:aws:sns:us-east-1:123456789012:MyTopic
```

Mise à jour d'un composant personnalisé

Mettez à jour un composant personnalisé à l'aide du AWS CLI

Vous pouvez utiliser le AWS CLI pour mettre à jour un composant personnalisé appelé `my-component` avec un nouveau nom de composant et un groupe d'instances mis à jour à l'aide de la commande suivante. `my-new-component`

```
aws application-insights update-component --resource-group-name my-resource-
group --component-name my-component --new-component-name my-new-component --
resource-list arn:aws:ec2:us-east-1:123456789012:instance/i-44444 arn:aws:ec2:us-
east-1:123456789012:instance/i-55555
```

Mettre à jour un composant personnalisé à l'aide AWS des outils pour Windows PowerShell

Vous pouvez utiliser le AWS Tools for Windows PowerShell pour mettre à jour un composant personnalisé appelé `my-component` avec un nouveau nom de composant et un groupe d'instances mis à jour à l'aide de la commande suivante. `my-new-component`

```
Update-CWAIComponent -ComponentName my-component -NewComponentName my-new-
component -ResourceGroupName my-resource-group -ResourceList arn:aws:ec2:us-
east-1:123456789012:instance/i-44444,arn:aws:ec2:us-east-1:123456789012:instance/
i-55555
```

Gestion et mise à jour du contrôle

Vous pouvez gérer et mettre à jour la surveillance de votre application Application Insights à l'aide de la ligne de commande.

Rubriques

- Liste des problèmes liés à votre application
- Description d'un problème d'application
- Description des anomalies ou des erreurs associées à un problème
- Description d'une anomalie ou d'une erreur liée à l'application
- Description des configurations de surveillance d'un composant
- Description de la configuration de surveillance recommandée d'un composant
- Mise à jour des configurations de surveillance d'un composant
- Supprimer un Resource Group spécifié de la surveillance Application Insights

Liste des problèmes liés à votre application

Répertoriez les problèmes liés à votre application à l'aide du AWS CLI

Pour AWS CLI répertorier les problèmes détectés avec votre application entre 1 000 et 10 000 millisecondes depuis l'époque Unix pour une application créée sur un groupe de ressources appelé`my-resource-group`, utilisez la commande suivante.

```
aws application-insights list-problems --resource-group-name my-resource-group --start-
time 1000 --end-time 10000
```

Répertoriez les problèmes liés à votre application à l'aide AWS des Outils pour Windows PowerShell

Pour AWS Tools for Windows PowerShell répertorier les problèmes détectés avec votre application entre 1 000 et 10 000 millisecondes depuis l'époque Unix pour une application créée sur un groupe de ressources appelémy-resource-group, utilisez la commande suivante.

```
$startDate = "8/6/2019 3:33:00"
$endDate = "8/6/2019 3:34:00"
Get-CWAIProblemList -ResourceGroupName my-resource-group -StartTime $startDate -
EndTime $endDate
```

Description d'un problème d'application

Décrivez un problème d'application à l'aide du AWS CLI

Pour utiliser le AWS CLI pour décrire un problème lié à l'identifiant du problèmep-1234567890, utilisez la commande suivante.

```
aws application-insights describe-problem –problem-id p-1234567890
```

Décrire un problème d'application à l'aide de AWS Tools for Windows PowerShell

Pour utiliser le AWS Tools for Windows PowerShell pour décrire un problème lié à l'identifiant du problèmep-1234567890, utilisez la commande suivante.

```
Get-CWAIProblem -ProblemId p-1234567890
```

Description des anomalies ou des erreurs associées à un problème

Décrivez les anomalies ou les erreurs associées à un problème à l'aide du AWS CLI

AWS CLI Pour décrire les anomalies ou les erreurs associées à un problème lié à l'identifiant du problèmep-1234567890, utilisez la commande suivante.

```
aws application-insights describe-problem-observations --problem-id -1234567890
```

Décrire les anomalies ou erreurs associées à un problème à l'aide des AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell Pour décrire les anomalies ou les erreurs associées à un problème lié à l'identifiant du problèmep-1234567890, utilisez la commande suivante.

```
Get-CWAIProblemObservation -ProblemId p-1234567890
```

Description d'une anomalie ou d'une erreur liée à l'application

Décrire une anomalie ou une erreur liée à l'application à l'aide de l'interface de ligne de commande (CLI) AWS

AWS CLI Pour décrire une anomalie ou une erreur dans l'application avec l'identifiant d'observation`o-1234567890`, utilisez la commande suivante.

```
aws application-insights describe-observation –observation-id o-1234567890
```

Décrivez une anomalie ou une erreur dans l'application à l'aide AWS des Outils pour Windows PowerShell

AWS Tools for Windows PowerShell Pour décrire une anomalie ou une erreur dans l'application avec l'identifiant d'observation`o-1234567890`, utilisez la commande suivante.

```
Get-CWAIObservation -ObservationId o-1234567890
```

Description des configurations de surveillance d'un composant

Décrire les configurations de surveillance d'un composant à l'aide de l' AWS CLI

AWS CLI Pour décrire la configuration de surveillance d'un composant appelé `my-component` dans une application créée sur le groupe de ressources`my-resource-group`, utilisez la commande suivante.

```
aws application-insights describe-component-configuration –resource-group-name my-
resource-group –component-name my-component
```

Décrire les configurations de surveillance d'un composant à l'aide AWS des outils pour Windows PowerShell

AWS Tools for Windows PowerShell Pour décrire la configuration de surveillance d'un composant appelé`my-component`, dans une application créée sur le groupe de ressources`my-resource-group`, utilisez la commande suivante.

```
Get-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-
resource-group
```

Pour plus d'informations sur la configuration des composants et pour obtenir des exemples de fichiers JSON, consultez Utilisation de configurations de composants.

Description de la configuration de surveillance recommandée d'un composant

Décrivez la configuration de surveillance recommandée pour un composant à l'aide du AWS CLI

Lorsque le composant fait partie d'une application .NET Worker, vous pouvez utiliser le AWS CLI pour décrire la configuration de surveillance recommandée d'un composant appelé `my-component` dans une application créée sur le groupe de ressources`my-resource-group`, à l'aide de la commande suivante.

```
aws application-insights describe-component-configuration-recommendation --resource-
group-name my-resource-group --component-name my-component --tier DOT_NET_WORKER
```

Décrire la configuration de surveillance recommandée pour un composant à l'aide de AWS Tools for Windows PowerShell

Lorsque le composant fait partie d'une application .NET Worker, vous pouvez utiliser le AWS Tools for Windows PowerShell pour décrire la configuration de surveillance recommandée d'un composant appelé `my-component` dans une application créée sur le groupe de ressources`my-resource-group`, à l'aide de la commande suivante.

```
Get-CWAIComponentConfigurationRecommendation -ComponentName my-component -
ResourceGroupName my-resource-group -Tier DOT_NET_WORKER
```

Pour plus d'informations sur la configuration des composants et pour obtenir des exemples de fichiers JSON, consultez Utilisation de configurations de composants.

Mise à jour des configurations de surveillance d'un composant

Mettre à jour les configurations de surveillance d'un composant à l'aide de l' AWS CLI

Pour mettre AWS CLI à jour le composant appelé `my-component` dans une application créée sur le groupe de ressources appelé`my-resource-group`, utilisez la commande suivante. La commande inclut les actions suivantes :

1. Activation de la surveillance du composant
2. Définition du niveau du composant sur `.NET Worker`.
3. Mise à jour de la configuration JSON du composant afin d'effectuer la lecture à partir du fichier local `configuration.txt`.

```
aws application-insights update-component-configuration --resource-group-name my-
resource-group --component-name my-component --tier DOT_NET_WORKER --monitor --
component-configuration "file://configuration.txt"
```

Mettre à jour les configurations de surveillance d'un composant à l'aide des AWS Tools for Windows PowerShell

Pour mettre AWS Tools for Windows PowerShell à jour le composant appelé `my-component` dans une application créée sur le groupe de ressources appelé`my-resource-group`, utilisez la commande suivante. La commande inclut les actions suivantes :

1. Activation de la surveillance du composant
2. Définition du niveau du composant sur `.NET Worker`.
3. Mise à jour de la configuration JSON du composant afin d'effectuer la lecture à partir du fichier local `configuration.txt`.

```
[string]$config = Get-Content -Path configuration.txt
Update-CWAIComponentConfiguration -ComponentName my-component -ResourceGroupName my-
resource-group -Tier DOT_NET_WORKER -Monitor 1 -ComponentConfiguration $config
```

Pour plus d'informations sur la configuration des composants et pour obtenir des exemples de fichiers JSON, consultez Utilisation de configurations de composants.

Supprimer un Resource Group spécifié de la surveillance Application Insights

Supprimer un groupe de ressources spécifié de la surveillance d'Application Insights à l'aide du AWS CLI

AWS CLI Pour supprimer une application créée sur le groupe de ressources appelé `my-resource-group` de la surveillance, utilisez la commande suivante.

```
aws application-insights delete-application --resource-group-name my-resource-group
```

Supprimer un groupe de ressources spécifié de la surveillance d'Application Insights à l'aide du AWS Tools for Windows PowerShell

AWS Tools for Windows PowerShell Pour supprimer une application créée sur le groupe de ressources appelé `my-resource-group` de la surveillance, utilisez la commande suivante.

```
Remove-CWAIApplication -ResourceGroupName my-resource-group
```

Configurer la surveillance pour les groupes de disponibilité SQL Always On

1. Créez une application pour le Resource Group avec les instances SQL HA EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Définissez les instances EC2 qui représentent le cluster SQL HA en créant un nouveau composant d'application.

```
aws application-insights create-component --resource-group-name
 "<RESOURCE_GROUP_NAME>" --component-name SQL_HA_CLUSTER --resource-list
  "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_1_ID>"
 "arn:aws:ec2:<REGION>:<ACCOUNT_ID>:instance/<CLUSTER_INSTANCE_2_ID>
```

3. Configurez le composant SQL HA.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "SQL_HA_CLUSTER" --
monitor --tier SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP --monitor  --component-
configuration '{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/
sec",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-
<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\
\MSSQL\\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
```

```
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}'
```

> **Note**
>
> Application Insights doit intégrer les journaux des événements d'application (niveau d'information) pour détecter les activités de cluster telles que le basculement.

Configurer la surveillance pour MySQL RDS

1. Créez une application pour le groupe de ressources avec l'instance de base de données RDS MySQL.

   ```
   aws application-insights create-application --region <REGION> --resource-group-name
     <RESOURCE_GROUP_NAME>
   ```

2. Le journal d'erreurs est activé par défaut. Le journal des requêtes lentes peut être activé à l'aide de groupes de paramètres de données. Pour plus d'informations, consultez Accès au journal des requêtes lentes et au journal général MySQL.

- `set slow_query_log = 1`
- `set log_output = FILE`

3. Exportez les journaux à surveiller vers CloudWatch des journaux. Pour plus d'informations, consultez Publier des journaux MySQL dans des CloudWatch journaux.
4. Configurez le composant MySQL RDS.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier DEFAULT --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logType\":
\"MYSQL\",\"monitor\":true},{\"logType\": \"MYSQL_SLOW_QUERY\",\"monitor\":false}]}"
```

Configurer la surveillance pour MySQL EC2

1. Créez une application pour le groupe de ressources avec les instances SQL HA EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Le journal d'erreurs est activé par défaut. Le journal des requêtes lentes peut être activé à l'aide de groupes de paramètres de données. Pour plus d'informations, consultez Accès au journal des requêtes lentes et au journal général MySQL.
   - `set slow_query_log = 1`
   - `set log_output = FILE`
3. Configurez le composant MySQL EC2.

```
aws application-insights  update-component-configuration --resource-group-name
 "<RESOURCE_GROUP_NAME>" --region <REGION> --component-name "<DB_COMPONENT_NAME>"
 --monitor --tier MYSQL --monitor  --component-configuration "{\"alarmMetrics\":
[{\"alarmMetricName\":\"CPUUtilization\",\"monitor\":true}],\"logs\":[{\"logGroupName
\":\"<UNIQUE_LOG_GROUP_NAME>\",\"logPath\":\"C:\\\\ProgramData\\\\MySQL\\\\MySQL
 Server **\\\\Data\\\\<FILE_NAME>.err\",\"logType\":\"MYSQL\",\"monitor\":true,
\"encoding\":\"utf-8\"}]}"
```

Configurer la surveillance pour PostgreSQL RDS

1. Créez une application pour le Resource Group avec l'instance de base de données PostgreSQL RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. La publication des journaux PostgreSQL sur n'est pas CloudWatch activée par défaut. Pour activer la surveillance, ouvrez la console RDS et sélectionnez la base de données à surveiller. Sélectionnez Modify (Modifier) dans le coin supérieur droit et cochez la case intitulée journal PostgreSQL. Sélectionnez Continuer pour enregistrer ce paramètre.
3. Vos journaux PostgreSQL sont exportés vers. CloudWatch
4. Configurez le composant PostgreSQL RDS.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"POSTGRESQL\",
         \"monitor\": true
      }
   ]
}"
```

Configurer la surveillance pour PostgreSQL EC2

1. Créez une application pour le Resource Group avec l'instance PostgreSQL EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configurez le composant PostgreSQL EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier POSTGRESQL --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/var/lib/pgsql/data/log/\",
         \"logType\":\"POSTGRESQL\",
         \"monitor\":true,
         \"encoding\":\"utf-8\"
      }
   ]
}"
```

Configurer la surveillance pour Oracle RDS

1. Créez une application pour le Resource Group avec l'instance de base de données Oracle RDS.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. La publication des journaux Oracle sur n' CloudWatch est pas activée par défaut. Pour activer la surveillance, ouvrez la console RDS et sélectionnez la base de données à surveiller. Sélectionnez Modify (Modifier) dans le coin supérieur droit, puis cochez les cases intitulées journal Alert (Alerte) et journal Listener (Écouteur). Sélectionnez Continuer pour enregistrer ce paramètre.
3. Vos journaux Oracle sont exportés vers CloudWatch.
4. Configurez le composant Oracle RDS.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier DEFAULT --component-configuration
```

```
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\": \"CPUUtilization\",
         \"monitor\": true
      }
   ],
   \"logs\":[
      {
         \"logType\": \"ORACLE_ALERT\",
         \"monitor\": true
      },
      {
         \"logType\": \"ORACLE_LISTENER\",
         \"monitor\": true
      }
   ]
}"
```

Configurer la surveillance pour Oracle EC2

1. Créez une application pour le Resource Group avec l'instance Oracle EC2.

```
aws application-insights create-application --region <REGION> --resource-group-name
  <RESOURCE_GROUP_NAME>
```

2. Configurez le composant Oracle EC2.

```
aws application-insights update-component-configuration --region <REGION> --resource-
group-name <RESOURCE_GROUP_NAME> --component-name <DB_COMPONENT_NAME> --monitor --
tier ORACLE --component-configuration
"{
   \"alarmMetrics\":[
      {
         \"alarmMetricName\":\"CPUUtilization\",
         \"monitor\":true
      }
   ],
   \"logs\":[
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/rdbms/*/*/trace\",
```

```
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      },
      {
         \"logGroupName\":\"<UNIQUE_LOG_GROUP_NAME>\",
         \"logPath\":\"/opt/oracle/diag/tnslsnr/$HOSTNAME/listener/trace/\",
         \"logType\":\"ORACLE_ALERT\",
         \"monitor\":true,
      }
   ]
}"
```

Application Insights CloudWatch Événements et notifications en cas de problèmes détectés

Pour chaque application ajoutée à CloudWatch Application Insights, un CloudWatch événement est publié pour les événements suivants dans la mesure du possible :

- Création de problème. Émis lorsque CloudWatch Application Insights détecte un nouveau problème.
  - Type de détail : « Problème Application Insights détecté »
  - Détail :
    - `problemId` : l'ID du problème détecté.
    - `region`: AWS Région dans laquelle le problème a été créé.
    - `resourceGroupName` : le Resource Group de l'application enregistrée pour lequel le problème a été détecté ;
    - `status` : le statut du problème. Les statuts et les définitions possibles sont les suivants :
      - `In progress` : Un nouveau problème a été identifié. Le problème reçoit encore des observations.
      - `Recovering` : Le problème est en train de se stabiliser. Vous pouvez résoudre manuellement le problème lorsqu'il est dans cet état.
      - `Resolved` : Le problème est résolu. Il n'y a pas de nouvelles observations concernant ce problème.
      - `Recurring` : Le problème a été résolu au cours des dernières 24 heures. Il s'est rouvert à la suite d'observations supplémentaires.
    - `severity`: la gravité du problème.
    - `problemUrl`: la console URL du problème.

- Mise à jour du problème. Émis lorsque le problème est mis à jour avec une nouvelle observation ou lorsqu'une observation existante est mise à jour et que le problème est ensuite mis à jour ; les mises à jour comprennent une résolution ou la clôture du problème.
  - Type de détail : « Mise à jour de problèmes Application Insights »
  - Détail :
    - `problemId` : l'ID du problème créé.
    - `region`: AWS Région dans laquelle le problème a été créé.
    - `resourceGroupName` : le Resource Group de l'application enregistrée pour lequel le problème a été détecté ;
    - `status` : le statut du problème.
    - `severity`: la gravité du problème.
    - `problemUrl`: la console URL du problème.

Comment recevoir une notification pour les événements de problèmes générés par une application

Dans la CloudWatch console, sélectionnez Règles sous Événements dans le volet de navigation de gauche. Dans la page Règles, sélectionnez Créer une règle. Choisissez Amazon CloudWatch Application Insights dans la liste déroulante Nom du service et choisissez le type d'événement. Ensuite, sélectionnez Ajouter une cible et sélectionnez la cible et les paramètres, par exemple une rubrique SNS ou une fonction Lambda.

Actions par le biais de AWS Systems Manager. CloudWatch Application Insights fournit une intégration intégrée à Systems Manager OpsCenter. Si vous choisissez d'utiliser cette intégration pour votre application, une intégration OpsItem est créée sur la OpsCenter console pour chaque problème détecté avec l'application. Depuis la OpsCenter console, vous pouvez consulter des informations résumées sur le problème détecté par CloudWatch Application Insights et choisir un runbook de Systems Manager Automation pour prendre des mesures correctives ou mieux identifier les processus Windows à l'origine des problèmes de ressources dans votre application.

Observabilité inter-comptes Application Insights

Grâce à CloudWatch l'observabilité entre comptes d'Application Insights, vous pouvez surveiller et dépanner vos applications qui couvrent plusieurs AWS comptes au sein d'une même région.

Vous pouvez utiliser Amazon CloudWatch Observability Access Manager pour configurer un ou plusieurs de vos AWS comptes en tant que compte de surveillance. Vous allez permettre au compte

de surveillance de consulter les données de votre compte source en créant un récepteur dans votre compte de surveillance. Vous utilisez le récepteur pour créer un lien entre votre compte source et votre compte de surveillance. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

Ressources requises

Pour garantir le bon fonctionnement de CloudWatch l'observabilité entre comptes Application Insights, assurez-vous que les types de télémétrie suivants sont partagés via le gestionnaire d'accès à l' CloudWatch observabilité.

- Applications dans CloudWatch Application Insights
- Métriques sur Amazon CloudWatch
- Groupes de journaux dans Amazon CloudWatch Logs
- Traces dans AWS X-Ray

Utilisation de configurations de composants

Une configuration de composant est un fichier texte au format JSON qui décrit les paramètres de configuration du composant. Cette section fournit un exemple de fragment de modèle, des descriptions des sections de configuration de composants et des exemples de configurations de composants.

Rubriques

- Fragment de modèle de configuration de composants
- Sections de configuration de composant
- Exemples de configuration de composants

Fragment de modèle de configuration de composants

L'exemple suivant montre un fragment de modèle au format JSON.

```
{
  "alarmMetrics" : [
    list of alarm metrics
  ],
  "logs" : [
    list of logs
```

```
  ],
  "processes" : [
   list of processes
  ],
  "windowsEvents" : [
    list of windows events channels configurations
  ],
  "alarms" : [
    list of CloudWatch alarms
  ],
  "jmxPrometheusExporter": {
    JMX Prometheus Exporter configuration
  },
  "hanaPrometheusExporter": {
      SAP HANA Prometheus Exporter configuration
  },
  "haClusterPrometheusExporter": {
      HA Cluster Prometheus Exporter configuration
  },
  "netWeaverPrometheusExporter": {
      SAP NetWeaver Prometheus Exporter configuration
  },
  "subComponents" : [
    {
      "subComponentType" : "AWS::EC2::Instance" ...
      component nested instances configuration
    },
    {
      "subComponentType" : "AWS::EC2::Volume" ...
      component nested volumes configuration
    }
  ]
}
```

Sections de configuration de composant

Une configuration de composant comprend plusieurs sections majeures. Les sections d'une configuration de composant peuvent être listées dans n'importe quel ordre.

- alarmMetrics (en option)

  Liste des métriques à surveiller pour le composant. Tous les types de composant peuvent avoir une section alarmMetrics.

- logs (en option)

  Liste des journaux à surveiller pour le composant. Seules les instances EC2 peuvent avoir une section de journaux.

- processes (facultatif)

  Liste des processus à surveiller pour le composant. Seules les instances EC2 peuvent avoir une section relative aux processus.

- sous-composants (en option)

  Configuration d'instance imbriquée et du volume de subComponent pour le composant. Les types de composants suivants peuvent avoir des instances imbriquées et une section subComponents : ELB, ASG, les instances EC2 regroupées personnalisées et des instances EC2.

- alertes (en option)

  Liste des alertes à surveiller pour le composant. Tous les types de composant peuvent avoir une section alerte.

- windowsEvents (en option)

  Liste des événements Windows à surveiller pour le composant. Seul Windows dispose d'une section `windowsEvents` sur les instances EC2.

- JMX PrometheusExporter (facultatif)

  Configuration de JMXPrometheus Exporter.

- hanaPrometheusExporter (facultatif)

  Configuration de l'exportateur SAP HANA Prometheus.

- haClusterPrometheusExportateur (facultatif)

  Configuration de l'exportateur Prometheus Cluster HA.

- netWeaverPrometheusExportateur (facultatif)

  Configuration de NetWeaver l'exportateur SAP Prometheus.

- sapAsePrometheusExportateur (facultatif)

  Configuration de SAP ASE Prometheus Exporter.

L'exemple suivant montre la syntaxe du fragment de la section subComponents au format JSON.

```
[
  {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [
      list of alarm metrics
    ],
    "logs" : [
      list of logs
    ],
    "processes": [
      list of processes
    ],
    "windowsEvents" : [
      list of windows events channels configurations
    ]
  },
  {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [
      list of alarm metrics
    ]
  }
]
```

Propriétés de la section Configuration de composant

Cette section décrit les propriétés de chaque section Configuration de composant.

Sections

- Métrique
- Journal
- Processus
- JMX Prometheus Exporter
- Exportateur HANA Prometheus
- Exportateur HA Cluster Prometheus
- NetWeaver Prometheus Exportateur
- SAP ASE Prometheus Exporter
- Événements Windows

- alerte

Métrique

Définit une métrique à surveiller pour le composant.

JSON

```
{
  "alarmMetricName" : "monitoredMetricName",
  "monitor" : true/false
}
```

Propriétés

- alarmMetricName (obligatoire)

  Le nom de la métrique à surveiller pour le composant. Pour les métriques prises en charge par Application Insights, consultez Logs et statistiques pris en charge par Amazon CloudWatch Application Insights.
- monitor (en option)

  Valeur booléenne pour indiquer s'il faut surveiller la métrique. La valeur par défaut est `true`.

Journal

Définit un journal à surveiller pour le composant.

JSON

```
{
  "logGroupName" : "logGroupName",
  "logPath" : "logPath",
  "logType" : "logType",
  "encoding" : "encodingType",
  "monitor" : true/false
}
```

Propriétés

- logGroupName (obligatoire)

Le nom du groupe de CloudWatch journaux à associer au journal surveillé. Pour les contraintes liées au nom des groupes de journaux, consultez CreateLogGroup.

- LogPath (obligatoire pour les composants de l'instance EC2 ; non requis pour les composants qui n'utilisent pas l' CloudWatchagent, tels que) AWS Lambda

  Chemin des journaux à surveiller. Le chemin d'accès au journal doit être un chemin absolu de fichier système Windows. Pour plus d'informations, consultez la section Fichier de configuration de l'CloudWatch agent : journaux.

- logType (obligatoire)

  Le type de journal décide des modèles de journal par rapport auxquels Application Insights analyse le journal. Le type de journal est sélectionné parmi les éléments suivants :

  - SQL_SERVER
  - MYSQL
  - MYSQL_SLOW_QUERY
  - POSTGRESQL
  - ORACLE_ALERT
  - ORACLE_LISTENER
  - IIS
  - APPLICATION
  - WINDOWS_EVENTS
  - WINDOWS_EVENTS_ACTIVE_DIRECTORY
  - WINDOWS_EVENTS_DNS
  - WINDOWS_EVENTS_IIS
  - WINDOWS_EVENTS_SHAREPOINT
  - SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP
  - SQL_SERVER_FAILOVER_CLUSTER_INSTANCE
  - DEFAULT
  - CUSTOM
  - STEP_FUNCTION
  - API_GATEWAY_ACCESS
  - API_GATEWAY_EXECUTION

  - SAP_HANA_LOGS
  - SAP_HANA_TRACE
  - SAP_HANA_HIGH_AVAILABILITY
  - SAP_NETWEAVER_DEV_TRACE_LOGS
  - PACEMAKER_HIGH_AVAILABILITY
- encoding (en option)

  Type d'encodage des journaux à surveiller. Le codage spécifié doit être inclus dans la liste des codages pris en charge par CloudWatch l'agent. S'il n'est pas fourni, CloudWatch Application Insights utilise le codage par défaut de type utf-8, sauf pour :

  - SQL_SERVER : encodage utf-16
  - IIS : encodage ascii
- surveiller (en option)

  Valeur booléenne qui indique s'il faut surveiller les journaux. La valeur par défaut est `true`.

Processus

Définit un processus à surveiller pour le composant.

JSON

```
{
  "processName" : "monitoredProcessName",
  "alarmMetrics" : [
      list of alarm metrics
  ]
}
```

Propriétés

- processName (obligatoire)

  Le nom du processus à surveiller pour le composant. Le nom du processus ne doit pas contenir de tige de processus, tel que `sqlservr` ou `sqlservr.exe`.
- alarmMetrics (obligatoire)

Une liste des métriques à surveiller pour ce processus. Pour consulter les indicateurs de processus pris en charge par CloudWatch Application Insights, voir Amazon Elastic Compute Cloud (EC2) .

JMX Prometheus Exporter

Définit les paramètres de JMX Prometheus Exporter.

JSON

```
"JMXPrometheusExporter": {
  "jmxURL" : "JMX URL",
  "hostPort" : "The host and port",
  "prometheusPort" : "Target port to emit Prometheus metrics"
}
```

Propriétés

- jmxURL (en option)

  Une URL JMX complète à laquelle se connecter.
- hostPort (en option)

  L'hôte et le port auquel se connecter par le biais du JMX distant. Seul un des `jmxURL` et `hostPort` peut être spécifié.
- prometheusPort (en option)

  Le port cible vers lequel envoyer les métriques Prometheus. S'il n'est pas spécifié, le port par défaut 9404 est utilisé.

Exportateur HANA Prometheus

Définit les paramètres de l'exportateur HANA Prometheus.

JSON

```
"hanaPrometheusExporter": {
    "hanaSid": "SAP HANA  SID",
    "hanaPort": "HANA database port",
    "hanaSecretName": "HANA secret name",
    "prometheusPort": "Target port to emit Prometheus metrics"
```

```
}
```

Propriétés

- Hanasid

  L'ID système SAP (SID) à trois caractères du système SAP HANA.
- Port Hana

  Port de base de données HANA par lequel l'exportateur interrogera les métriques HANA.
- hanaSecretName

  Le AWS Secrets Manager secret qui stocke les informations d'identification des utilisateurs de surveillance HANA. L'exportateur HANA Prometheus utilise ces informations d'identification pour se connecter à la base de données et interroger les métriques HANA.
- prometheusPort (en option)

  Le port cible vers lequel Prometheus envoie des métriques. S'il n'est pas spécifié, le port par défaut 9668 est utilisé.

Exportateur HA Cluster Prometheus

Définit les paramètres de l'exportateur HA Cluster Prometheus.

JSON

```
"haClusterPrometheusExporter": {
    "prometheusPort": "Target port to emit Prometheus metrics"
}
```

Propriétés

- prometheusPort (en option)

  Le port cible vers lequel Prometheus envoie des métriques. S'il n'est pas spécifié, le port par défaut 9664 est utilisé.

NetWeaver Prometheus Exportateur

Définit les paramètres de NetWeaver Prometheus Exporter.

JSON

```
"netWeaverPrometheusExporter": {
    "sapSid": "SAP NetWeaver  SID",
    "instanceNumbers": [ "Array of instance Numbers of SAP NetWeaver system "],
"prometheusPort": "Target port to emit Prometheus metrics"
}
```

Propriétés

- sapSid

  L'identifiant du système SAP (SID) à 3 caractères du NetWeaver système SAP.

- instanceNumbers

  Tableau des numéros d'instance du NetWeaver système SAP.

  Exemple : `"instanceNumbers": [ "00", "01"]`

- prometheusPort (en option)

  Le port cible vers lequel envoyer les métriques Prometheus. S'il n'est pas spécifié, le port par défaut `9680` est utilisé.

SAP ASE Prometheus Exporter

Définit les paramètres de SAP ASE Prometheus Exporter.

JSON

```
"sapASEPrometheusExporter": {
    "sapAseSid": "SAP ASE SID",
    "sapAsePort": "SAP ASE database port",
    "sapAseSecretName": "SAP ASE secret name",
    "prometheusPort": "Target port to emit Prometheus metrics",
    "agreeToEnableASEMonitoring": true
}
```

Propriétés

- sapAseSid

L'ID système SAP (SID) à trois caractères du système SAP ASE.

- sapAsePort

  Le port de base de données SAP ASE par lequel l'exportateur interrogera les métriques ASE.

- sapAseSecretNom

  Le AWS Secrets Manager secret qui stocke les informations d'identification des utilisateurs de surveillance ASE. L'exportateur SAP ASE Prometheus utilise ces informations d'identification pour se connecter à la base de données et interroger les métriques ASE.

- prometheusPort (en option)

  Le port cible vers lequel Prometheus envoie des métriques. S'il n'est pas spécifié, le port par défaut 9399 est utilisé. S'il existe une autre base de données ASE qui utilise le port par défaut, le port 9499 est utilisé.

Événements Windows

Définit les événements Windows à journaliser.

JSON

```
{
  "logGroupName" : "logGroupName",
  "eventName" : "eventName",
  "eventLevels" : ["ERROR","WARNING","CRITICAL","INFORMATION","VERBOSE"],
  "monitor" : true/false
}
```

Propriétés

- logGroupName (obligatoire)

  Le nom du groupe de CloudWatch journaux à associer au journal surveillé. Pour les contraintes liées au nom des groupes de journaux, consultez CreateLogGroup.

- eventName (obligatoire)

  Type des événements Windows à consigner. Ceci est équivalent au nom du canal du journal des événements de Windows. Par exemple, système, sécurité CustomEventName, etc. Ce champ est obligatoire pour chaque type d'événement Windows à consigner.

- eventLevels (obligatoire)

  Niveaux d'événement à consigner. Vous devez spécifier chaque niveau à consigner. Les valeurs possibles incluent `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` et `VERBOSE`. Ce champ est obligatoire pour chaque type d'événement Windows à consigner.
- monitor (en option)

  Valeur booléenne qui indique s'il faut surveiller les journaux. La valeur par défaut est `true`.

alerte

Définit une CloudWatch alarme à surveiller pour le composant.

JSON

```
{
  "alarmName" : "monitoredAlarmName",
  "severity" : HIGH/MEDIUM/LOW
}
```

Propriétés

- alarmName (obligatoire)

  Nom de l' CloudWatch alarme à surveiller pour le composant.
- gravité (en option)

  Indique le degré de panne lorsque l'alerte se déclenche.

Exemples de configuration de composants

Les exemples suivants illustrent des configurations de composants au format JSON pour les services pertinents.

Exemples de configurations de composants

- Table Amazon DynamoDB
- Amazon EC2 Auto Scaling (ASG)
- Cluster Amazon EKS
- Instance Amazon Elastic Compute Cloud (EC2)

- Amazon Elastic Container Service (Amazon ECS)
- Services Amazon ECS
- Tâches Amazon ECS
- Amazon Elastic File System (Amazon EFS)
- Amazon FSx
- Aurora MySQL Amazon Relational Database Service (RDS)
- Instance Amazon Relational Database Service (RDS)
- Surveillance de l'état Amazon Route 53
- Zone hébergée Amazon Route 53
- Amazon Route 53 Resolver point de terminaison
- Amazon Route 53 Resolver configuration de journalisation des requêtes
- Compartiment Amazon S3
- Amazon Simple Queue Service (SQS)
- Rubrique Amazon SNS
- Amazon Virtual Private Cloud (Amazon VPC)
- Passerelles de traduction d'adresses réseau (NAT) Amazon VPC
- Étapes d'API REST API Gateway d'API
- Application Elastic Load Balancing
- Fonction AWS Lambda
- AWS Network Firewall groupe de règles
- AWS Network Firewall association de groupes de règles
- AWS Step Functions
- Instances Amazon EC2 regroupées par le client
- Elastic Load Balancing
- Java
- Kubernetes sur Amazon EC2
- RDS MariaDB et RDS MySQL
- RDS Oracle
- RDS PostgreSQL
- SAP ASE sur Amazon EC2

- Haute disponibilité SAP ASE sur Amazon EC2
- SAP HANA sur Amazon EC2
- Haute disponibilité SAP HANA sur Amazon EC2
- SAP NetWeaver sur Amazon EC2
- NetWeaver Haute disponibilité de SAP sur Amazon EC2
- Groupes de disponibilité SQL Always On
- Instance de cluster de basculement SQL

Table Amazon DynamoDB

L'exemple suivant illustre une configuration de composant au format JSON pour une table Amazon DynamoDB.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "SystemErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "UserErrors",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedReadCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ConsumedWriteCapacityUnits",
      "monitor": false
    },
    {
      "alarmMetricName": "ReadThrottleEvents",
      "monitor": false
    },
    {
      "alarmMetricName": "WriteThrottleEvents",
      "monitor": false
    },
    {
```

```
      "alarmMetricName": "ConditionalCheckFailedRequests",
      "monitor": false
    },
    {
      "alarmMetricName": "TransactionConflict",
      "monitor": false
    }
  ],
  "logs": []
}
```

Amazon EC2 Auto Scaling (ASG)

L'exemple suivant illustre une configuration de composant au format JSON pour Amazon EC2 Auto Scaling (ASG).

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUCreditBalance"
      }, {
        "alarmMetricName" : "EBSIOBalance%"
      }
    ],
    "subComponents" : [
      {
        "subComponentType" : "AWS::EC2::Instance",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "CPUUtilization"
          }, {
            "alarmMetricName" : "StatusCheckFailed"
          }
        ],
        "logs" : [
          {
            "logGroupName" : "my_log_group",
            "logPath" : "C:\\LogFolder\\*",
            "logType" : "APPLICATION"
          }
        ],
        "processes" : [
          {
```

```
            "processName" : "my_process",
            "alarmMetrics" : [
              {
                  "alarmMetricName" : "procstat cpu_usage",
                  "monitor" : true
              }, {
                  "alarmMetricName" : "procstat memory_rss",
                  "monitor" : true
              }
          ]
      }
  ],
        "windowsEvents" : [
          {
            "logGroupName" : "my_log_group_2",
            "eventName" : "Application",
            "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType" : "AWS::EC2::Volume",
        "alarmMetrics" : [
          {
            "alarmMetricName" : "VolumeQueueLength"
          }, {
            "alarmMetricName" : "BurstBalance"
          }
        ]
      }
    ],
    "alarms" : [
      {
        "alarmName" : "my_asg_alarm",
        "severity" : "LOW"
      }
    ]
  }
```

Cluster Amazon EKS

L'exemple suivant illustre une configuration de composant au format JSON pour un cluster Amazon EKS.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName": "cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_filesystem_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "node_memory_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "node_network_total_bytes",
          "monitor":true
       },
       {
          "alarmMetricName": "pod_cpu_reserved_capacity",
          "monitor":true
       },
       {
          "alarmMetricName": "pod_cpu_utilization",
          "monitor":true
       },
       {
          "alarmMetricName": "pod_cpu_utilization_over_pod_limit",
          "monitor":true
       },
       {
```

```
            "alarmMetricName": "pod_memory_reserved_capacity",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_memory_utilization",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_memory_utilization_over_pod_limit",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_network_rx_bytes",
            "monitor":true
         },
         {
            "alarmMetricName": "pod_network_tx_bytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName": "/aws/containerinsights/kubernetes/application",
            "logType":"APPLICATION",
            "monitor":true,
            "encoding":"utf-8"
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::EC2::Instance",
            "alarmMetrics":[
               {
                  "alarmMetricName":"CPUUtilization",
                  "monitor":true
               },
               {
                  "alarmMetricName":"StatusCheckFailed",
                  "monitor":true
               },
               {
                  "alarmMetricName":"disk_used_percent",
                  "monitor":true
               },
```

```
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
               "logPath":"",
               "logType":"APPLICATION",
               "monitor":true,
               "encoding":"utf-8"
            }
         ],
         "processes" : [
           {
               "processName" : "my_process",
               "alarmMetrics" : [
                   {
                       "alarmMetricName" : "procstat cpu_usage",
                       "monitor" : true
                   }, {
                       "alarmMetricName" : "procstat memory_rss",
                       "monitor" : true
                   }
               ]
           }
       ],
         "windowsEvents":[
            {
               "logGroupName":"my_log_group_2",
               "eventName":"Application",
               "eventLevels":[
                  "ERROR",
                  "WARNING",
                  "CRITICAL"
               ],
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::AutoScaling::AutoScalingGroup",
         "alarmMetrics":[
```

```
            {
               "alarmMetricName":"CPUCreditBalance",
               "monitor":true
            },
            {
               "alarmMetricName":"EBSIOBalance%",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
               "alarmMetricName":"VolumeReadBytes",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteBytes",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeReadOps",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeWriteOps",
               "monitor":true
            },
            {
               "alarmMetricName":"VolumeQueueLength",
               "monitor":true
            },
            {
               "alarmMetricName":"BurstBalance",
               "monitor":true
            }
         ]
      }
   ]
}
```

> **Note**
>
> - La section `subComponents` de `AWS::EC2::Instance`, `AWS::EC2::Volume` et de `AWS::AutoScaling::AutoScalingGroup` s'applique uniquement au cluster Amazon EKS s'exécutant sur le type de lancement EC2.
> - La section `windowsEvents` de `AWS::EC2::Instance` dans `subComponents` s'applique uniquement à Windows s'exécutant sur des instances Amazon EC2.

Instance Amazon Elastic Compute Cloud (EC2)

L'exemple suivant illustre une configuration de composant au format JSON pour une instance Amazon EC2.

> **Important**
>
> Lorsqu'une instance Amazon EC2 entre dans l'état `stopped`, elle est retirée de la surveillance. Lorsqu'il revient à un `running` état, il est ajouté à la liste des composants non surveillés sur la page des détails de l'application de la console CloudWatch Application Insights. Si la surveillance automatique des nouvelles ressources est activée pour l'application, l'instance est ajoutée à la liste des Monitored components (Composants surveillés). Toutefois, les journaux et les métriques sont définis sur la valeur par défaut de la charge de travail. La configuration précédente du journal et des métriques n'est pas enregistrée.

```
{
    "alarmMetrics" : [
      {
        "alarmMetricName" : "CPUUtilization",
        "monitor" : true
      }, {
        "alarmMetricName" : "StatusCheckFailed"
      }
    ],
    "logs" : [
      {
        "logGroupName" : "my_log_group",
        "logPath" : "C:\\LogFolder\\*",
```

```
        "logType" : "APPLICATION",
        "monitor" : true
      },
      {
        "logGroupName" : "my_log_group_2",
        "logPath" : "C:\\LogFolder2\\*",
        "logType" : "IIS",
        "encoding" : "utf-8"
      }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ],
    "windowsEvents" : [
      {
        "logGroupName" : "my_log_group_3",
        "eventName" : "Application",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
      }, {
        "logGroupName" : "my_log_group_4",
        "eventName" : "System",
        "eventLevels" : [ "ERROR", "WARNING", "CRITICAL" ],
        "monitor" : true
    }],
     "alarms" : [
      {
        "alarmName" : "my_instance_alarm_1",
        "severity" : "HIGH"
      },
      {
        "alarmName" : "my_instance_alarm_2",
        "severity" : "LOW"
```

```
      }
    ],
     "subComponents" : [
     {
       "subComponentType" : "AWS::EC2::Volume",
       "alarmMetrics" : [
       {
         "alarmMetricName" : "VolumeQueueLength",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "VolumeThroughputPercentage",
         "monitor" : "true"
       },
       {
         "alarmMetricName" : "BurstBalance",
         "monitor" : "true"
       }
    }]
  }
```

Amazon Elastic Container Service (Amazon ECS)

L'exemple suivant illustre une configuration de composant au format JSON pour Amazon Elastic Container Service (Amazon ECS).

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CpuUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilized",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkRxBytes",
          "monitor":true
       },
       {
          "alarmMetricName":"NetworkTxBytes",
          "monitor":true
```

```
        },
        {
           "alarmMetricName":"RunningTaskCount",
           "monitor":true
        },
        {
           "alarmMetricName":"PendingTaskCount",
           "monitor":true
        },
        {
           "alarmMetricName":"StorageReadBytes",
           "monitor":true
        },
        {
           "alarmMetricName":"StorageWriteBytes",
           "monitor":true
        }
     ],
     "logs":[
        {
           "logGroupName":"/ecs/my-task-definition",
           "logType":"APPLICATION",
           "monitor":true
        }
     ],
     "subComponents":[
        {
           "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
           "alarmMetrics":[
              {
                 "alarmMetricName":"HTTPCode_Backend_4XX",
                 "monitor":true
              },
              {
                 "alarmMetricName":"HTTPCode_Backend_5XX",
                 "monitor":true
              },
              {
                 "alarmMetricName":"Latency",
                 "monitor":true
              },
              {
                 "alarmMetricName":"SurgeQueueLength",
                 "monitor":true
```

```
            },
            {
               "alarmMetricName":"UnHealthyHostCount",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
         "alarmMetrics":[
            {
               "alarmMetricName":"HTTPCode_Target_4XX_Count",
               "monitor":true
            },
            {
               "alarmMetricName":"HTTPCode_Target_5XX_Count",
               "monitor":true
            },
            {
               "alarmMetricName":"TargetResponseTime",
               "monitor":true
            },
            {
               "alarmMetricName":"UnHealthyHostCount",
               "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Instance",
         "alarmMetrics":[
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
```

```
                "alarmMetricName":"mem_used_percent",
                "monitor":true
            }
         ],
         "logs":[
            {
                "logGroupName":"my_log_group",
                "logPath":"/mylog/path",
                "logType":"APPLICATION",
                "monitor":true
            }
         ],
         "processes" : [
            {
                "processName" : "my_process",
                "alarmMetrics" : [
                    {
                        "alarmMetricName" : "procstat cpu_usage",
                        "monitor" : true
                    }, {
                        "alarmMetricName" : "procstat memory_rss",
                        "monitor" : true
                    }
                ]
            }
         ],
         "windowsEvents":[
            {
                "logGroupName":"my_log_group_2",
                "eventName":"Application",
                "eventLevels":[
                   "ERROR",
                   "WARNING",
                   "CRITICAL"
                ],
                "monitor":true
            }
         ]
      },
      {
         "subComponentType":"AWS::EC2::Volume",
         "alarmMetrics":[
            {
                "alarmMetricName":"VolumeQueueLength",
```

```
                "monitor":"true"
            },
            {
                "alarmMetricName":"VolumeThroughputPercentage",
                "monitor":"true"
            },
            {
                "alarmMetricName":"BurstBalance",
                "monitor":"true"
            }
        ]
    }
  ]
}
```

> **Note**
>
> - La section `subComponents` de `AWS::EC2::Instance` et de `AWS::EC2::Volume` s'applique uniquement aux clusters Amazon ECS avec un service ECS ou une tâche ECS s'exécutant sur le type de lancement EC2.
> - La section `windowsEvents` de `AWS::EC2::Instance` dans `subComponents` s'applique uniquement à Windows s'exécutant sur des instances Amazon EC2.

Services Amazon ECS

Les exemples suivants illustrent la configuration d'un composant au format JSON pour un service Amazon ECS service.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"CPUUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"MemoryUtilization",
          "monitor":true
       },
       {
          "alarmMetricName":"CpuUtilized",
```

```
            "monitor":true
         },
         {
            "alarmMetricName":"MemoryUtilized",
            "monitor":true
         },
         {
            "alarmMetricName":"NetworkRxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"NetworkTxBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"RunningTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"PendingTaskCount",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageReadBytes",
            "monitor":true
         },
         {
            "alarmMetricName":"StorageWriteBytes",
            "monitor":true
         }
      ],
      "logs":[
         {
            "logGroupName":"/ecs/my-task-definition",
            "logType":"APPLICATION",
            "monitor":true
         }
      ],
      "subComponents":[
         {
            "subComponentType":"AWS::ElasticLoadBalancing::LoadBalancer",
            "alarmMetrics":[
               {
                  "alarmMetricName":"HTTPCode_Backend_4XX",
```

```
                "monitor":true
            },
            {
                "alarmMetricName":"HTTPCode_Backend_5XX",
                "monitor":true
            },
            {
                "alarmMetricName":"Latency",
                "monitor":true
            },
            {
                "alarmMetricName":"SurgeQueueLength",
                "monitor":true
            },
            {
                "alarmMetricName":"UnHealthyHostCount",
                "monitor":true
            }
        ]
    },
    {
        "subComponentType":"AWS::ElasticLoadBalancingV2::LoadBalancer",
        "alarmMetrics":[
            {
                "alarmMetricName":"HTTPCode_Target_4XX_Count",
                "monitor":true
            },
            {
                "alarmMetricName":"HTTPCode_Target_5XX_Count",
                "monitor":true
            },
            {
                "alarmMetricName":"TargetResponseTime",
                "monitor":true
            },
            {
                "alarmMetricName":"UnHealthyHostCount",
                "monitor":true
            }
        ]
    },
    {
        "subComponentType":"AWS::EC2::Instance",
        "alarmMetrics":[
```

```
            {
               "alarmMetricName":"CPUUtilization",
               "monitor":true
            },
            {
               "alarmMetricName":"StatusCheckFailed",
               "monitor":true
            },
            {
               "alarmMetricName":"disk_used_percent",
               "monitor":true
            },
            {
               "alarmMetricName":"mem_used_percent",
               "monitor":true
            }
         ],
         "logs":[
            {
               "logGroupName":"my_log_group",
               "logPath":"/mylog/path",
               "logType":"APPLICATION",
               "monitor":true
            }
         ],
         "processes" : [
            {
               "processName" : "my_process",
               "alarmMetrics" : [
               {
                   "alarmMetricName" : "procstat cpu_usage",
                   "monitor" : true
               }, {
                   "alarmMetricName" : "procstat memory_rss",
                   "monitor" : true
               }
           ]
       }
   ],
         "windowsEvents":[
            {
               "logGroupName":"my_log_group_2",
               "eventName":"Application",
               "eventLevels":[
```

```
                    "ERROR",
                    "WARNING",
                    "CRITICAL"
                ],
                "monitor":true
            }
        ]
    },
    {
        "subComponentType":"AWS::EC2::Volume",
        "alarmMetrics":[
            {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":"true"
            },
            {
                "alarmMetricName":"VolumeThroughputPercentage",
                "monitor":"true"
            },
            {
                "alarmMetricName":"BurstBalance",
                "monitor":"true"
            }
        ]
    }
  ]
}
```

> **Note**
>
> - La section `subComponents` de `AWS::EC2::Instance` et de `AWS::EC2::Volume` s'applique uniquement aux clusters Amazon ECS s'exécutant sur le type de lancement EC2.
> - La section `windowsEvents` de `AWS::EC2::Instance` dans `subComponents` s'applique uniquement à Windows s'exécutant sur des instances Amazon EC2.

Tâches Amazon ECS

Les exemples suivants illustrent une configuration de composant au format JSON pour une tâche Amazon ECS.

```
{
    "logs":[
       {
          "logGroupName":"/ecs/my-task-definition",
          "logType":"APPLICATION",
          "monitor":true
       }
    ],
    "processes" : [
        {
            "processName" : "my_process",
            "alarmMetrics" : [
                {
                    "alarmMetricName" : "procstat cpu_usage",
                    "monitor" : true
                }, {
                    "alarmMetricName" : "procstat memory_rss",
                    "monitor" : true
                }
            ]
        }
    ]
 }
```

Amazon Elastic File System (Amazon EFS)

L'exemple suivant illustre une configuration de composant au format JSON pour Amazon EFS.

```
{
   "alarmMetrics": [
     {
       "alarmMetricName": "BurstCreditBalance",
       "monitor": true
     },
     {
       "alarmMetricName": "PercentIOLimit",
       "monitor": true
     },
     {
       "alarmMetricName": "PermittedThroughput",
       "monitor": true
     },
     {
```

```
        "alarmMetricName": "MeteredIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "TotalIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "DataWriteIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "DataReadIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "MetadataIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "ClientConnections",
        "monitor": true
      },
      {
        "alarmMetricName": "TimeSinceLastSync",
        "monitor": true
      },
      {
        "alarmMetricName": "Throughput",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfPermittedThroughputUtilization",
        "monitor": true
      },
      {
        "alarmMetricName": "ThroughputIOPS",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentThroughputDataReadIOBytes",
        "monitor": true
      },
      {
```

```
        "alarmMetricName": "PercentThroughputDataWriteIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfIOPSDataReadIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "PercentageOfIOPSDataWriteIOBytes",
        "monitor": true
      },
      {
        "alarmMetricName": "AverageDataReadIOBytesSize",
        "monitor": true
      },
      {
        "alarmMetricName": "AverageDataWriteIOBytesSize",
        "monitor": true
      }
    ],
    "logs": [
     {
     "logGroupName": "/aws/efs/utils",
     "logType": "EFS_MOUNT_STATUS",
     "monitor": true,
     }
    ]
 }
```

Amazon FSx

L'exemple suivant illustre une configuration de composant au format JSON pour Amazon FSx.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DataReadBytes",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteBytes",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "DataReadOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "DataWriteOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "MetadataOperations",
      "monitor": true
    },
    {
      "alarmMetricName": "FreeStorageCapacity",
      "monitor": true
    }
  ]
}
```

Aurora MySQL Amazon Relational Database Service (RDS)

L'exemple suivant illustre une configuration de composant au format JSON pour Amazon RDS Aurora MySQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    },
    {
      "alarmMetricName": "CommitLatency",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
```

```
  ]
}
```

Instance Amazon Relational Database Service (RDS)

L'exemple suivant illustre une configuration de composant au format JSON pour une instance Amazon RDS.

```
{
  "alarmMetrics" : [
    {
      "alarmMetricName" : "BurstBalance",
      "monitor" : true
    }, {
      "alarmMetricName" : "WriteThroughput",
      "monitor" : false
    }
  ],

  "alarms" : [
    {
      "alarmName" : "my_rds_instance_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Surveillance de l'état Amazon Route 53

L'exemple suivant illustre une configuration de composant au format JSON pour une surveillance de l'état Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ChildHealthCheckHealthyCount",
      "monitor": true
    },
    {
      "alarmMetricName": "ConnectionTime",
      "monitor": true
    },
    {
```

```
      "alarmMetricName": "HealthCheckPercentageHealthy",
      "monitor": true
    },
    {
      "alarmMetricName": "HealthCheckStatus",
      "monitor": true
    },
    {
      "alarmMetricName": "SSLHandshakeTime",
      "monitor": true
    },
    {
      "alarmMetricName": "TimeToFirstByte",
      "monitor": true
    }
  ]
}
```

Zone hébergée Amazon Route 53

L'exemple suivant illustre une configuration de composant au format JSON pour une zone hébergée Amazon Route 53.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "DNSQueries",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECInternalFailure",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeysNeedingAction",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyMaxNeedingActionAge",
      "monitor": true
    },
    {
      "alarmMetricName": "DNSSECKeySigningKeyAge",
```

```
      "monitor": true
    }
  ],
  "logs": [
    {
      "logGroupName":"/hosted-zone/logs",
      "logType": "ROUTE53_DNS_PUBLIC_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Amazon Route 53 Resolver point de terminaison

L'exemple suivant montre une configuration de composant au format JSON pour le Amazon Route 53 Resolver point de terminaison.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "EndpointHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "EndpointUnHealthyENICount",
      "monitor": true
    },
    {
      "alarmMetricName": "InboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryVolume",
      "monitor": true
    },
    {
      "alarmMetricName": "OutboundQueryAggregateVolume",
      "monitor": true
    }
  ]
}
```

Amazon Route 53 Resolver configuration de journalisation des requêtes

L'exemple suivant illustre la configuration d'un composant au format JSON pour la configuration de journalisation de requêtes Amazon Route 53 Resolver .

```
{
  "logs": [
    {
      "logGroupName": "/resolver-query-log-config/logs",
      "logType": "ROUTE53_RESOLVER_QUERY_LOGS",
      "monitor": true
    }
  ]
}
```

Compartiment Amazon S3

L'exemple suivant illustre une configuration de composant au format JSON pour le compartiment Amazon S3.

```
{
    "alarmMetrics" : [
        {
            "alarmMetricName" : "ReplicationLatency",
            "monitor" : true
        }, {
            "alarmMetricName" : "5xxErrors",
            "monitor" : true
        }, {
            "alarmMetricName" : "BytesDownloaded"
            "monitor" : true
        }
    ]
}
```

Amazon Simple Queue Service (SQS)

L'exemple suivant illustre une configuration de composants au format JSON pour Amazon Simple Queue Service.

```
{
  "alarmMetrics" : [
```

```
    {
      "alarmMetricName" : "ApproximateAgeOfOldestMessage"
    }, {
      "alarmMetricName" : "NumberOfEmptyReceives"
    }
  ],
  "alarms" : [
    {
      "alarmName" : "my_sqs_alarm",
      "severity" : "MEDIUM"
    }
  ]
}
```

Rubrique Amazon SNS

L'exemple suivant illustre une configuration de composant au format JSON pour la rubrique Amazon SNS.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NumberOfNotificationsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-InvalidAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFilteredOut-NoMessageAttributes",
      "monitor": true
    },
    {
      "alarmMetricName": "NumberOfNotificationsFailedToRedriveToDlq",
      "monitor": true
    }
  ]
}
```

Amazon Virtual Private Cloud (Amazon VPC)

L'exemple suivant illustre une configuration de composant au format JSON pour Amazon VPC.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "NetworkAddressUsage",
      "monitor": true
    },
    {
      "alarmMetricName": "NetworkAddressUsagePeered",
      "monitor": true
    },
    {
      "alarmMetricName": "VPCFirewallQueryVolume",
      "monitor": true
    }
  ]
}
```

Passerelles de traduction d'adresses réseau (NAT) Amazon VPC

L'exemple suivant illustre une configuration de composant au format JSON pour les passerelles NAT.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ErrorPortAllocation",
      "monitor": true
    },
    {
      "alarmMetricName": "IdleTimeoutCount",
      "monitor": true
    }
  ]
}
```

Étapes d'API REST API Gateway d'API

L'exemple suivant illustre une configuration de composant au format JSON pour les étapes de l'API REST Gateway d'API.

```
{
     "alarmMetrics" : [
         {
```

```
            "alarmMetricName" : "4XXError",
            "monitor" : true
        },
        {
            "alarmMetricName" : "5XXError",
            "monitor" : true
        }
    ],
    "logs" : [
        {
            "logType" : "API_GATEWAY_EXECUTION",
            "monitor" : true
        },
        {
            "logType" : "API_GATEWAY_ACCESS",
            "monitor" : true
        }
    ]
}
```

Application Elastic Load Balancing

L'exemple suivant illustre une configuration de composant au format JSON pour Application Elastic Load Balancing.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ActiveConnectionCount",
    }, {
      "alarmMetricName": "TargetResponseTime"
    }
  ],
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
        }, {
          "alarmMetricName": "StatusCheckFailed"
        }
      ],
```

```
      "logs": [
        {
          "logGroupName": "my_log_group",
          "logPath": "C:\\LogFolder\\*",
          "logType": "APPLICATION",
        }
      ],
      "windowsEvents": [
        {
          "logGroupName": "my_log_group_2",
          "eventName": "Application",
          "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
        }
      ]
    }, {
      "subComponentType": "AWS::EC2::Volume",
      "alarmMetrics": [
        {
          "alarmMetricName": "VolumeQueueLength",
        }, {
          "alarmMetricName": "BurstBalance"
        }
      ]
    }
  ],

  "alarms": [
    {
      "alarmName": "my_alb_alarm",
      "severity": "LOW"
    }
  ]
}
```

Fonction AWS Lambda

L'exemple suivant illustre une configuration de composant au format JSON pour AWS Lambda .

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "Errors",
      "monitor": true
    },
```

```
    {
      "alarmMetricName": "Throttles",
      "monitor": true
    },
    {
      "alarmMetricName": "IteratorAge",
      "monitor": true
    },
    {
      "alarmMetricName": "Duration",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "DEFAULT",
      "monitor": true
    }
  ]
}
```

AWS Network Firewall groupe de règles

L'exemple suivant illustre une configuration de composant au format JSON pour un groupe de règles AWS Network Firewall .

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Network Firewall association de groupes de règles

L'exemple suivant illustre une configuration de composant au format JSON pour une association de groupes de règles AWS Network Firewall .

```
{
  "alarmMetrics": [
    {
```

```
      "alarmMetricName": "FirewallRuleGroupQueryVolume",
      "monitor": true
    }
  ]
}
```

AWS Step Functions

L'exemple suivant illustre la configuration d'un composant au format JSON pour AWS Step Functions.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "ExecutionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "LambdaFunctionsFailed",
      "monitor": true
    },
    {
      "alarmMetricName": "ProvisionedRefillRate",
      "monitor": true
    }
  ],
  "logs": [
    {
     "logGroupName": "/aws/states/HelloWorld-Logs",
      "logType": "STEP_FUNCTION",
      "monitor": true,
    }
  ]
}
```

Instances Amazon EC2 regroupées par le client

L'exemple suivant illustre une configuration de composant au format JSON pour les instances Amazon EC2 regroupées par le client.

```
{
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
```

```
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization",
          },
          {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION",
          }
        ],
        "processes": [
            {
                "processName": "my_process",
                "alarmMetrics": [
                    {
                        "alarmMetricName": "procstat cpu_usage",
                        "monitor": true
                    }, {
                        "alarmMetricName": "procstat memory_rss",
                        "monitor": true
                    }
                ]
            }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ]
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength",
          }, {
            "alarmMetricName": "BurstBalance"
          }
```

```
        ]
      }
    ],
    "alarms": [
      {
        "alarmName": "my_alarm",
        "severity": "MEDIUM"
      }
    ]
  }
```

Elastic Load Balancing

L'exemple suivant illustre une configuration de composant au format JSON pour Elastic Load Balancing.

```
{
    "alarmMetrics": [
      {
        "alarmMetricName": "EstimatedALBActiveConnectionCount"
      }, {
        "alarmMetricName": "HTTPCode_Backend_5XX"
      }
    ],
    "subComponents": [
      {
        "subComponentType": "AWS::EC2::Instance",
        "alarmMetrics": [
          {
            "alarmMetricName": "CPUUtilization"
          }, {
            "alarmMetricName": "StatusCheckFailed"
          }
        ],
        "logs": [
          {
            "logGroupName": "my_log_group",
            "logPath": "C:\\LogFolder\\*",
            "logType": "APPLICATION"
          }
        ],
        "processes": [
          {
```

```
            "processName": "my_process",
            "alarmMetrics": [
                {
                  "alarmMetricName": "procstat cpu_usage",
                  "monitor": true
                }, {
                  "alarmMetricName": "procstat memory_rss",
                  "monitor": true
                }
              ]
          }
        ],
        "windowsEvents": [
          {
            "logGroupName": "my_log_group_2",
            "eventName": "Application",
            "eventLevels": [ "ERROR", "WARNING", "CRITICAL" ],
            "monitor": true
          }
        ]
      }, {
        "subComponentType": "AWS::EC2::Volume",
        "alarmMetrics": [
          {
            "alarmMetricName": "VolumeQueueLength"
          }, {
            "alarmMetricName": "BurstBalance"
          }
        ]
      }
    ],

    "alarms": [
      {
        "alarmName": "my_elb_alarm",
        "severity": "HIGH"
      }
    ]
  }
```

Java

L'exemple suivant illustre une configuration de composant au format JSON pour Java.

```
{
  "alarmMetrics": [ {
    "alarmMetricName": "java_lang_threading_threadcount",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_used",
    "monitor": true
  },
  {
    "alarmMetricName": "java_lang_memory_heapmemoryusage_committed",
    "monitor": true
  }],
  "logs": [ ],
  "JMXPrometheusExporter": {
      "hostPort": "8686",
      "prometheusPort": "9404"
  }
}
```

> **Note**
>
> Application Insights ne prend pas en charge la configuration de l'authentification pour Prometheus JMX Exporter. Pour savoir comment configurer l'authentification, consultez l'exemple de configuration de Prometheus JMX Exporter.

Kubernetes sur Amazon EC2

L'exemple suivant illustre une configuration de composant au format JSON pour Kubernetes sur Amazon EC2.

```
{
    "alarmMetrics":[
       {
          "alarmMetricName":"cluster_failed_node_count",
          "monitor":true
       },
       {
          "alarmMetricName":"node_cpu_reserved_capacity",
          "monitor":true
       },
```

```
        {
           "alarmMetricName":"node_cpu_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"node_filesystem_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"node_memory_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName":"node_memory_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"node_network_total_bytes",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_cpu_utilization_over_pod_limit",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_reserved_capacity",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_utilization",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_memory_utilization_over_pod_limit",
           "monitor":true
        },
```

```
        {
           "alarmMetricName":"pod_network_rx_bytes",
           "monitor":true
        },
        {
           "alarmMetricName":"pod_network_tx_bytes",
           "monitor":true
        }
     ],
     "logs":[
        {
           "logGroupName":"/aws/containerinsights/kubernetes/application",
           "logType":"APPLICATION",
           "monitor":true,
           "encoding":"utf-8"
        }
     ],
     "subComponents":[
        {
           "subComponentType":"AWS::EC2::Instance",
           "alarmMetrics":[
              {
                 "alarmMetricName":"CPUUtilization",
                 "monitor":true
              },
              {
                 "alarmMetricName":"StatusCheckFailed",
                 "monitor":true
              },
              {
                 "alarmMetricName":"disk_used_percent",
                 "monitor":true
              },
              {
                 "alarmMetricName":"mem_used_percent",
                 "monitor":true
              }
           ],
           "logs":[
              {
                 "logGroupName":"APPLICATION-KubernetesClusterOnEC2-IAD",
                 "logPath":"",
                 "logType":"APPLICATION",
                 "monitor":true,
```

```
                "encoding":"utf-8"
             }
          ],
          "processes" : [
            {
                "processName" : "my_process",
                "alarmMetrics" : [
                    {
                        "alarmMetricName" : "procstat cpu_usage",
                        "monitor" : true
                    }, {
                        "alarmMetricName" : "procstat memory_rss",
                        "monitor" : true
                    }
                ]
            }
        ]
       },
       {
          "subComponentType":"AWS::EC2::Volume",
          "alarmMetrics":[
             {
                "alarmMetricName":"VolumeReadBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteBytes",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeReadOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeWriteOps",
                "monitor":true
             },
             {
                "alarmMetricName":"VolumeQueueLength",
                "monitor":true
             },
             {
                "alarmMetricName":"BurstBalance",
                "monitor":true
```

```
            }
          ]
       }
    ]
 }
```

RDS MariaDB et RDS MySQL

L'exemple suivant illustre une configuration de composants au format JSON pour RDS MySQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "MYSQL",
      "monitor": true,
    },
    {
      "logType": "MYSQL_SLOW_QUERY",
      "monitor": false
    }
  ]
}
```

RDS Oracle

L'exemple suivant illustre une configuration de composant au format JSON pour RDS Oracle.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "ORACLE_ALERT",
```

```
      "monitor": true,
    },
    {
      "logType": "ORACLE_LISTENER",
      "monitor": false
    }
  ]
}
```

RDS PostgreSQL

L'exemple suivant illustre une configuration de composant au format JSON pour RDS PostgreSQL.

```
{
  "alarmMetrics": [
    {
      "alarmMetricName": "CPUUtilization",
      "monitor": true
    }
  ],
  "logs": [
    {
      "logType": "POSTGRESQL",
      "monitor": true
    }
  ]
}
```

SAP ASE sur Amazon EC2

L'exemple suivant illustre une configuration de composant au format JSON pour SAP ASE sur Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
```

```
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_db_log_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_locked_login",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_data_cache_hit_ratio",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
```

```
          "monitor": true,
          "encoding": "utf-8"
        }
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

Haute disponibilité SAP ASE sur Amazon EC2

L'exemple suivant illustre une configuration de composant au format JSON pour la haute disponibilité SAP ASE sur Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "asedb_database_availability",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_trunc_log_on_chkpt_enabled",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_db_backup_age_in_days",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_last_transaction_log_backup_age_in_hours",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_suspected_database",
          "monitor": true
        },
        {
```

```
          "alarmMetricName": "asedb_db_space_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_mode",
          "monitor": true
        },
        {
          "alarmMetricName": "asedb_ha_replication_latency_in_minutes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_ASE_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2.log",
          "logType": "SAP_ASE_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_BACKUP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/ASE-*/install/SY2_BS.log",
          "logType": "SAP_ASE_BACKUP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_REP_SERVER_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/repservername/repservername.log",
          "logType": "SAP_ASE_REP_SERVER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_ASE_RMA_AGENT_LOGS-my-resource-group",
          "logPath": "/sybase/SY2/DM/RMA-*/instances/AgentContainer/logs/",
          "logType": "SAP_ASE_RMA_AGENT_LOGS",
          "monitor": true,
          "encoding": "utf-8"
```

```
        },
        {
          "logGroupName": "SAP_ASE_FAULT_MANAGER_LOGS-my-resource-group",
          "logPath": "/opt/sap/FaultManager/dev_sybdbfm",
          "logType": "SAP_ASE_FAULT_MANAGER_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
  ],
  "sapAsePrometheusExporter": {
    "sapAseSid": "ASE",
    "sapAsePort": "4901",
    "sapAseSecretName": "ASE_DB_CREDS",
    "prometheusPort": "9399",
    "agreeToEnableASEMonitoring": true
}
```

SAP HANA sur Amazon EC2

L'exemple suivant illustre une configuration de composant au format JSON pour SAP HANA sur Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "hanadb_max_trigger_read_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_table_allocation_limit_used_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_cpu_usage_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_plan_cache_hit_ratio_percent",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_last_data_backup_age_days",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
          "logGroupName": "SAP_HANA_LOGS-my-resource-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.log",
          "logType": "SAP_HANA_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
```

```
  }
}
```

Haute disponibilité SAP HANA sur Amazon EC2

L'exemple suivant illustre une configuration de composant au format JSON pour SAP HANA High Availability on Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "hanadb_server_startup_time_variations_seconds",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_5_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_level_4_alerts_count",
          "monitor": true
        },
        {
          "alarmMetricName": "hanadb_out_of_memory_events_count",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_stonith_enabled",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_HANA_TRACE-my-resourge-group",
          "logPath": "/usr/sap/HDB/HDB00/*/trace/*.trc",
          "logType": "SAP_HANA_TRACE",
          "monitor": true,
          "encoding": "utf-8"
        },
        {
```

```
          "logGroupName": "SAP_HANA_HIGH_AVAILABILITY-my-resource-group",
          "logPath": "/var/log/pacemaker/pacemaker.log",
          "logType": "SAP_HANA_HIGH_AVAILABILITY",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "hanaPrometheusExporter": {
    "hanaSid": "HDB",
    "hanaPort": "30013",
    "hanaSecretName": "HANA_DB_CREDS",
    "prometheusPort": "9668"
  },
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  }
}
```

SAP NetWeaver sur Amazon EC2

L'exemple suivant montre une configuration de composant au format JSON pour SAP NetWeaver sur Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "CPUUtilization",
          "monitor": true
        },
        {
          "alarmMetricName": "StatusCheckFailed",
          "monitor": true
        },
        {
          "alarmMetricName": "disk_used_percent",
          "monitor": true
        },
        {
```

```
            "alarmMetricName": "mem_used_percent",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_ResponseTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_ResponseTimeDialog",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_DBRequestTime",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_LongRunners",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_AbortedJobs",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_BasisSystem",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Database",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_Security",
            "monitor": true
          },
          {
            "alarmMetricName": "sap_alerts_System",
            "monitor": true
          },
          {
```

```
          "alarmMetricName": "sap_alerts_QueueTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Availability",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_dispatcher_queue_max",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_max",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_now",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_locks_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-ML4",
          "logPath": "/usr/sap/ML4/*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
```

```
      ]
    }
  ],
  "netWeaverPrometheusExporter": {
    "sapSid": "ML4",
    "instanceNumbers": [
      "00",
      "11"
    ],
    "prometheusPort": "9680"
  }
}
```

NetWeaver Haute disponibilité de SAP sur Amazon EC2

L'exemple suivant montre une configuration de composant au format JSON pour SAP NetWeaver High Availability sur Amazon EC2.

```
{
  "subComponents": [
    {
      "subComponentType": "AWS::EC2::Instance",
      "alarmMetrics": [
        {
          "alarmMetricName": "ha_cluster_corosync_ring_errors",
          "monitor": true
        },
        {
          "alarmMetricName": "ha_cluster_pacemaker_fail_count",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_check_failover_config_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_HA_get_failover_config_HAActive",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_AbortedJobs",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "sap_alerts_Availability",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_BasisSystem",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_DBRequestTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Database",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_FrontendResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_LongRunners",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_QueueTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTime",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialog",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_ResponseTimeDialogRFC",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_Security",
          "monitor": true
        },
```

```
        {
          "alarmMetricName": "sap_alerts_Shortdumps",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_SqlError",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_alerts_System",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_enqueue_server_replication_state",
          "monitor": true
        },
        {
          "alarmMetricName": "sap_start_service_processes",
          "monitor": true
        }
      ],
      "logs": [
        {
          "logGroupName": "SAP_NETWEAVER_DEV_TRACE_LOGS-NetWeaver-PR1",
          "logPath": "/usr/sap/<SID>/D*/work/dev_w*",
          "logType": "SAP_NETWEAVER_DEV_TRACE_LOGS",
          "monitor": true,
          "encoding": "utf-8"
        }
      ]
    }
  ],
  "haClusterPrometheusExporter": {
    "prometheusPort": "9664"
  },
  "netWeaverPrometheusExporter": {
    "sapSid": "PR1",
    "instanceNumbers": [
      "11",
      "12"
    ],
    "prometheusPort": "9680"
  }
```

```
}
```

Groupes de disponibilité SQL Always On

L'exemple suivant illustre une configuration de composant au format JSON pour SQL Always On Availability Group.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Buffer cache hit ratio",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Buffer Manager Page life expectancy",
      "monitor" : true
```

```
    }, {
      "alarmMetricName" : "SQLServer:General Statistics Processes blocked",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:General Statistics User Connections",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Locks Number of Deadlocks/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:SQL Statistics Batch Requests/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica File Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Bytes Received/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log remaining for undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Log Send Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Mirrored Write Transaction/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Recovery Queue",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redo Bytes Remaining",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Redone Bytes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Total Log requiring undo",
      "monitor" : true
    }, {
      "alarmMetricName" : "SQLServer:Database Replica Transaction Delay",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
```

```
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_ALWAYSON_AVAILABILITY_GROUP-<RESOURCE_GROUP_NAME>",
      "logPath" : "C:\\Program Files\\Microsoft SQL Server\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
```

```
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Instance de cluster de basculement SQL

L'exemple suivant illustre une configuration de composant au format JSON pour une instance de cluster de basculement SQL.

```
{
  "subComponents" : [ {
    "subComponentType" : "AWS::EC2::Instance",
    "alarmMetrics" : [ {
      "alarmMetricName" : "CPUUtilization",
      "monitor" : true
    }, {
      "alarmMetricName" : "StatusCheckFailed",
      "monitor" : true
    }, {
      "alarmMetricName" : "Processor % Processor Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory % Committed Bytes In Use",
      "monitor" : true
    }, {
      "alarmMetricName" : "Memory Available Mbytes",
      "monitor" : true
    }, {
      "alarmMetricName" : "Paging File % Usage",
      "monitor" : true
    }, {
      "alarmMetricName" : "System Processor Queue Length",
      "monitor" : true
    }, {
      "alarmMetricName" : "Network Interface Bytes Total/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "PhysicalDisk % Disk Time",
      "monitor" : true
    }, {
      "alarmMetricName" : "Bytes Received/sec",
```

```
      "monitor" : true
    }, {
      "alarmMetricName" : "Normal Messages Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Urgent Message Queue Length/se",
      "monitor" : true
    }, {
      "alarmMetricName" : "Reconnect Count",
      "monitor" : true
    }, {
      "alarmMetricName" : "Unacknowledged Message Queue Length/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Outstanding",
      "monitor" : true
    }, {
      "alarmMetricName" : "Messages Sent/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Database Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Update Messages/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Flushes/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Crypto Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Restored/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Registry Checkpoints Saved/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster API Calls/sec",
      "monitor" : true
    }, {
```

```
      "alarmMetricName" : "Resource API Calls/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Cluster Handles/sec",
      "monitor" : true
    }, {
      "alarmMetricName" : "Resource Handles/sec",
      "monitor" : true
    } ],
    "windowsEvents" : [ {
      "logGroupName" : "WINDOWS_EVENTS-Application-<RESOURCE_GROUP_NAME>",
      "eventName" : "Application",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL"],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-System-<RESOURCE_GROUP_NAME>",
      "eventName" : "System",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL", "INFORMATION" ],
      "monitor" : true
    }, {
      "logGroupName" : "WINDOWS_EVENTS-Security-<RESOURCE_GROUP_NAME>",
      "eventName" : "Security",
      "eventLevels" : [ "WARNING", "ERROR", "CRITICAL" ],
      "monitor" : true
    } ],
    "logs" : [ {
      "logGroupName" : "SQL_SERVER_FAILOVER_CLUSTER_INSTANCE-<RESOURCE_GROUP_NAME>",
      "logPath" : "\\\\amznfsxjmzbykwn.mydomain.aws\\SQLDB\\MSSQL**.MSSQLSERVER\\MSSQL\
\Log\\ERRORLOG",
      "logType" : "SQL_SERVER",
      "monitor" : true,
      "encoding" : "utf-8"
    } ]
  }, {
    "subComponentType" : "AWS::EC2::Volume",
    "alarmMetrics" : [ {
      "alarmMetricName" : "VolumeReadBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeWriteBytes",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeReadOps",
      "monitor" : true
```

```
    }, {
    "alarmMetricName" : "VolumeWriteOps",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeQueueLength",
      "monitor" : true
    }, {
    "alarmMetricName" : "VolumeThroughputPercentage",
      "monitor" : true
    }, {
    "alarmMetricName" : "BurstBalance",
      "monitor" : true
    } ]
  } ]
}
```

Création et configuration de la surveillance CloudWatch d'Application Insights à l'aide CloudFormation de modèles

Vous pouvez ajouter la surveillance Application Insights, y compris les indicateurs clés et la télémétrie, à votre application, à votre base de données et à votre serveur Web, directement à partir de AWS CloudFormation modèles.

Cette section fournit des exemples de AWS CloudFormation modèles aux formats JSON et YAML pour vous aider à créer et à configurer la surveillance d'Application Insights.

Pour consulter la référence aux ressources et aux propriétés d'Application Insights dans le guide de AWS CloudFormation l'utilisateur, consultez la référence aux types de ApplicationInsights ressources.

Exemples de modèle

- Création d'une application Application Insights pour l'ensemble du AWS CloudFormation stack
- Création d'une application Application Insights avec des paramètres détaillés
- Créez une application Application Insights avec un mode de configuration de composants CUSTOM
- Créez une application Application Insights avec un mode de configuration de composants DEFAULT
- Créez une application Application Insights avec un mode de configuration de composants DEFAULT_WITH_OVERWRITE

Création d'une application Application Insights pour l'ensemble du AWS CloudFormation stack

Pour appliquer le modèle suivant, vous devez créer des AWS ressources et un ou plusieurs groupes de ressources à partir desquels créer des applications Application Insights afin de surveiller ces ressources. Pour plus d'informations, consultez Premiers pas avec les Groueps de ressources AWS.

Les deux premières parties du modèle suivant spécifient une ressource et un Resource Group. La dernière partie du modèle crée une application Application Insights pour le Resource Group, mais ne configure pas l'application ni n'applique aucune surveillance. Pour plus d'informations, consultez les détails de la CreateApplicationcommande dans le manuel Amazon CloudWatch Application Insights API Reference.

Modèle au format JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Resource Group stack",
    "Resources": {
        "EC2Instance": {
            "Type": "AWS::EC2::Instance",
            "Properties": {
                "ImageId" : "ami-abcd1234efgh5678i",
                "SecurityGroupIds" : ["sg-abcd1234"]
            }
        },
        ...
        "ResourceGroup": {
            "Type": "AWS::ResourceGroups::Group",
            "Properties": {
                "Name": "my_resource_group"
            }
        },
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
                "ResourceGroupName": "my_resource_group"
            },
            "DependsOn" : "ResourceGroup"
        }
    }
}
```

Modèle au format YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Resource Group stack
Resources:
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: ami-abcd1234efgh5678i
      SecurityGroupIds:
      - sg-abcd1234
  ...
  ResourceGroup:
    Type: AWS::ResourceGroups::Group
    Properties:
      Name: my_resource_group
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
    DependsOn: ResourceGroup
```

La section de modèle suivante applique la configuration de surveillance par défaut à l'application Application Insights. Pour plus d'informations, consultez les détails de la CreateApplicationcommande dans le manuel Amazon CloudWatch Application Insights API Reference.

Lorsque `AutoConfigurationEnabled` est défini sur `true`, tous les composants de l'application sont configurés avec les paramètres de surveillance recommandés pour le niveau application `DEFAULT`. Pour plus d'informations sur ces paramètres et niveaux, consultez DescribeComponentConfigurationRecommendationet UpdateComponentConfigurationdans le manuel Amazon CloudWatch Application Insights API Reference.

Modèle au format JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Test Application Insights Application stack",
    "Resources": {
        "AppInsightsApp": {
            "Type": "AWS::ApplicationInsights::Application",
            "Properties": {
```

```
                "ResourceGroupName": "my_resource_group",
                "AutoConfigurationEnabled": true
            }
        }
    }
}
```

Modèle au format YAML

```
---
AWSTemplateFormatVersion: '2010-09-09'
Description: Test Application Insights Application stack
Resources:
  AppInsightsApp:
    Type: AWS::ApplicationInsights::Application
    Properties:
      ResourceGroupName: my_resource_group
      AutoConfigurationEnabled: true
```

Création d'une application Application Insights avec des paramètres détaillés

Le modèle suivant exécute ces actions :

- Crée une application Application Insights avec notification d' CloudWatch événements et OpsCenter activée. Pour plus d'informations, consultez les détails de la CreateApplicationcommande dans le manuel Amazon CloudWatch Application Insights API Reference.
- Étiquette l'application avec deux étiquettes, dont l'une n'a aucune valeur d'étiquette Pour plus d'informations, consultez TagResourcele manuel Amazon CloudWatch Application Insights API Reference.
- Crée deux composants de groupe d'instance personnalisés. Pour plus d'informations, consultez CreateComponentle manuel Amazon CloudWatch Application Insights API Reference.
- Crée deux jeux de motifs de journal. Pour plus d'informations, consultez CreateLogPatternle manuel Amazon CloudWatch Application Insights API Reference.
- Définit `AutoConfigurationEnabled` sur `true`, qui configure tous les composants de l'application avec les paramètres de surveillance recommandés pour le niveau `DEFAULT`. Pour plus d'informations, consultez DescribeComponentConfigurationRecommendationle manuel Amazon CloudWatch Application Insights API Reference.

Modèle au format JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "CWEMonitorEnabled": true,
        "OpsCenterEnabled": true,
        "OpsItemSNSTopicArn": "arn:aws:sns:us-east-1:123456789012:my_topic",
        "AutoConfigurationEnabled": true,
        "Tags": [
            {
                "Key": "key1",
                "Value": "value1"
            },
            {
                "Key": "key2",
                "Value": ""
            }
        ],
        "CustomComponents": [
            {
                "ComponentName": "test_component_1",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            },
            {
                "ComponentName": "test_component_2",
                "ResourceList": [
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i",
                    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i"
                ]
            }
        ],
        "LogPatternSets": [
            {
                "PatternSetName": "pattern_set_1",
                "LogPatterns": [
                    {
                        "PatternName": "deadlock_pattern",
                        "Pattern": ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))",
                        "Rank": 1
                    }
```

```
                ]
            },
            {
                "PatternSetName": "pattern_set_2",
                "LogPatterns": [
                    {
                        "PatternName": "error_pattern",
                        "Pattern": ".*[\\s\\[]ERROR[\\s\\]].*",
                        "Rank": 1
                    },
                    {
                        "PatternName": "warning_pattern",
                        "Pattern": ".*[\\s\\[]WARN(ING)?[\\s\\]].*",
                        "Rank": 10
                    }
                ]
            }
        ]
    }
}
```

Modèle au format YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  CWEMonitorEnabled: true
  OpsCenterEnabled: true
  OpsItemSNSTopicArn: arn:aws:sns:us-east-1:123456789012:my_topic
  AutoConfigurationEnabled: true
  Tags:
  - Key: key1
    Value: value1
  - Key: key2
    Value: ''
  CustomComponents:
  - ComponentName: test_component_1
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  - ComponentName: test_component_2
    ResourceList:
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
```

```
    - arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1234efgh5678i
  LogPatternSets:
  - PatternSetName: pattern_set_1
    LogPatterns:
    - PatternName: deadlock_pattern
      Pattern: ".*\\sDeadlocked\\sSchedulers(([^\\w].*)|($))"
      Rank: 1
  - PatternSetName: pattern_set_2
    LogPatterns:
    - PatternName: error_pattern
      Pattern: ".*[\\s\\[]ERROR[\\s\\]].*"
      Rank: 1
    - PatternName: warning_pattern
      Pattern: ".*[\\s\\[]WARN(ING)?[\\s\\]].*"
      Rank: 10
```

Créez une application Application Insights avec un mode de configuration de composants **CUSTOM**

Le modèle suivant exécute ces actions :

- Crée une application Application Insights. Pour plus d'informations, consultez CreateApplicationle manuel Amazon CloudWatch Application Insights API Reference.
- Le composant `my_component` définit `ComponentConfigurationMode` sur `CUSTOM`, ce qui entraîne ce composant d'être configuré avec le configuration spécifiée dans `CustomComponentConfiguration`. Pour plus d'informations, consultez UpdateComponentConfigurationle manuel Amazon CloudWatch Application Insights API Reference.

Modèle au format JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group,
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "CUSTOM",
                "CustomComponentConfiguration": {
```

```
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            },
                            ...
                        ],
                        "Logs": [
                            {
                                "LogGroupName": "my_log_group_1",
                                "LogPath": "C:\\LogFolder_1\\*",
                                "LogType": "DOT_NET_CORE",
                                "Encoding": "utf-8",
                                "PatternSet": "my_pattern_set_1"
                            },
                            ...
                        ],
                        "WindowsEvents": [
                            {
                                "LogGroupName": "my_windows_event_log_group_1",
                                "EventName": "Application",
                                "EventLevels": [
                                    "ERROR",
                                    "WARNING",
                                    ...
                                ],
                                "Encoding": "utf-8",
                                "PatternSet": "my_pattern_set_2"
                            },
                            ...
                        ],
                        "Alarms": [
                            {
                                "AlarmName": "my_alarm_name",
                                "Severity": "HIGH"
                            },
                            ...
                        ]
                    },
                    "SubComponentTypeConfigurations": [
                        {
                            "SubComponentType": "EC2_INSTANCE",
                            "SubComponentConfigurationDetails": {
                                "AlarmMetrics": [
```

```
                                {
                                    "AlarmMetricName": "DiskReadOps"
                                },
                                ...
                            ],
                            "Logs": [
                                {
                                    "LogGroupName": "my_log_group_2",
                                    "LogPath": "C:\\LogFolder_2\\*",
                                    "LogType": "IIS",
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_3"
                                },
                                ...
                            ],
                            "processes" : [
                                {
                                    "processName" : "my_process",
                                    "alarmMetrics" : [
                                {
                                    "alarmMetricName" : "procstat cpu_usage",
                                    "monitor" : true
                                }, {
                                    "alarmMetricName" : "procstat memory_rss",
                                    "monitor" : true
                                }
                            ]
                        }
                    ],
                            "WindowsEvents": [
                                {
                                    "LogGroupName": "my_windows_event_log_group_2",
                                    "EventName": "Application",
                                    "EventLevels": [
                                        "ERROR",
                                        "WARNING",
                                        ...
                                    ],
                                    "Encoding": "utf-8",
                                    "PatternSet": "my_pattern_set_4"
                                },
                                ...
                            ]
                        }
```

```
                        }
                    ]
                }
            }
        ]
    }
}
```

Modèle au format YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: CUSTOM
    CustomComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
        ...
        Logs:
        - LogGroupName: my_log_group_1
          LogPath: C:\LogFolder_1\*
          LogType: DOT_NET_CORE
          Encoding: utf-8
          PatternSet: my_pattern_set_1
        ...
        WindowsEvents:
        - LogGroupName: my_windows_event_log_group_1
          EventName: Application
          EventLevels:
          - ERROR
          - WARNING
          ...
          Encoding: utf-8
          PatternSet: my_pattern_set_2
        ...
        Alarms:
        - AlarmName: my_alarm_name
          Severity: HIGH
```

```
      ...
    SubComponentTypeConfigurations:
    - SubComponentType: EC2_INSTANCE
      SubComponentConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: DiskReadOps
        ...
        Logs:
        - LogGroupName: my_log_group_2
          LogPath: C:\LogFolder_2\*
          LogType: IIS
          Encoding: utf-8
          PatternSet: my_pattern_set_3
        ...
        Processes:
        - ProcessName: my_process
          AlarmMetrics:
          - AlarmMetricName: procstat cpu_usage
            ...
          ...
        WindowsEvents:
        - LogGroupName: my_windows_event_log_group_2
          EventName: Application
          EventLevels:
          - ERROR
          - WARNING
          ...
          Encoding: utf-8
          PatternSet: my_pattern_set_4
        ...
```

Créez une application Application Insights avec un mode de configuration de composants **DEFAULT**

Le modèle suivant exécute ces actions :

- Crée une application Application Insights. Pour plus d'informations, consultez CreateApplicationle manuel Amazon CloudWatch Application Insights API Reference.
- Le composant `my_component` définit `ComponentConfigurationMode` sur `DEFAULT` et `Tier` sur `SQL_SERVER`, ce qui entraîne la configuration de ce composant avec les paramètres de configuration recommandés par Application Insights pour le niveau `SQL_Server`. Pour plus d'informations, consultez DescribeComponentConfigurationet consultez

UpdateComponentConfigurationle manuel Amazon CloudWatch Application Insights API Reference.

Modèle au format JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentARN": "my_component",
                "Tier": "SQL_SERVER",
                "ComponentConfigurationMode": "DEFAULT"
            }
        ]
    }
}
```

Modèle au format YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentARN: my_component
    Tier: SQL_SERVER
    ComponentConfigurationMode: DEFAULT
```

Créez une application Application Insights avec un mode de configuration de composants **DEFAULT_WITH_OVERWRITE**

Le modèle suivant exécute ces actions :

- Crée une application Application Insights. Pour plus d'informations, consultez CreateApplicationle manuel Amazon CloudWatch Application Insights API Reference.
- Le composant `my_component` définit `ComponentConfigurationMode` sur `DEFAULT_WITH_OVERWRITE` et `tier` sur `DOT_NET_CORE`, ce qui entraîne la configuration de ce composant avec les paramètres de configuration recommandés par Application Insights

pour le niveau `DOT_NET_CORE`. Les paramètres de configuration écrasés sont spécifiés dans `DefaultOverwriteComponentConfiguration` :

- Au niveau du composant, les paramètres `AlarmMetrics` sont remplacés.
- Au niveau des sous-composants, pour les sous-composants de type `EC2_Instance`, les paramètres `Logs` sont remplacés.

Pour plus d'informations, consultez UpdateComponentConfigurationle manuel Amazon CloudWatch Application Insights API Reference.

Modèle au format JSON

```
{
    "Type": "AWS::ApplicationInsights::Application",
    "Properties": {
        "ResourceGroupName": "my_resource_group",
        "ComponentMonitoringSettings": [
            {
                "ComponentName": "my_component",
                "Tier": "DOT_NET_CORE",
                "ComponentConfigurationMode": "DEFAULT_WITH_OVERWRITE",
                "DefaultOverwriteComponentConfiguration": {
                    "ConfigurationDetails": {
                        "AlarmMetrics": [
                            {
                                "AlarmMetricName": "StatusCheckFailed"
                            }
                        ]
                    },
                    "SubComponentTypeConfigurations": [
                        {
                            "SubComponentType": "EC2_INSTANCE",
                            "SubComponentConfigurationDetails": {
                                "Logs": [
                                    {
                                        "LogGroupName": "my_log_group",
                                        "LogPath": "C:\\LogFolder\\*",
                                        "LogType": "IIS",
                                        "Encoding": "utf-8",
                                        "PatternSet": "my_pattern_set"
                                    }
                                ]
```

```
                            }
                        }
                    ]
                }
            }
        ]
    }
}
```

Modèle au format YAML

```
---
Type: AWS::ApplicationInsights::Application
Properties:
  ResourceGroupName: my_resource_group
  ComponentMonitoringSettings:
  - ComponentName: my_component
    Tier: DOT_NET_CORE
    ComponentConfigurationMode: DEFAULT_WITH_OVERWRITE
    DefaultOverwriteComponentConfiguration:
      ConfigurationDetails:
        AlarmMetrics:
        - AlarmMetricName: StatusCheckFailed
      SubComponentTypeConfigurations:
      - SubComponentType: EC2_INSTANCE
        SubComponentConfigurationDetails:
          Logs:
          - LogGroupName: my_log_group
            LogPath: C:\LogFolder\*
            LogType: IIS
            Encoding: utf-8
            PatternSet: my_pattern_set
```

Didacticiel : Configurer la surveillance pour SAP ASE

Ce didacticiel explique comment configurer CloudWatch Application Insights pour configurer la surveillance de vos bases de données SAP ASE. Vous pouvez utiliser les tableaux de bord automatiques d' CloudWatch Application Insights pour visualiser les détails des problèmes, accélérer le dépannage et faciliter le délai moyen de résolution (MTTR) de vos bases de données SAP ASE.

Sujets Application Insights pour SAP ASE

- Environnements compatibles

Environnements compatibles

CloudWatch Application Insights prend en charge le déploiement de AWS ressources pour les systèmes et modèles suivants. Vous fournissez et installez le logiciel de base de données SAP ASE ainsi que le logiciel d'application SAP.

- Une ou plusieurs bases de données SAP ASE sur une seule instance Amazon EC2 : SAP ASE dans une architecture à nœud unique en expansion.
- Configuration de la haute disponibilité de la base de données SAP ASE inter-AZ : SAP ASE avec haute disponibilité configurée sur deux zones de disponibilité à l'aide du clustering SUSE/RHEL.

> **Note**
>
> CloudWatch Application Insights ne prend en charge qu'un seul environnement SAP System ID (SID) ASE HA. Si plusieurs SID ASE HA sont connectés, la surveillance ne sera configurée que pour le premier SID détecté.

Systèmes d’exploitation pris en charge

CloudWatch Application Insights for SAP ASE prend en charge l'architecture x86-64 sur les systèmes d'exploitation suivants :

- SuSE Linux 12 SP4
- SuSE Linux 12 SP5
- SuSE Linux 15

- Utiliser Linux 15 SP1
- SUSE Linux 15 SP2
- SuSE Linux 15 SP3
- SuSE Linux 15 SP4
- SuSE Linux 15 SP1 pour SAP
- SuSE Linux 15 SP2 pour SAP
- SuSE Linux 15 SP3 pour SAP
- SuSE Linux 15 SP4 pour SAP
- SuSE Linux 12 SP4 pour SAP
- SuSE Linux 12 SP5 pour SAP
- RedHat Linux 7.6
- RedHat Linux 7.7
- RedHat Linux 7.9
- RedHat Linux 8.1
- RedHat Linux 8.4
- RedHat Linux 8.6

Fonctionnalités

CloudWatch Application Insights for SAP ASE fournit les fonctionnalités suivantes :

- Détection automatique de la charge de travail SAP ASE
- Création automatique d'alarme SAP ASE basée sur un seuil statique
- Création automatique d'alarme SAP ASE basée sur la détection d'anomalies
- Reconnaissance automatique des modèles de journaux SAP ASE
- Tableau de bord d'état pour SAP ASE
- Tableau de bord des problèmes pour SAP ASE

Prérequis

Vous devez remplir les conditions préalables suivantes pour configurer une base de données SAP ASE avec CloudWatch Application Insights :

- Paramètres de configuration SAP ASE : les paramètres de configuration suivants doivent être activés sur votre base de données ASE – `"enable monitoring"`, `"sql text pipe max messages"`, `"sql text pipe active"`. Cela permet à CloudWatch Application Insights de fournir des fonctionnalités de surveillance complètes pour votre base de données. Si ces paramètres ne sont pas activés sur votre base de données ASE, Application Insights leur permettra automatiquement de collecter les métriques nécessaires pour permettre la surveillance.
- Utilisateur de base de données SAP ASE : l'utilisateur de base de données indiqué lors de l'intégration d'Application Insights doit être autorisé à accéder aux éléments suivants :
  - Tables système de la base de données principale et des bases de données utilisateur (locataires)
  - Surveillance des tables
- SAP HostCtrl — Installez et configurez SAP HostCtrl sur votre instance Amazon EC2.
- CloudWatch Agent Amazon — Assurez-vous que vous n'exécutez pas d' CloudWatch agent préexistant sur votre instance Amazon EC2. Si CloudWatch l'agent est installé, veillez à supprimer la configuration des ressources que vous utilisez dans CloudWatch Application Insights du fichier de configuration de l' CloudWatch agent existant afin d'éviter un conflit de fusion. Pour plus d'informations, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.
- AWS Activation de Systems Manager : installez l'agent SSM sur vos instances et activez les instances activées pour SSM. Pour plus d'informations sur SSM Agent, consultez Utilisation de SSM Agent dans le Guide de l'utilisateur Systems Manager AWS .
- Rôles d'instance Amazon EC2 : vous devez attacher les rôles d'instance Amazon EC2 suivants pour configurer votre base de données.
  - Vous devez joindre le rôle `AmazonSSMManagedInstanceCore` pour activer Systems Manager. Pour plus d'informations, consultez AWS Systems Manager Exemples de politiques basées sur l'identité.
  - Vous devez joindre le `CloudWatchAgentServerPolicy` pour permettre l'émission de métriques et de journaux d'instance CloudWatch. Pour plus d'informations, consultez Créer des rôles et des utilisateurs IAM à utiliser avec l' CloudWatch agent Amazon.
  - Vous devez attacher la politique intégrée IAM suivante au rôle d'instance Amazon EC2 pour lire le mot de passe stocké dans AWS Secrets Manager. Pour plus d'informations sur les politiques en ligne, consultez Politiques en ligne dans le AWS Identity and Access Management Guide de l'utilisateur IAM.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- AWS Resource Groups— Vous devez créer un groupe de ressources qui inclut toutes les AWS ressources associées utilisées par votre pile d'applications pour intégrer vos applications à CloudWatch Application Insights. Cela comprend les instances Amazon EC2 et les volumes Amazon EBS exécutant votre base de données SAP ASE. S'il existe plusieurs bases de données par compte, nous vous recommandons de créer un groupe de ressources qui inclut les AWS ressources de chaque système de base de données SAP ASE.
- Autorisations IAM : pour les utilisateurs non-administrateurs ;
  - Vous devez créer une politique AWS Identity and Access Management (IAM) qui permet à Application Insights de créer un rôle lié à un service et de l'associer à votre identité d'utilisateur. Pour savoir comment attacher la politique, consultez Politique IAM.
  - L'utilisateur doit être autorisé à créer un secret pour stocker les informations d'identification AWS Secrets Manager de l'utilisateur de la base de données. Pour de plus amples informations, consultez Example: Permission to create secrets (Exemple : Autorisation de créer des secrets).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Rôle lié à un service — Application Insights utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est créé pour vous lorsque vous créez votre première application Application Insights dans la console Application Insights. Pour plus d'informations, consultez Utilisation de rôles liés à un service pour Application Insights CloudWatch .

Configuration de la surveillance de votre base de données SAP ASE

Réalisez les étapes suivantes pour configurer la surveillance de votre base de données SAP ASE.

1. Ouvrez la CloudWatch console.
2. Dans le volet de navigation de gauche, sélectionnez Application Insights sous Insights.
3. La page Application Insights affiche la liste des applications qui sont surveillées à l'aide d'Application Insights et l'état de surveillance de chaque application. Dans le coin supérieur droit, sélectionnez Add an application (Ajouter une application).
4. Sur la page Spécifier les détails de l'application, dans la liste déroulante sous Groupe de ressources, sélectionnez le groupe de ressources AWS qui contient vos ressources de base de données SAP ASE. Si vous n'avez pas encore créé de Resource Group pour votre application, vous pouvez le faire en sélectionnant Create new resource group (Créer un Resource Group) sous le menu déroulant Resource Group. Pour plus d'informations sur les Resource Groups, consultez le Guide d'utilisateur AWS Resource Groups.
5. Sous Surveiller les CloudWatch événements, cochez la case pour intégrer la surveillance des informations d'application aux CloudWatch événements afin d'obtenir des informations provenant d'Amazon EBS, d'Amazon EC2 AWS CodeDeploy, d'Amazon ECS AWS Health , des API et des notifications, d'Amazon RDS, d'Amazon S3 et. AWS Step Functions
6. Sous Intégrer à AWS Systems Manager OpsCenter, cochez la case à côté de Générer AWS Systems Manager OpsCenter OpsItems pour les actions correctives afin de consulter et de recevoir des notifications lorsque des problèmes sont détectés dans les applications sélectionnées. Pour suivre les opérations effectuées pour résoudre les éléments de travail opérationnels, appelés OpsItems, liés à vos AWS ressources, fournissez un ARN de rubrique SNS.
7. Vous pouvez éventuellement saisir des balises pour vous aider à identifier et à organiser vos ressources. CloudWatch Application Insights prend en charge les groupes de ressources AWS CloudFormation basés sur des balises et des piles, à l'exception des groupes. Application Auto Scaling Pour plus d'informations, consultez Utilisation de Tag Editor dans le Guide de l'utilisateur AWS Resource Groups .

8. Sélectionnez Next (Suivant) pour continuer à mettre en place la surveillance.
9. Sur la page Réviser les composants détectés, les composants surveillés et leurs charges de travail détectées automatiquement par CloudWatch Application Insights sont répertoriés.

> **Note**
>
> Les composants contenant une charge de travail à haute disponibilité SAP ASE détectée ne prennent en charge qu'une seule charge de travail par composant. Les composants contenant une charge de travail à nœud unique SAP ASE détectée prennent en charge plusieurs charges de travail, mais vous ne pouvez ni ajouter ni supprimer de charges de travail. Toutes les charges de travail détectées automatiquement seront surveillées.

10. Choisissez Suivant.
11. Sur la page Spécifier les détails du composant, saisissez le nom d'utilisateur et le mot de passe de vos bases de données SAP ASE.
12. Passez en revue la configuration de surveillance de votre application et sélectionnez Submit (Envoyer).
13. La page de détails de l'application s'ouvre. Vous pouvez consulter le Récapitulatif de l'application, la liste des Composants surveillés et charges de travail surveillés et les Composants et charges de travail non surveillés. Si vous sélectionnez le bouton radio à côté d'un composant ou d'une charge de travail, vous pouvez également afficher l'Historique de configuration, les Modèles de journaux et toutes les Balises que vous avez créées. Lorsque vous envoyez votre configuration, votre compte déploie toutes les métriques et alarmes de votre système SAP ASE, ce qui peut prendre jusqu'à 2 heures.

Gestion de la surveillance de votre base de données SAP ASE

Vous pouvez gérer les informations d'identification des utilisateurs, les métriques et les chemins d'accès aux journaux de votre base de données SAP ASE en réalisant les étapes suivantes :

1. Ouvrez la CloudWatch console.
2. Dans le volet de navigation de gauche, sélectionnez Application Insights sous Insights.
3. La page Application Insights affiche la liste des applications qui sont surveillées à l'aide d'Application Insights et l'état de surveillance de chaque application.
4. Sous Monitored components (Composants surveillés), sélectionnez la case d'option à côté du nom du composant. Ensuite, sélectionnez Manage monitoring (Gérer la surveillance).

5. Sous EC2 instance group logs (Journaux des groupes d'instances EC2), vous pouvez mettre à jour le chemin d'accès du journal, le jeu de modèles de journaux et le nom du groupe de journaux existants. En outre, vous pouvez ajouter jusqu'à trois autres Journaux d'application.
6. Sous Métriques, vous pouvez choisir les métriques SAP ASE en fonction de vos besoins. Les noms des métriques SAP ASE sont préfixés par `asedb`. Vous pouvez ajouter jusqu'à 60 métriques par composant.
7. Sous Configuration ASE, saisissez le mot de passe et le nom d'utilisateur de la base de données SAP ASE. Il s'agit du nom d'utilisateur et du mot de passe utilisés par CloudWatch l'agent Amazon pour se connecter à la base de données SAP ASE.
8. Sous Alarmes personnalisées, vous pouvez ajouter des alarmes supplémentaires à surveiller par CloudWatch Application Insights.
9. Vérifiez la configuration de surveillance de votre application et sélectionnez Submit (Envoyer). Lorsque vous envoyez votre configuration, votre compte met à jour toutes les métriques et les alertes de votre système SAP HANA, ce qui peut prendre jusqu'à 2 heures.

Configurer le seuil d'alerte

CloudWatch Application Insights crée automatiquement une CloudWatch métrique Amazon pour l'alarme à surveiller, ainsi que le seuil correspondant à cette métrique. L'alerte passe à l'état `ALARM` lorsque la métrique dépasse le seuil pendant un certain nombre de périodes d'évaluation. Notez que ces paramètres ne sont pas conservés par Application Insights.

Pour modifier une alerte pour une métrique unique, réalisez les étapes suivantes :

1. Ouvrez la CloudWatch console.
2. Dans le panneau de navigation, sélectionnez Alarms (alertes), All alarms (Toutes les alertes).
3. Sélectionnez le bouton radio situé à côté de l'alarme créée automatiquement par CloudWatch Application Insights. Puis sélectionnez Actions, puis sélectionnez Edit (Modifier) dans le menu déroulant.
4. Modifiez les paramètres suivants sous Metric (Métrique).
    a. Sous Statistic (Statistiques), sélectionnez l'une des statistiques ou l'un des centiles prédéfinis, ou spécifiez un centile personnalisé. Par exemple, `p95.45`.
    b. Sous Période, sélectionnez la période d'évaluation de l'alerte. Lors de l'évaluation de l'alerte, chaque période est regroupée en un point de données.
5. Modifiez les paramètres suivants sous Conditions.

a. Choisissez si la métrique doit être supérieure, inférieure ou égale au seuil.

b. Spécifiez la valeur de seuil.

6. Sous Additional configuration (Configuration supplémentaire), modifiez les paramètres suivants :

a. Sous Datapoints to alarm (Points de données pour alerte), spécifiez le nombre de points de données ou de périodes d'évaluation qui doivent se trouver dans l'état `ALARM` pour déclencher l'alerte. Lorsque les deux valeurs correspondent, une alerte est créée et passe à l'état `ALARM` si le nombre désigné de périodes consécutives est dépassé. Pour créer une alerte `m` sur `n`, spécifiez pour le premier point de données une valeur inférieure à celle du second. Pour plus d'informations sur l'évaluation des alertes, consultez Évaluation d'une alerte.

b. 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 sur le traitement des données manquantes, voir Configuration de la manière dont les CloudWatch alarmes traitent les données manquantes.

c. Si l'alerte utilise un centile comme statistique surveillée, une zone Percentiles with low samples (Centiles avec exemples de bas niveau) s'affiche. Choisissez d'évaluer ou d'ignorer les cas à faible taux d'échantillons. Si vous sélectionnez ignore (ignorer : conserver l'état d'alerte), l'état actuel de l'alerte est toujours conservé lorsque la taille de l'échantillon est trop réduite. Pour plus d'informations sur les centiles avec de faibles échantillons, consultez la section CloudWatch Alarmes basées sur les percentiles et échantillons de données faibles.

7. Sélectionnez Next (Suivant).

8. Sous Notification, sélectionnez la rubrique SNS qui doit recevoir une notification lorsque l'alerte passe à l'état `ALARM`, `OK` ou `INSUFFICIENT_DATA`.

9. Sélectionnez Update alarm (Mettre à jour une alerte).

Affichage et résolution des problèmes SAP ASE détectés par Application Insights

Cette section vous aide à résoudre les problèmes de dépannage courants qui surviennent lorsque vous configurez la surveillance de SAP ASE sur Application Insights.

Erreurs du serveur de sauvegarde SAP ASE

Vous pouvez identifier le message d'erreur en consultant le tableau de bord créé dynamiquement. Le tableau de bord affiche le message d'erreur signalé dans le serveur de sauvegarde SAP ASE. Pour

plus de détails sur les journaux du serveur de sauvegarde SAP ASE, veuillez consulter Journalisation des erreurs du serveur de sauvegarde dans la documentation SAP (langue française non garantie).

Transactions de longue durée SAP ASE

Identifiez la transaction de longue durée en cours et confirmez si elle peut être arrêtée ou si la durée est intentionnelle. Pour plus de détails, veuillez consulter 2180410 — Comment afficher les enregistrements du journal des transactions pour les transactions de longue durée ? (langue française non garantie). — SAP ASE.

Connexions utilisateur SAP ASE

Vérifiez si votre base de données SAP ASE est dimensionnée en fonction de la charge de travail que vous souhaitez exécuter sur la base de données. Pour plus de détails, veuillez consulter Configuration des connexions d'utilisateurs dans la documentation SAP (langue française non garantie).

Espace disque SAP ASE

Vous pouvez identifier la couche de base de données à l'origine du problème en consultant le tableau de bord créé dynamiquement. Le tableau de bord présente les métriques et les extraits de fichiers journaux correspondants. Il est important de comprendre la cause de la montée en charge du disque et, le cas échéant, d'augmenter la taille du disque physique, l'espace disque alloué, ou les deux. Pour plus de détails, veuillez consulter la section relative auredimensionnement de disque dans la documentation SAP (langue française non garantie).

Dépannage des applications Insights pour SAP ASE

Cette section fournit des étapes pour vous aider à résoudre les erreurs courantes renvoyées par le tableau de bord Application Insights.

Erreur	Erreur renvoyée	Cause profonde	Résolution
Impossible d'ajouter plus de 60 métriques de surveillance.	`Component cannot have more than 60 monitored metric`	La limite actuelle du nombre de métriques est de 60 métriques surveillées par composant.	Supprimez les métriques inutiles pour respecter la limite.

Erreur	Erreur renvoyée	Cause profonde	Résolution
Aucune métrique ou alarme SAP n'apparaît après le processus d'intégration.	La commande `run` sur le `AWS-ConfigureAWSPackage` a échoué dans AWS Systems Manager. Le résultat montre l'erreur suivante : `CT-LIBRARY error:ct_connect(): protocol specific layer: external error: The attempt to connect to the server failed`	Le nom d'utilisateur et le mot de passe sont peut-être incorrects.	Vérifiez que le nom d'utilisateur et le mot de passe sont valides, puis relancez le processus d'intégration.

Didacticiel : configurer la surveillance pour SAP HANA

Ce didacticiel explique comment configurer CloudWatch Application Insights pour configurer la surveillance de vos bases de données SAP HANA. Vous pouvez utiliser les tableaux de bord automatiques d' CloudWatch Application Insights pour visualiser les détails des problèmes, accélérer le dépannage et faciliter le délai moyen de résolution (MTTR) de vos bases de données SAP HANA.

Sujets Application Insights pour SAP HANA

- Environnements compatibles
- Systèmes d’exploitation pris en charge
- Fonctionnalités
- Prérequis
- Configurer votre base de données SAP HANA pour la surveillance
- Gérer la surveillance de votre base de données SAP H
- Afficher et résoudre les problèmes SAP HANA détectés par CloudWatch Application Insights
- Détection d'anomalies pour SAP HANA

- Dépannage des applications Insights pour SAP HANA

Environnements compatibles

CloudWatch Application Insights prend en charge le déploiement de AWS ressources pour les systèmes et modèles suivants. Vous fournissez et installez le logiciel de base de données SAP HANA ainsi que le logiciel d'application SAP

- Base de données SAP HANA sur une seule instance Amazon EC2 — SAP HANA dans une architecture mono-nœud évolutive, avec jusqu'à 24 To de mémoire.
- Base de données SAP HANA sur plusieurs instances Amazon EC2 — SAP HANA dans une architecture multi-nœuds évolutive.
- Configuration de la haute disponibilité de la base de données SAP HANA — SAP HANA avec haute disponibilité configurée dans deux zones de disponibilité à l'aide du clustering SUSE/RHEL.

> **Note**
>
> CloudWatch Application Insights ne prend en charge que les environnements SID HANA uniques. Si plusieurs SID HANA sont connectés, la surveillance sera configurée uniquement pour le premier SID détecté.

Systèmes d’exploitation pris en charge

CloudWatch Application Insights for SAP HANA prend en charge l'architecture x86-64 sur les systèmes d'exploitation suivants :

- SuSE Linux 12 SP4 pour SAP
- SuSE Linux 12 SP5 pour SAP
- SuSE Linux 15
- Utiliser Linux 15 SP1
- SUSE Linux 15 SP2
- SuSE Linux 15 pour SAP
- SuSE Linux 15 SP1 pour SAP
- SuSE Linux 15 SP2 pour SAP

- SuSE Linux 15 SP3 pour SAP
- SuSE Linux 15 SP4 pour SAP
- SuSE Linux 15 SP5 pour SAP
- RedHat Linux 8.6 pour SAP avec haute disponibilité et services de mise à jour
- RedHat Linux 8.5 pour SAP avec haute disponibilité et services de mise à jour
- RedHat Linux 8.4 pour SAP avec haute disponibilité et services de mise à jour
- RedHat Linux 8.3 pour SAP avec haute disponibilité et services de mise à jour
- RedHat Linux 8.2 pour SAP avec haute disponibilité et services de mise à jour
- RedHat Linux 8.1 pour SAP avec haute disponibilité et services de mise à jour
- RedHat Linux 7.9 pour SAP avec haute disponibilité et services de mise à jour

Fonctionnalités

CloudWatch Application Insights for SAP HANA fournit les fonctionnalités suivantes :

- Détection automatique des charges de travail SAP HANA
- Création automatique d'alerte SAP HANA basée sur un seuil statique
- Création automatique d'alerte SAP HANA basée sur la détection d'anomalies
- Reconnaissance automatique des modèles de journaux SAP HANA
- Tableau de bord de Health pour SAP HANA
- Tableau de bord des problèmes pour SAP HANA

Prérequis

Vous devez remplir les conditions préalables suivantes pour configurer une base de données SAP HANA avec CloudWatch Application Insights :

- SAP HANA — Installez une base de données SAP HANA 2.0 SPS05 active et accessible sur une instance Amazon EC2.
- Utilisateur de base de données SAP HANA : un utilisateur de base de données doté de rôles de surveillance doit être créé dans la base de données SYSTEM et dans tous les locataires.

  Exemple

  Les commandes SQL suivantes créent un utilisateur avec des rôles de surveillance.

```
su - <sid>adm
hdbsql -u SYSTEM -p <SYSTEMDB password> -d SYSTEMDB
CREATE USER CW_HANADB_EXPORTER_USER PASSWORD <Monitoring user password> NO
 FORCE_FIRST_PASSWORD_CHANGE;
CREATE ROLE CW_HANADB_EXPORTER_ROLE;
GRANT MONITORING TO CW_HANADB_EXPORTER_ROLE;
GRANT CW_HANADB_EXPORTER_ROLE TO CW_HANADB_EXPORTER_USER;
```

- Python 3.8 — Installez Python 3.8 ou une version ultérieure sur votre système d'exploitation. Utilisez la dernière version de Python. Si Python3 n'est pas détecté sur votre système d'exploitation, Python 3.6 sera installé.

  Pour plus d'informations, consultez le installation example.

  > **Note**
  >
  > L'installation manuelle de Python 3.8 ou version ultérieure est requise pour les systèmes d'exploitation SuSE Linux 15 SP4, RedHat Linux 8.6 et versions ultérieures.

- Pip3 — Installez le programme d'installation, pip3, sur votre système d'exploitation. Si pip3 n'est pas détecté sur votre système d'exploitation, il sera installé.
- hdbclient — CloudWatch Application Insights utilise le pilote python pour se connecter à la base de données SAP HANA. Si le client n'est pas installé sous python3, assurez-vous que la version du fichier tar hdbclient est sous. `2.10 or later` `/hana/shared/SID/hdbclient/`
- CloudWatch Agent Amazon — Assurez-vous que vous n'exécutez pas d' CloudWatch agent préexistant sur votre instance Amazon EC2. Si CloudWatch l'agent est installé, veillez à supprimer la configuration des ressources que vous utilisez dans CloudWatch Application Insights du fichier de configuration de l' CloudWatch agent existant afin d'éviter un conflit de fusion. Pour plus d'informations, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.
- AWS Activation de Systems Manager : installez l'agent SSM sur vos instances, et les instances doivent être activées pour SSM. Pour plus d'informations sur l'installation de l'agent SSM, consultez la section Travailler avec l'agent SSM dans le guide de l'utilisateur de AWS Systems Manager.
- Rôles d'instance Amazon EC2 : vous devez attacher les rôles d'instance Amazon EC2 suivants pour configurer votre base de données.

- Vous devez joindre le rôle `AmazonSSMManagedInstanceCore` pour activer Systems Manager. Pour plus d'informations, consultez AWS Systems Manager Exemples de politiques basées sur l'identité.
- Vous devez joindre le `CloudWatchAgentServerPolicy` pour permettre l'émission de métriques et de journaux d'instance CloudWatch. Pour plus d'informations, voir Création de rôles et d'utilisateurs IAM à utiliser avec l' CloudWatchagent.
- Vous devez attacher la politique intégrée IAM suivante au rôle d'instance Amazon EC2 pour lire le mot de passe stocké dans AWS Secrets Manager. Pour plus d'informations sur les politiques en ligne, consultez Politiques en ligne dans le AWS Identity and Access Management Guide de l'utilisateur IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
        }
    ]
}
```

- AWS groupes de ressources : vous devez créer un groupe de ressources qui inclut toutes les AWS ressources associées utilisées par votre pile d'applications pour intégrer vos applications à CloudWatch Application Insights. Cela comprend les instances Amazon EC2 et les volumes Amazon EBS exécutant votre base de données SAP HANA. S'il existe plusieurs bases de données par compte, nous vous recommandons de créer un groupe de ressources qui inclut les AWS ressources de chaque système de base de données SAP HANA.
- Autorisations IAM : pour les utilisateurs non-administrateurs ;
  - Vous devez créer une politique AWS Identity and Access Management (IAM) qui permet à Application Insights de créer un rôle lié à un service et de l'associer à votre identité d'utilisateur. Pour savoir comment attacher la politique, consultez Politique IAM.
  - L'utilisateur doit être autorisé à créer un secret pour stocker les informations d'identification AWS Secrets Manager de l'utilisateur de la base de données. Pour de plus amples informations, consultez Example: Permission to create secrets (Exemple : Autorisation de créer des secrets).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:CreateSecret"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:ApplicationInsights-*"
    }
  ]
}
```

- Rôle lié à un service — Application Insights utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est créé pour vous lorsque vous créez votre première application Application Insights dans la console Application Insights. Pour plus d'informations, consultez Utilisation de rôles liés à un service pour Application Insights CloudWatch .

Configurer votre base de données SAP HANA pour la surveillance

Réalisez les étapes suivantes pour configurer la surveillance de votre base de données SAP HANA

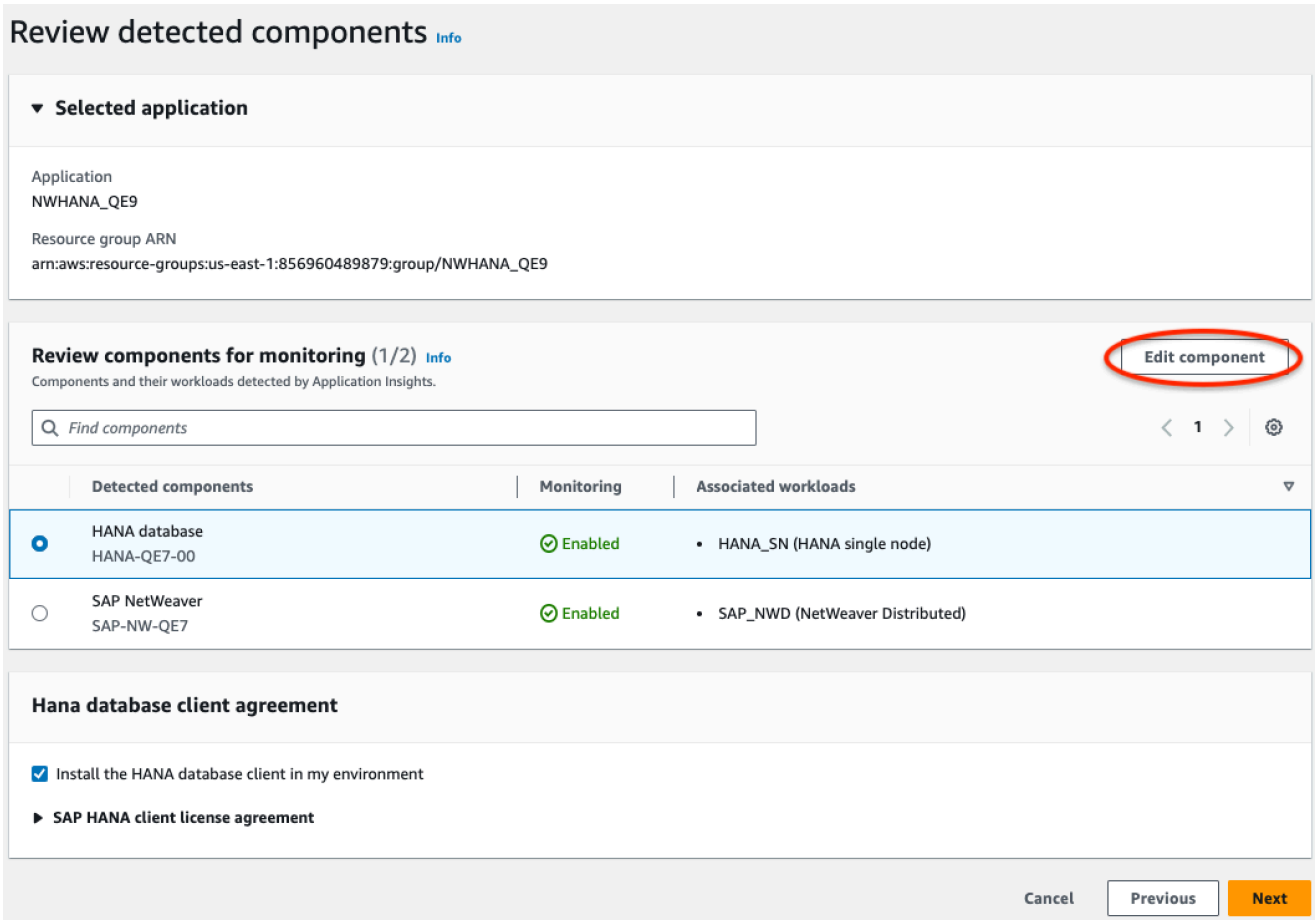
1. Ouvrez la CloudWatch console.
2. Dans le volet de navigation de gauche, sélectionnez Application Insights sous Insights.
3. La page Application Insights affiche la liste des applications qui sont surveillées à l'aide d'Application Insights et l'état de surveillance de chaque application. Dans le coin supérieur droit, sélectionnez Add an application (Ajouter une application).
4. Dans la page Spécification des détails de l'application, dans la liste déroulante sous Resource group, sélectionnez le AWS Resource Group qui contient vos ressources de base de données SAP HANA. Si vous n'avez pas encore créé de Resource Group pour votre application, vous pouvez le faire en sélectionnant Create new resource group (Créer un Resource Group) sous le menu déroulant Resource Group. Pour plus d'informations sur les Resource Groups, consultez le Guide d'utilisateur AWS Resource Groups.
5. Sous Surveiller les CloudWatch événements, cochez la case pour intégrer la surveillance des informations d'application aux CloudWatch événements afin d'obtenir des informations provenant

d'Amazon EBS, d'Amazon EC2 AWS CodeDeploy, d'Amazon ECS AWS Health , des API et des notifications, d'Amazon RDS, d'Amazon S3 et. AWS Step Functions

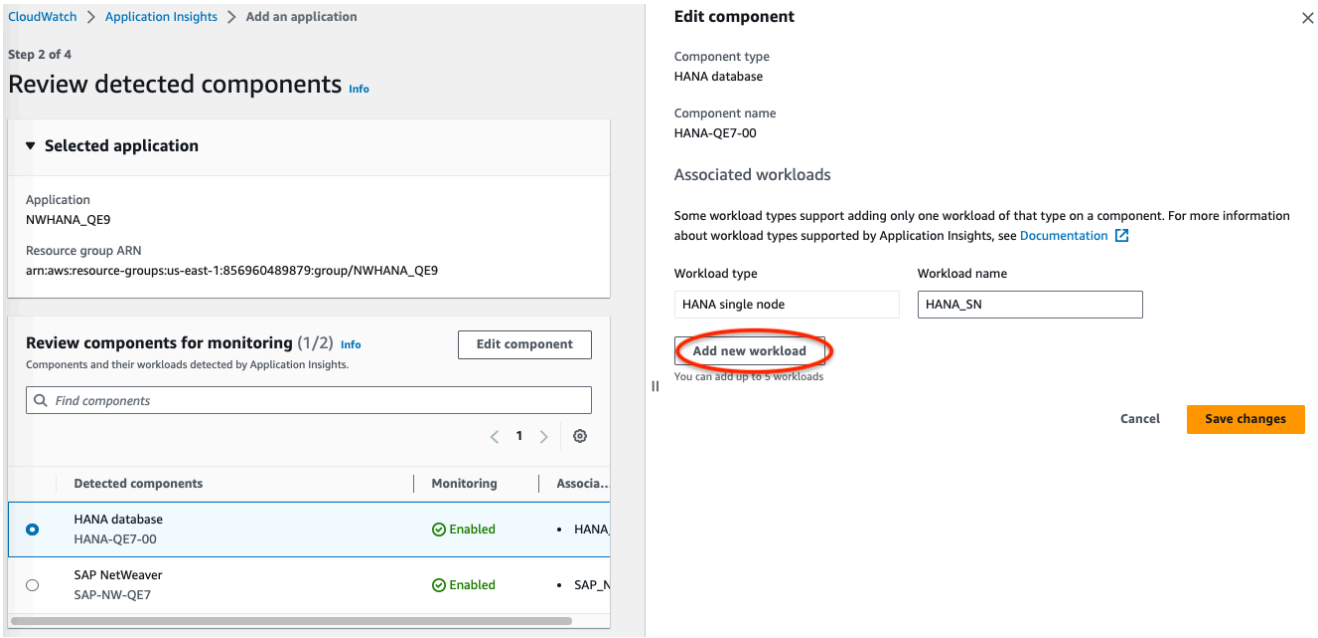
6. Sous Intégrer à AWS Systems Manager OpsCenter, cochez la case à côté de Générer AWS Systems Manager OpsCenter OpsItems pour les actions correctives afin de consulter et de recevoir des notifications lorsque des problèmes sont détectés dans les applications sélectionnées. Pour suivre les opérations effectuées pour résoudre les éléments de travail opérationnels, appelés OpsItems, liés à vos AWS ressources, fournissez un ARN de rubrique SNS.
7. Vous pouvez éventuellement saisir des balises pour vous aider à identifier et à organiser vos ressources. CloudWatch Application Insights prend en charge les groupes de ressources AWS CloudFormation basés sur des balises et des piles, à l'exception des groupes. Application Auto Scaling Pour plus d'informations, consultez Utilisation de Tag Editor dans le Guide de l'utilisateur AWS Resource Groups .
8. Sélectionnez Next (Suivant) pour continuer à mettre en place la surveillance.
9. Sur la page Réviser les composants détectés, les composants surveillés et leurs charges de travail détectées automatiquement par CloudWatch Application Insights sont répertoriés.

    a. Pour ajouter des charges de travail à un composant contenant une charge de travail à nœud simple SAP HANA détectée, sélectionnez le composant, puis choisissez Modifier le composant.

    > **Note**
    >
    > Les composants contenant une charge de travail à plusieurs nœuds SAP HANA ou HANA High Availability détectée ne prennent en charge qu'une seule charge de travail par composant.



b. Pour ajouter une charge de travail, choisissez Ajouter une charge de travail.



c. Lorsque vous avez fini de modifier les charges de travail, choisissez Enregistrer les modifications.

10. Choisissez Suivant.
11. Sur la page Spécifier les détails du composant, saisissez le nom d'utilisateur et le mot de passe.
12. Passez en revue la configuration de surveillance de votre application et sélectionnez Submit (Envoyer).
13. La page de détails de l'application s'ouvre. Vous pouvez consulter le Récapitulatif de l'application, la liste des Composants surveillés et charges de travail surveillés et les Composants et charges de travail non surveillés. Si vous sélectionnez le bouton radio à côté d'un composant ou d'une charge de travail, vous pouvez également afficher l'Historique de configuration, les Modèles de journaux et toutes les Balises que vous avez créées. Lorsque vous envoyez votre configuration, votre compte déploie toutes les métriques et alertes de votre système SAP HANA, qui peuvent prendre jusqu'à 2 heures.

Gérer la surveillance de votre base de données SAP H

Vous pouvez gérer les informations d'identification des utilisateurs, les métriques et les chemins d'accès aux journaux de votre base de données SAP HANA en réalisant les étapes suivantes :

1. Ouvrez la CloudWatch console.
2. Dans le volet de navigation de gauche, sélectionnez Application Insights sous Insights.
3. La page Application Insights affiche la liste des applications qui sont surveillées à l'aide d'Application Insights et l'état de surveillance de chaque application.
4. Sous Monitored components (Composants surveillés), sélectionnez la case d'option à côté du nom du composant. Ensuite, sélectionnez Manage monitoring (Gérer la surveillance).
5. Sous EC2 instance group logs (Journaux des groupes d'instances EC2), vous pouvez mettre à jour le chemin d'accès du journal, le jeu de modèles de journaux et le nom du groupe de journaux existants. En outre, vous pouvez ajouter jusqu'à trois autres Journaux d'application.
6. Sous Metrics (Métriques), vous pouvez choisir les métrique SAP HANA en fonction de vos besoins. Les noms de métriques SAP HANA sont préfixés par `hanadb`. Vous pouvez ajouter jusqu'à 40 métriques par composant.
7. Sous HANA configuration (Configuration HANA), saisissez le mot de passe et le nom d'utilisateur de la base de données SAP HANA. Il s'agit du nom d'utilisateur et du mot de passe utilisés par CloudWatch l'agent Amazon pour se connecter à la base de données SAP HANA.

8. Sous Alarmes personnalisées, vous pouvez ajouter des alarmes supplémentaires à surveiller par CloudWatch Application Insights.
9. Vérifiez la configuration de surveillance de votre application et sélectionnez Submit (Envoyer). Lorsque vous envoyez votre configuration, votre compte met à jour toutes les métriques et les alertes de votre système SAP HANA, ce qui peut prendre jusqu'à 2 heures.

Afficher et résoudre les problèmes SAP HANA détectés par CloudWatch Application Insights

Les sections suivantes fournissent des étapes pour vous aider à résoudre les scénarios de dépannage courants qui se produisent lorsque vous configurez la surveillance pour SAP HANA sur Application Insights.

Résolution des problèmes liés aux rubriques

- La base de données SAP HANA atteint la limite de
- Événement de disque plein
- La sauvegarde SAP HANA a cessé de fonctionner

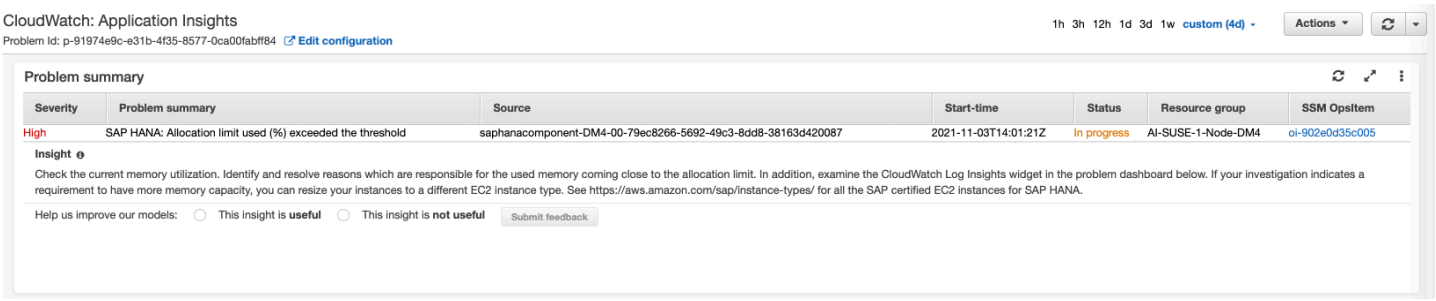
La base de données SAP HANA atteint la limite de

Description

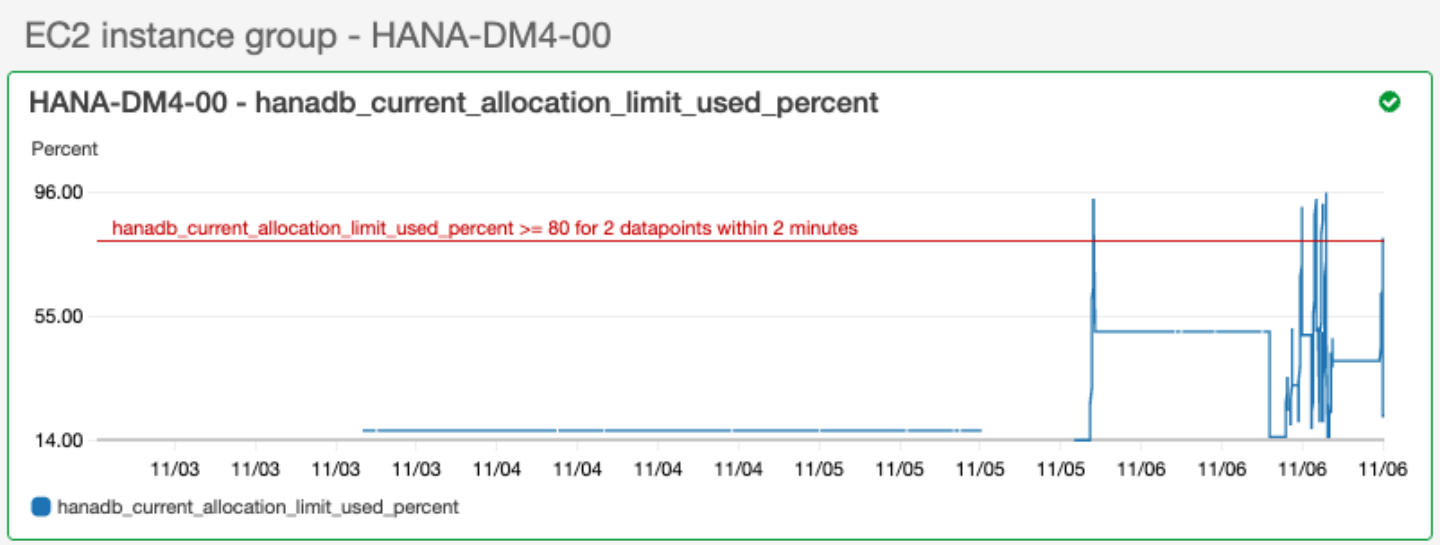
Votre application SAP qui est soutenue par une base de données SAP HANA fonctionne mal en raison d'une pression de mémoire élevée, entraînant une dégradation des performances des applications.

Résolution

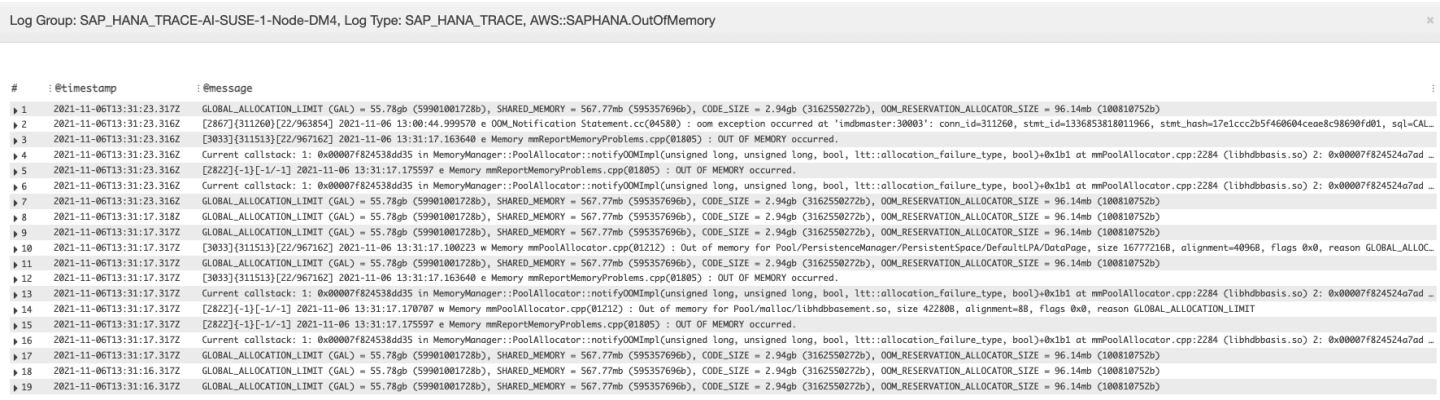
Vous pouvez identifier la couche d'application à l'origine du problème en consultant le tableau de bord créé dynamiquement, qui présente les métriques et les extraits de fichiers journaux correspondants. Dans l'exemple suivant, le problème peut être dû à une charge de données importante dans le système SAP HANA.



L'allocation de mémoire utilisée dépasse le seuil de 80 % de la limite totale d'allocation de mémoire.



Le groupe de journaux affiche le schéma `BNR-DATA` et la table `IMDBMASTER_30003` a manqué de mémoire. En outre, le groupe de journaux affiche l'heure exacte du problème, la limite de localisation globale actuelle, la mémoire partagée, la taille du code et la taille de l'allocation de réservation OOM.



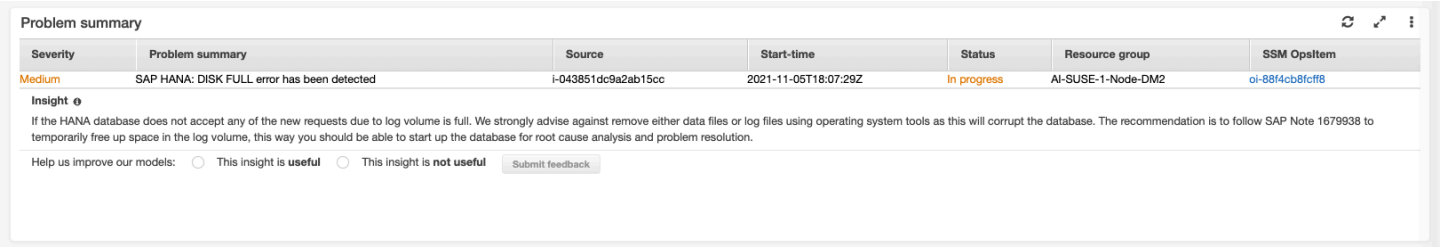
Événement de disque plein

Description

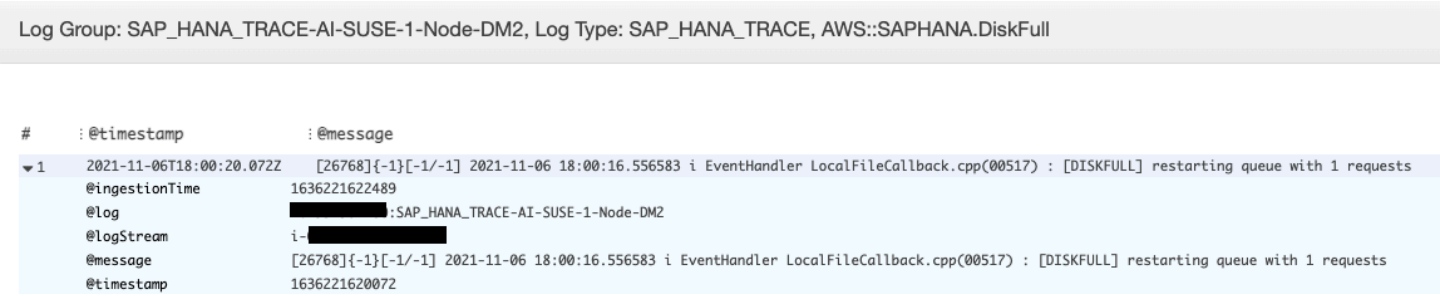
Votre application SAP qui est soutenue par une base de données SAP HANA cesse de répondre, ce qui entraîne une incapacité d'accéder à la base de données.

Résolution

Vous pouvez identifier la couche de base de données à l'origine du problème en consultant le tableau de bord créé dynamiquement, qui présente les métriques et les extraits de fichiers journaux correspondants. Dans l'exemple suivant, le problème peut être que l'administrateur n'a pas réussi à activer la sauvegarde automatique des journaux, ce qui a provoqué le remplissage du répertoire sap/hana/log.



Le widget groupe de journaux dans le tableau de bord des problèmes affiche l'événement `DISKFULL`.



La sauvegarde SAP HANA a cessé de fonctionner

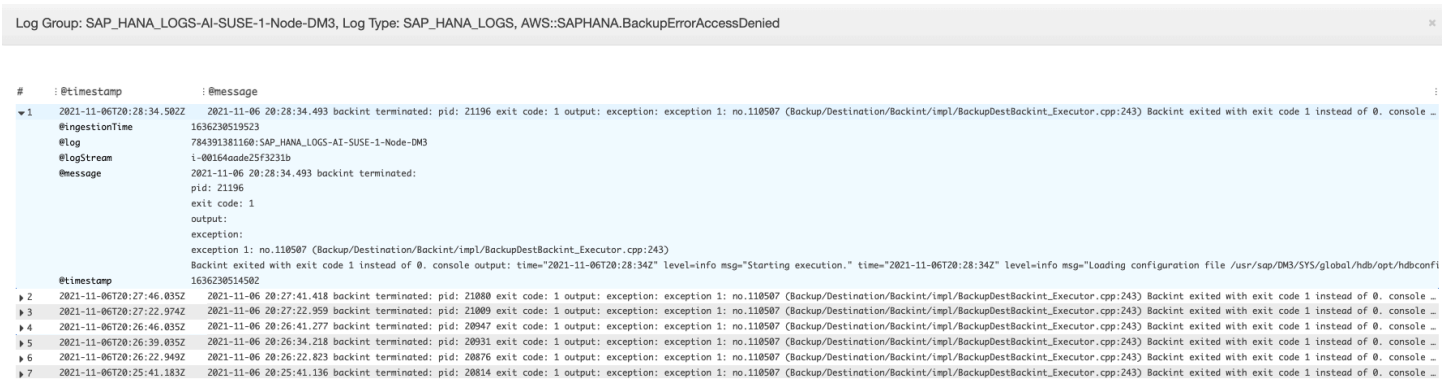
Description

Votre application SAP qui est soutenue par une base de données SAP HANA a cessé de fonctionner.

Résolution

Vous pouvez identifier la couche de base de données à l'origine du problème en consultant le tableau de bord créé dynamiquement, qui présente les métriques et les extraits de fichiers journaux correspondants.

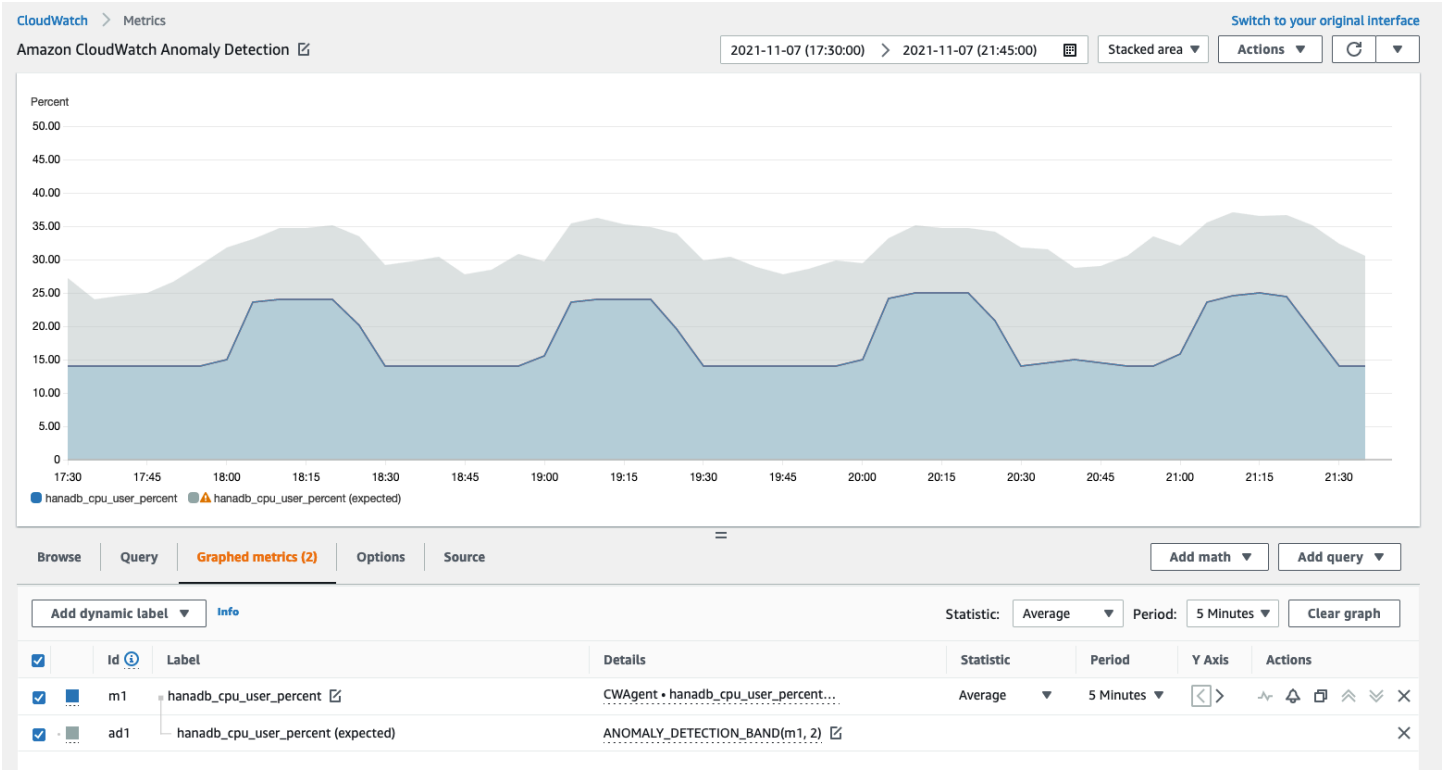
Le widget groupe de journaux dans le tableau de bord des problèmes affiche l'événement `ACCESS DENIED`. Cela comprend des informations supplémentaires, telles que le compartiment S3, le dossier de compartiment S3 et la région du compartiment S3.



Détection d'anomalies pour SAP HANA

Pour des métriques SAP HANA spécifiques, telles que le nombre de threads, CloudWatch applique des algorithmes statistiques et d'apprentissage automatique pour définir le seuil. Ces algorithmes analysent en permanence les métriques de la base de données SAP HANA, déterminent les lignes de base normales et font apparaître les anomalies avec une intervention minimale de l'utilisateur. Les algorithmes génèrent un modèle de détection des anomalies, qui génère une gamme de valeurs attendues représentant le comportement normal de la métrique.

Les algorithmes de détection des anomalies tiennent compte de la saisonnalité et des changements de tendance des métriques. Les changements saisonniers peuvent être horaires, quotidiens ou hebdomadaires, comme le montrent les exemples suivants de l'utilisation du CPU de SAP HANA.



Une fois que vous avez créé un modèle, la détection des CloudWatch anomalies évalue le modèle en permanence et l'ajuste pour s'assurer qu'il est aussi précis que possible. Cela comprend le recyclage du modèle pour ajuster si les valeurs de métrique évoluent au fil du temps ou subissent des changements brusques. cela comprend également des prédicteurs pour améliorer les modèles de métriques saisonnières, pointues ou clairsemées.

Dépannage des applications Insights pour SAP HANA

Cette section fournit des étapes pour vous aider à résoudre les erreurs courantes renvoyées par le tableau de bord Application Insights.

Impossible d'ajouter plus de 60 métriques surveillées

La sortie affiche l'erreur suivante.

```
Component cannot have more than 60 monitored metrics
```

Cause première : la limite de mesures actuelle est de 60 mesures surveillées par composant.

Résolution — Pour rester en deçà de la limite, supprimez les métriques inutiles.

Aucune SAP métrique n'apparaît après le processus d'intégration

Utilisez les informations suivantes pour découvrir pourquoi les métriques SAP n'apparaissent pas sur le tableau de bord après le processus d'intégration. La première étape consiste à déterminer pourquoi les métriques SAP n'apparaissent pas à l'aide des journaux AWS Management Console ou de l'exportateur d'une instance Amazon EC2. Passez ensuite en revue le résultat d'erreur pour trouver une solution.

Résoudre les problèmes liés au fait que les métriques SAP n'apparaissent pas après l'intégration

Vous pouvez utiliser les journaux AWS Management Console ou les journaux d'exportation d'une instance Amazon EC2 pour le dépannage.

AWS Management Console

Résolution des problèmes : aucune métrique SAP n'apparaît après l'intégration à l'aide de la console

1. Ouvrez la AWS Systems Manager console à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le volet de navigation de gauche, choisissez State Manager.

3. Sous Associations, vérifiez le statut du documentAWSEC2-ApplicationInsightsCloudwatchAgentInstallAndConfigure. Si le statut estFailed, sous ID d'exécution, sélectionnez l'identifiant d'échec et visualisez le résultat.
4. Sous Associations, vérifiez le statut du documentAWS-ConfigureAWSPackage. Si le statut estFailed, sous ID d'exécution, sélectionnez l'identifiant d'échec et visualisez le résultat.

Exporter logs from Amazon EC2 instance

Résolution des problèmes : aucune métrique SAP n'apparaît après l'intégration à l'aide des journaux d'exportation

1. Connectez-vous à l'instance Amazon EC2 sur laquelle s'exécute votre base de données SAP HANA.
2. Trouvez la convention de dénomination appropriée pour WORKLOAD_SHORT_NAME utiliser la commande suivante. Vous utiliserez ce nom abrégé dans les deux étapes suivantes.

```
sudo systemctl | grep exporter
```

> **Note**
>
> Application Insights ajoute un suffixe WORKLOAD_SHORT_NAME au nom du service en fonction de la charge de travail en cours d'exécution. Les noms abrégés des déploiements à nœud unique, à nœuds multiples et à haute disponibilité de SAP HANA sont HANA_SNHANA_MN, et. HANA_HA

3. Pour vérifier l'absence d'erreurs dans les journaux de service de l'Exporter Manager, exécutez la commande suivante en WORKLOAD_SHORT_NAME remplaçant par le nom abrégé que vous y avez trouvéStep 2.

```
sudo journalctl -e --unit=prometheus-
hanadb_exporter_manager_WORKLOAD_SHORT_NAME.service
```

4. Si les journaux de service du gestionnaire de l'exportateur n'affichent aucune erreur, vérifiez qu'il n'y a pas d'erreur dans les journaux de service de l'exportateur en exécutant la commande suivante.

```
sudo journalctl -e --unit=prometheus-hanadb_exporter_WORKLOAD_SHORT_NAME.service
```

Résolution des causes profondes courantes pour lesquelles les métriques SAP n'apparaissent pas après l'intégration

Les exemples suivants décrivent comment résoudre les causes profondes courantes qui font que les métriques SAP n'apparaissent pas après l'intégration.

- La sortie affiche l'erreur suivante.

```
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/amazon-
cloudwatch-agent.d/default ...
Reading json config file path: /opt/aws/amazon-cloudwatch-agent/etc/
amazon-cloudwatch-agent.d/ssm_AmazonCloudWatch-ApplicationInsights-
SSMParameterForTESTCWEC2INSTANCEi0d88867f1f3e36285.tmp ...
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Failed to merge multiple json config files.
2023/11/30 22:25:17 Under path : /metrics/append_dimensions | Error : Different
 values are specified for append_dimensions
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/disk | Error : Different
 values are specified for disk
2023/11/30 22:25:17 Under path : /metrics/metrics_collected/mem | Error : Different
 values are specified for mem
2023/11/30 22:25:17 Configuration validation first phase failed. Agent version: 1.0.
 Verify the JSON input is only using features supported by this version.
```

  Résolution — Application Insights essaie de configurer les mêmes métriques que celles préconfigurées dans le fichier de configuration de l' CloudWatch agent existant. Supprimez les fichiers existants `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.d/` ou supprimez les métriques à l'origine du conflit dans le fichier de configuration de l' CloudWatch agent existant.

- La sortie affiche l'erreur suivante.

```
Unable to find a host with system database, for more info rerun using -v
```

  Résolution — Le nom d'utilisateur, le mot de passe ou le port de base de données sont peut-être incorrects. Vérifiez que le nom d'utilisateur, le mot de passe et le port sont valides, puis relancez le processus d'intégration.

- La sortie affiche l'erreur suivante.

```
This hdbcli installer is not compatible with your Python interpreter
```

Résolution — Mettez à niveau pip3 et wheel comme indiqué dans l'exemple suivant pour Python 3.6.

```
python3.6 -m pip install --upgrade pip setuptools wheel
```

- La sortie affiche l'erreur suivante.

```
Unable to install hdbcli using pip3. Please try to install it
```

Résolution — Assurez-vous d'avoir suivi les `hdbclient` prérequis ou effectuez l'installation `hdbclient` manuellement sous pip3.

- La sortie affiche l'erreur suivante.

```
Package 'boto3' requires a different Python: 3.6.15 not in '>= 3.7'
```

Résolution — Python 3.8 ou supérieur est requis pour cette version du système d'exploitation. Vérifiez les prérequis pour Python 3.8 et installez-le.

- Le résultat indique l'une des erreurs d'installation suivantes.

```
Can not execute `setup.py` since setuptools is not available in the build environment
```

or

```
[SSL: CERTIFICATE_VERIFY_FAILED]
```

Résolution — Installez Python à l'aide des commandes SUSE Linux, comme indiqué dans l'exemple suivant. L'exemple suivant installe la dernière version de Python 3.8.

```
wget https://www.python.org/ftp/python/3.8.<LATEST_RELEASE>/
Python-3.8.<LATEST_RELEASE>.tgz
tar xf Python-3.*
cd Python-3.*/
sudo zypper install make gcc-c++ gcc automake autoconf libtool
sudo zypper install zlib-devel
sudo zypper install libopenssl-devel libffi-devel
./configure --with-ensurepip=install
sudo make
sudo make install
```

```
sudo su
python3.8 -m pip install --upgrade pip setuptools wheel
```

Tutoriel : Configuration de la surveillance pour SAP NetWeaver

Ce didacticiel explique comment configurer Amazon CloudWatch Application Insights pour configurer la surveillance pour SAP NetWeaver. Vous pouvez utiliser les tableaux de bord automatiques d' CloudWatch Application Insights pour visualiser les détails des problèmes, accélérer le dépannage et réduire le temps moyen de résolution (MTTR) pour vos serveurs d'applications SAP NetWeaver.

CloudWatch NetWeaver Sujets relatifs aux informations sur les applications pour SAP

- Environnements compatibles
- Systèmes d’exploitation pris en charge
- Fonctionnalités
- Prérequis
- Configurez vos serveurs NetWeaver d'applications SAP pour la surveillance
- Gérez la surveillance de vos serveurs NetWeaver d'applications SAP
- Afficher et résoudre les NetWeaver problèmes SAP détectés par CloudWatch Application Insights
- Résolution des problèmes liés aux applications pour SAP NetWeaver

Environnements compatibles

CloudWatch Application Insights prend en charge le déploiement de AWS ressources pour les systèmes et modèles suivants.

- Déploiement du système NetWeaver standard SAP.
- Déploiements SAP NetWeaver Distributed sur plusieurs instances Amazon EC2.
- Configuration de NetWeaver haute disponibilité SAP inter-AZ : SAP NetWeaver avec haute disponibilité configuré sur deux zones de disponibilité à l'aide du clustering SUSE/RHEL.

Systèmes d’exploitation pris en charge

CloudWatch Application Insights for SAP NetWeaver est compatible avec les systèmes d'exploitation suivants :

- Oracle Linux 8
- Red Hat Enterprise Linux 7.6
- Red Hat Enterprise Linux 7.7
- Red Hat Enterprise Linux 7.9
- Red Hat Enterprise Linux 8.1
- Red Hat Enterprise Linux 8.2
- Red Hat Enterprise Linux 8.4
- Red Hat Enterprise Linux 8.6
- SUSE Linux Enterprise Server 15 pour SAP
- SUSE Linux Enterprise Server 15 SP1 pour SAP
- SUSE Linux Enterprise Server 15 SP2 pour SAP
- SUSE Linux Enterprise Server 15 SP3 pour SAP
- SUSE Linux Enterprise Server 15 SP4 pour SAP
- SUSE Linux Enterprise Server 12 SP4 pour SAP
- SUSE Linux Enterprise Server 12 SP5 pour SAP
- SUSE Linux Enterprise Server 15, à l'exception des modèles de haute disponibilité
- SUSE Linux Enterprise Server 15 SP1, à l'exception des modèles de haute disponibilité
- SUSE Linux Enterprise Server 15 SP2, à l'exception des modèles de haute disponibilité
- SUSE Linux Enterprise Server 15 SP3, à l'exception des modèles de haute disponibilité
- SUSE Linux Enterprise Server 15 SP4, à l'exception des modèles de haute disponibilité
- SUSE Linux Enterprise Server 12 SP4, à l'exception des modèles de haute disponibilité
- SUSE Linux Enterprise Server 12 SP5, à l'exception des modèles de haute disponibilité

Fonctionnalités

CloudWatch Application Insights for SAP NetWeaver 7.0x—7.5x (y compris ABAP Platform) fournit les fonctionnalités suivantes :

- Détection automatique de la NetWeaver charge de travail SAP
- Création automatique NetWeaver d'alarmes SAP sur la base de seuils statiques
- Reconnaissance automatique du modèle de NetWeaver journal SAP
- Tableau de bord Health pour SAP NetWeaver
- Tableau de bord des problèmes pour SAP NetWeaver

Prérequis

Vous devez remplir les conditions préalables suivantes pour configurer SAP NetWeaver avec CloudWatch Application Insights :

- AWS Activation de Systems Manager : installez l'agent SSM sur vos instances Amazon EC2 et activez les instances pour SSM. Pour plus d'informations sur l'installation de l'Agent SSM, veuillez consulter la rubrique Configuration de AWS Systems Manager dans le Guide de l'utilisateur AWS Systems Manager.
- Rôles d'instance Amazon EC2 : vous devez associer les rôles d'instance Amazon EC2 suivants pour configurer votre surveillance SAP. NetWeaver
  - Vous devez joindre le rôle `AmazonSSMManagedInstanceCore` pour activer Systems Manager. Pour plus d'informations, consultez AWS Systems Manager Exemples de politiques basées sur l'identité.
  - Vous devez joindre la `CloudWatchAgentServerPolicy` politique pour permettre l'émission des métriques et des journaux de l'instance CloudWatch. Pour plus d'informations, voir Création de rôles et d'utilisateurs IAM à utiliser avec l' CloudWatch agent.
- AWS groupes de ressources : vous devez créer un groupe de ressources qui inclut toutes les AWS ressources associées utilisées par votre pile d'applications pour intégrer vos applications à CloudWatch Application Insights. Cela inclut les instances Amazon EC2, les volumes Amazon EFS et Amazon EBS exécutant vos serveurs d'applications SAP NetWeaver . S'il existe plusieurs NetWeaver systèmes SAP par compte, nous vous recommandons de créer un groupe de ressources qui inclut les AWS ressources de chaque NetWeaver système SAP. Pour plus d'informations sur la création de groupes de ressources, consultez le Guide de l'utilisateur des groupes et des balises de ressources AWS (français non garanti).
- Autorisations IAM : pour les utilisateurs qui n'ont pas d'accès administratif, vous devez créer une politique AWS Identity and Access Management (IAM) qui permet à Application Insights de créer un rôle lié à un service et de l'associer à l'identité de l'utilisateur. Pour plus d'informations sur la façon de créer la politique IAM, consultez Politique IAM.

- Rôle lié à un service — Application Insights utilise des rôles liés à un service AWS Identity and Access Management (IAM). Un rôle lié à un service est créé pour vous lorsque vous créez votre première application Application Insights dans la console Application Insights. Pour plus d'informations, consultez Utilisation de rôles liés à un service pour Application Insights CloudWatch .
- Amazon CloudWatch agent — Application Insights installe et configure l' CloudWatch agent. Si CloudWatch l'agent est installé, Application Insights conserve votre configuration. Pour éviter un conflit de fusion, supprimez la configuration des ressources que vous souhaitez utiliser dans Application Insights du fichier de configuration de l' CloudWatch agent existant. Pour plus d'informations, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

Configurez vos serveurs NetWeaver d'applications SAP pour la surveillance

Suivez les étapes ci-dessous pour configurer la surveillance de vos serveurs NetWeaver d'applications SAP.

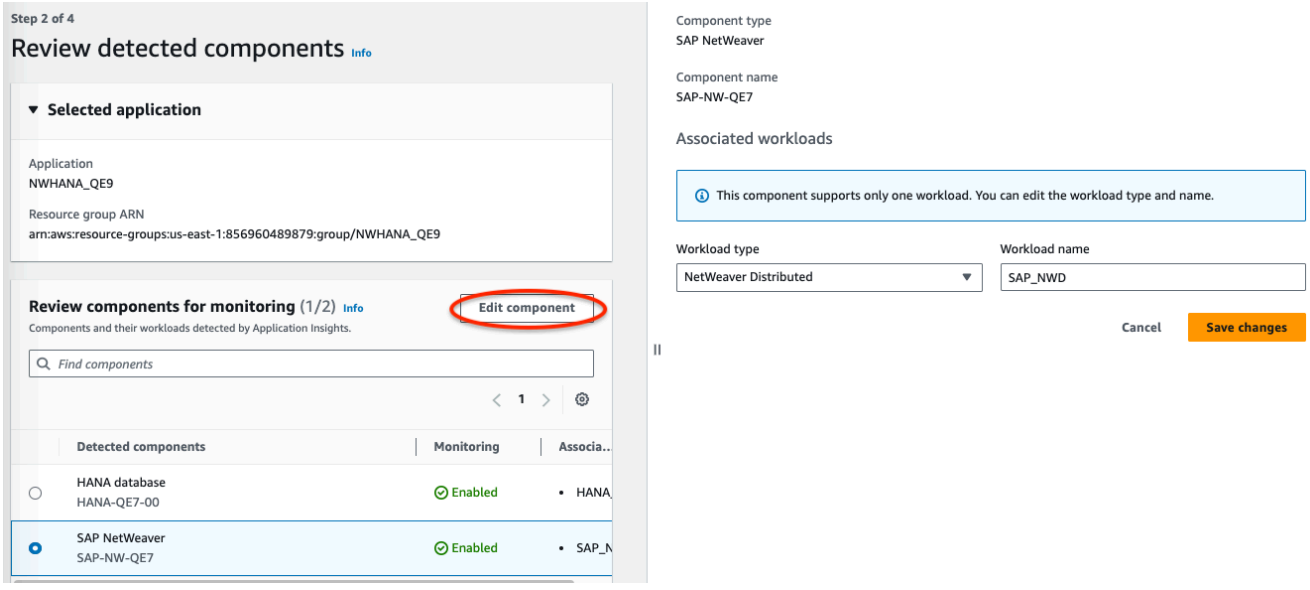
Pour configurer la surveillance

1. Ouvrez la CloudWatch console.
2. Dans le panneau de navigation de gauche, sélectionnez Application Insights sous Insights.
3. La page Application Insights affiche la liste des applications qui sont surveillées à l'aide d'Application Insights et l'état de surveillance de chaque application. Dans le coin supérieur droit, sélectionnez Add an application (Ajouter une application).
4. Sur la page Spécifier les détails de l'application, dans la liste déroulante sous Groupe de ressources, sélectionnez le groupe de AWS ressources que vous avez créé et qui contient vos NetWeaver ressources SAP. Si vous n'avez pas encore créé de groupe de ressources pour votre application, vous pouvez le faire en sélectionnant Create new resource group (Créer un groupe de ressources) dans la liste déroulante Resource Group (Groupe de ressources).
5. Sous Automatic monitoring of new resources (Surveillance automatique des nouvelles ressources), cochez la case pour permettre à Application Insights de surveiller automatiquement les ressources qui sont ajoutées au groupe de ressources de l'application après l'intégration.
6. Sous Surveiller les EventBridge événements, cochez la case pour intégrer la surveillance des informations d'application aux CloudWatch événements afin d'obtenir des informations provenant d'Amazon EBS, d'Amazon EC2 AWS CodeDeploy, d'Amazon ECS AWS Health , des API et des notifications, d'Amazon RDS, d'Amazon S3 et. AWS Step Functions

7. Sous Intégrer à AWS Systems Manager OpsCenter, cochez la case à côté de Générer AWS Systems Manager OpsCenter OpsItems pour les actions correctives afin de consulter et de recevoir des notifications lorsque des problèmes sont détectés dans les applications sélectionnées. Pour suivre les opérations effectuées pour résoudre les éléments de travail opérationnels, appelés OpsItems, liés à vos AWS ressources, fournissez un ARN de rubrique SNS.
8. Vous pouvez éventuellement saisir des balises pour vous aider à identifier et à organiser vos ressources. CloudWatch Application Insights prend en charge les groupes de ressources AWS CloudFormation basés sur des balises et des piles, à l'exception des groupes. Application Auto Scaling Pour plus d'informations, consultez Utilisation de Tag Editor dans le Guide de l'utilisateur AWS Resource Groups .
9. Pour examiner les composants détectés, choisissez Suivant.
10. Sur la page Réviser les composants détectés, les composants surveillés et leurs charges de travail détectées automatiquement par CloudWatch Application Insights sont répertoriés.

    - Pour modifier le type et le nom de la charge de travail, choisissez Modifier le composant.

> **Note**
>
> Les composants contenant une charge de travail NetWeaver distribuée ou à NetWeaver haute disponibilité détectée ne prennent en charge qu'une seule charge de travail par composant.



11. Choisissez Suivant.
12. Sur la page Specify component details (Spécifier les détails du composant), cliquez sur Next (Suivant).
13. Passez en revue la configuration de surveillance de votre application, puis sélectionnez Soumettre.
14. La page de détails de l'application s'ouvre, où vous pouvez afficher le Récaputilatif de l'application, le Tableau de bord, les Composants et les Charges de travail. Vous pouvez également afficher la Configuration history (Historique de la configuration), les Log patterns (Modèles de journaux) et les Tags (Balises) que vous avez créés. Une fois que vous avez soumis votre CloudWatch candidature, Application Insights déploie toutes les mesures et alarmes de votre NetWeaver système SAP, ce qui peut prendre jusqu'à une heure.

Gérez la surveillance de vos serveurs NetWeaver d'applications SAP

Suivez les étapes ci-dessous pour gérer la surveillance de vos serveurs NetWeaver d'applications SAP.

Pour gérer la surveillance

1. Ouvrez la CloudWatch console.
2. Dans le panneau de navigation de gauche, sélectionnez Application Insights sous Insights.
3. Choisissez l'onglet List view (Vue en liste).
4. La page Application Insights affiche la liste des applications qui sont surveillées à l'aide d'Application Insights et l'état de surveillance de chaque application.
5. Sélectionnez votre application.
6. Choisissez l'onglet Components (Composants).
7. Sous Monitored components (Composants surveillés), sélectionnez la case d'option à côté du nom du composant. Ensuite, sélectionnez Manage monitoring (Gérer la surveillance).
8. Sous Instance logs (Journaux d'instance), vous pouvez mettre à jour le chemin d'accès du journal, le jeu de modèles de journaux et le nom du groupe de journaux existants. En outre, vous pouvez ajouter jusqu'à trois autres Journaux d'application.
9. Sous Métriques, vous pouvez sélectionner les NetWeaver métriques SAP en fonction de vos besoins. Les noms des NetWeaver métriques SAP sont préfixés par`sap`. Vous pouvez ajouter jusqu'à 40 métriques par composant.

10. Sous Alarmes personnalisées, vous pouvez ajouter des alarmes supplémentaires à surveiller par CloudWatch Application Insights.
11. Vérifiez la configuration de surveillance de votre application et sélectionnez Save (Enregistrer). Lorsque vous soumettez votre configuration, votre compte met à jour toutes les mesures et alarmes de vos NetWeaver systèmes SAP.

Afficher et résoudre les NetWeaver problèmes SAP détectés par CloudWatch Application Insights

Les sections suivantes fournissent des étapes pour vous aider à résoudre les scénarios de dépannage courants qui se produisent lorsque vous configurez la surveillance pour SAP NetWeaver sur Application Insights.

Résolution des problèmes liés aux rubriques

- Problèmes de connectivité aux NetWeaver bases de données SAP
- Problèmes de disponibilité des NetWeaver applications SAP

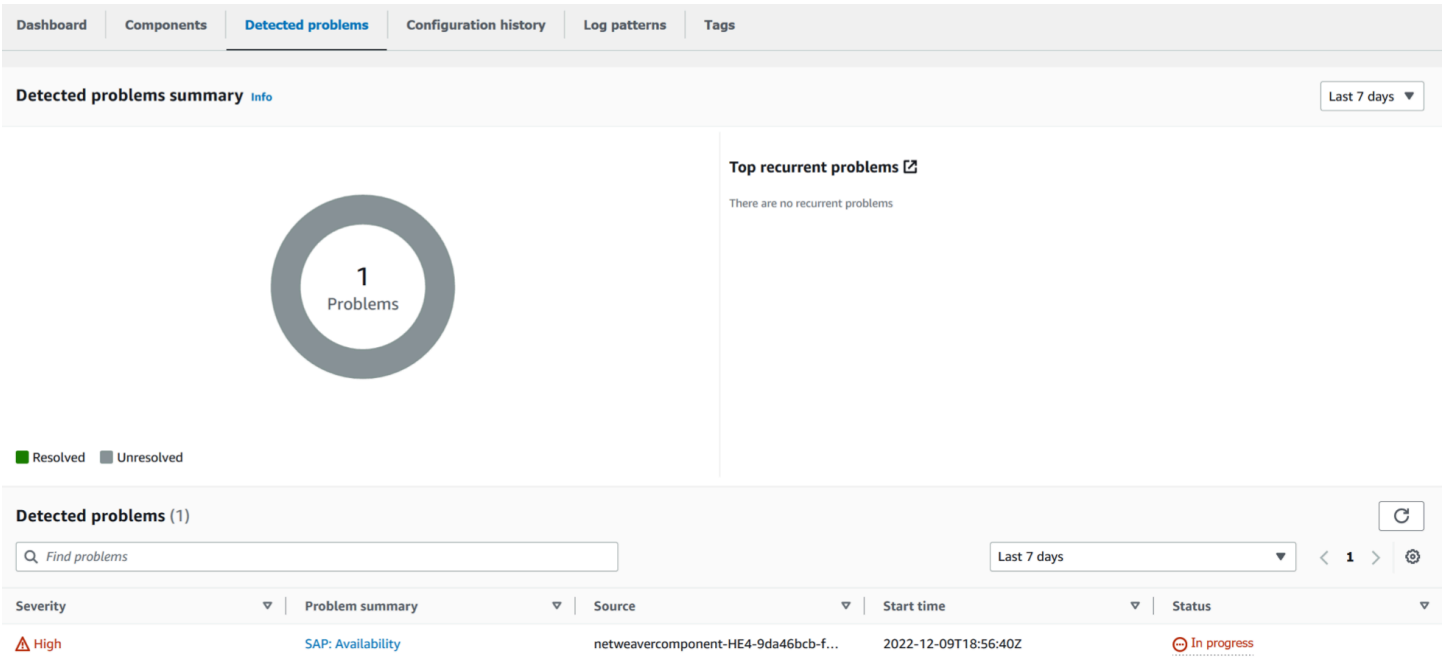
Problèmes de connectivité aux NetWeaver bases de données SAP

Description

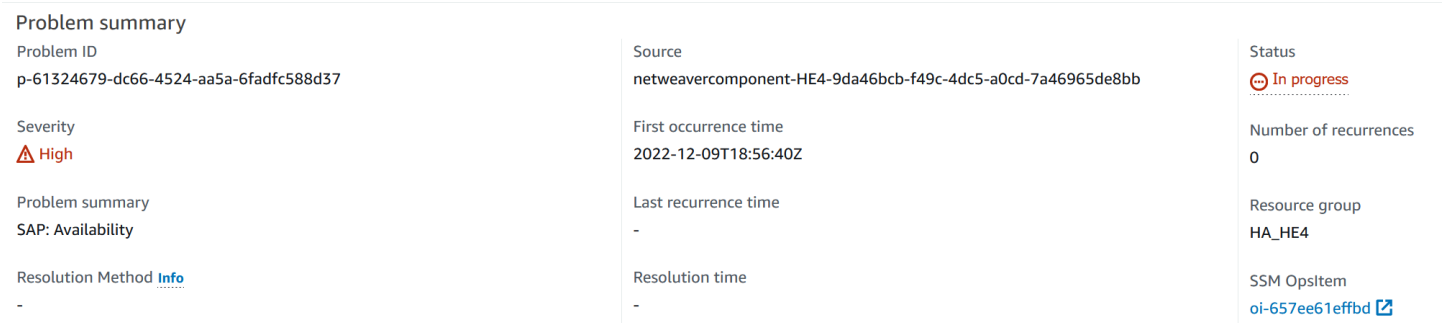
Votre NetWeaver application SAP rencontre des problèmes de connectivité à la base de données.

Cause

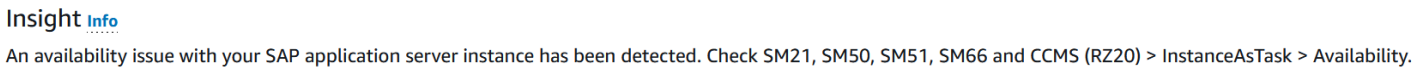
Vous pouvez identifier le problème de connectivité en accédant à la console CloudWatch Application Insights et en consultant le tableau de bord des problèmes de SAP NetWeaver Application Insights. Sélectionnez le lien sous Problem summary (Résumé du problème) pour voir le problème spécifique.



Dans l'exemple suivant, sous Problem summary (Résumé du problème), SAP : la disponibilité est le problème.

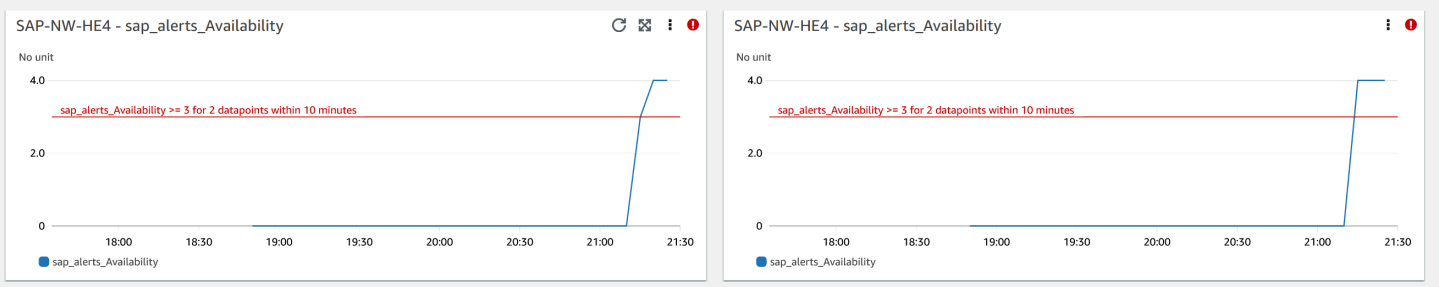


Immédiatement après Problem summary (Résumé du problème), la section Insight (Information) fournit plus de contexte sur l'erreur et indique où vous pouvez obtenir plus d'informations sur les causes du problème.

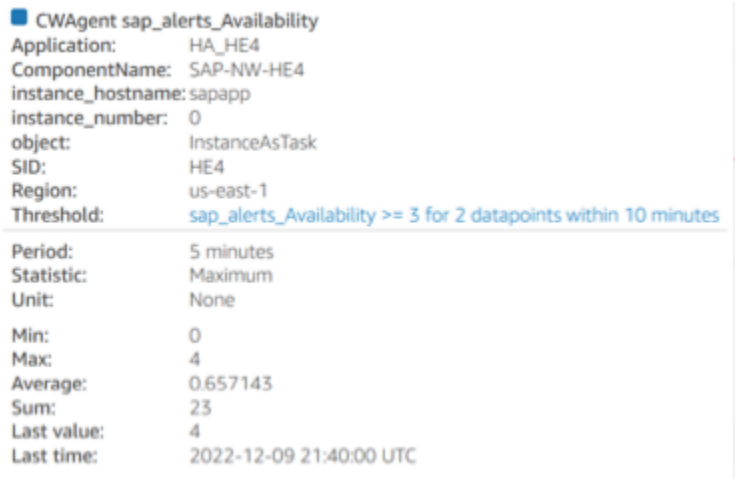


Sur le même tableau de bord du problème, vous pouvez afficher les journaux et métriques connexes que la détection des problèmes a regroupés pour vous aider à isoler la cause de l'erreur. La `sap_alerts_Availability` métrique suit la disponibilité du NetWeaver système SAP au fil du temps. Vous pouvez utiliser le suivi historique pour établir une corrélation entre le moment où la métrique a déclenché un état d'erreur ou a franchi le seuil d'alarme. Dans l'exemple suivant, le

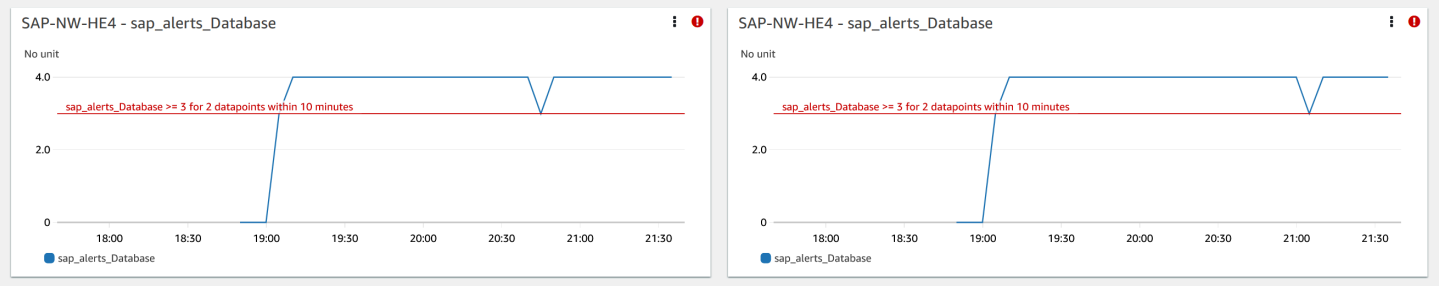
NetWeaver système SAP présente un problème de disponibilité. L'exemple montre deux alarmes car il y a deux instances de serveur d'application SAP et une alarme a été créée pour chaque instance.



Pour plus d'informations sur chaque alarme, survolez le nom de la métrique `sap_alerts_Availability`.



Dans l'exemple suivant, la métrique `sap_alerts_Database` indique que la couche de base de données présente un problème ou une défaillance. Cette alarme indique que SAP NetWeaver a rencontré des problèmes de connexion ou de communication avec sa base de données.



La base de données étant une ressource clé pour SAP NetWeaver, vous pouvez recevoir de nombreuses alarmes associées en cas de problème ou de défaillance de la base de données. Dans l'exemple suivant, les métriques `sap_alerts_FrontendResponseTime` et `sap_alerts_LongRunners` sont déclenchées, car la base de données n'est pas disponible.

Résolution

Application Insights surveille le problème détecté toutes les heures. S'il n'y a aucune nouvelle entrée de journal associée dans vos fichiers NetWeaver journaux SAP, les anciennes entrées de journal seront considérées comme résolues. Vous devez corriger toutes les conditions d'erreur liées aux CloudWatch alarmes. Une fois les conditions d'erreur corrigées, l'alarme est résolue lorsque les alarmes et les journaux sont récupérés. Lorsque toutes les erreurs du CloudWatch journal et les alarmes sont résolues, Application Insights cesse de détecter les erreurs et le problème est automatiquement résolu en moins d'une heure. Nous vous recommandons de résoudre toutes les conditions d'erreurs de journal et les alarmes afin de disposer des derniers problèmes sur le tableau de bord des problèmes.

Dans l'exemple suivant, le problème de disponibilité de SAP est résolu.

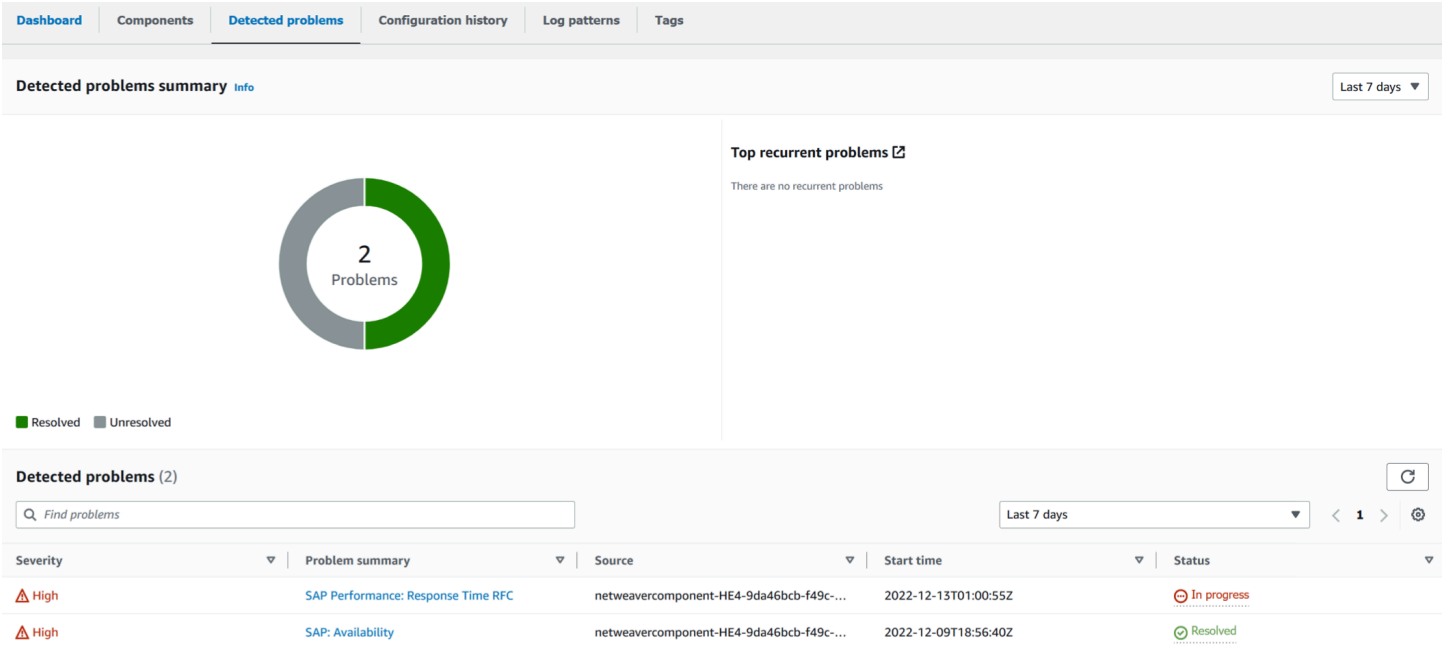
Problèmes de disponibilité des NetWeaver applications SAP

Description

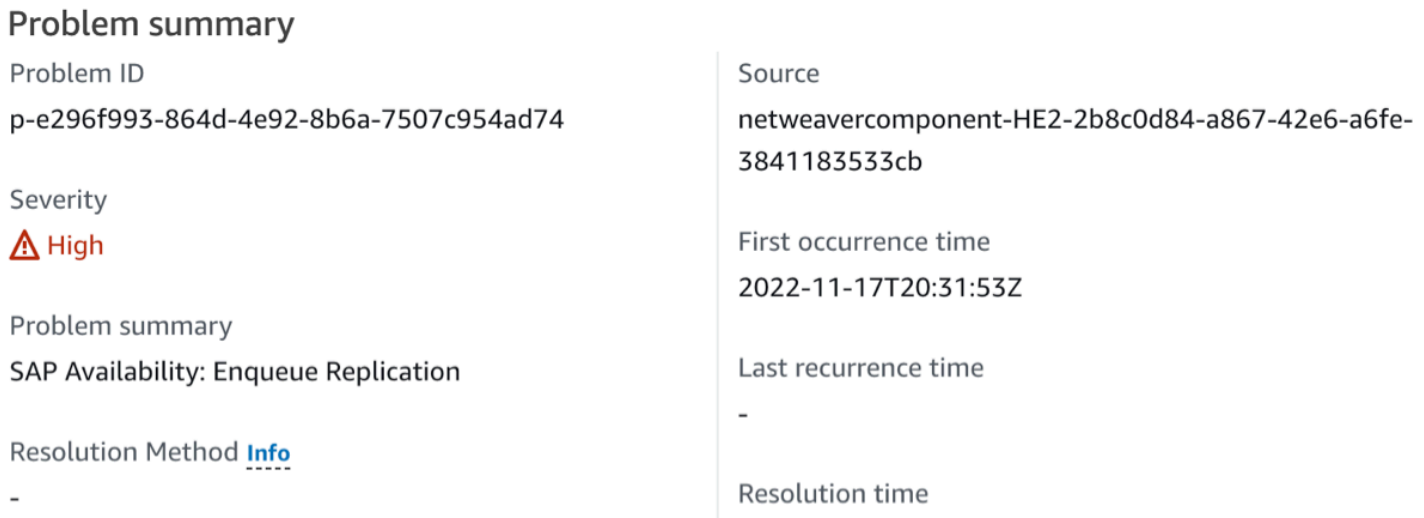
Votre réplication SAP NetWeaver High Availability Queue a cessé de fonctionner.

Cause

Vous pouvez identifier le problème de connectivité en accédant à la console CloudWatch Application Insights et en consultant le tableau de bord des problèmes de SAP NetWeaver Application Insights. Sélectionnez le lien sous Problem summary (Résumé du problème) pour voir le problème spécifique.



Dans l'exemple suivant, sous Problem summary (Résumé du problème), la réplication High Availability Enqueue est le problème.

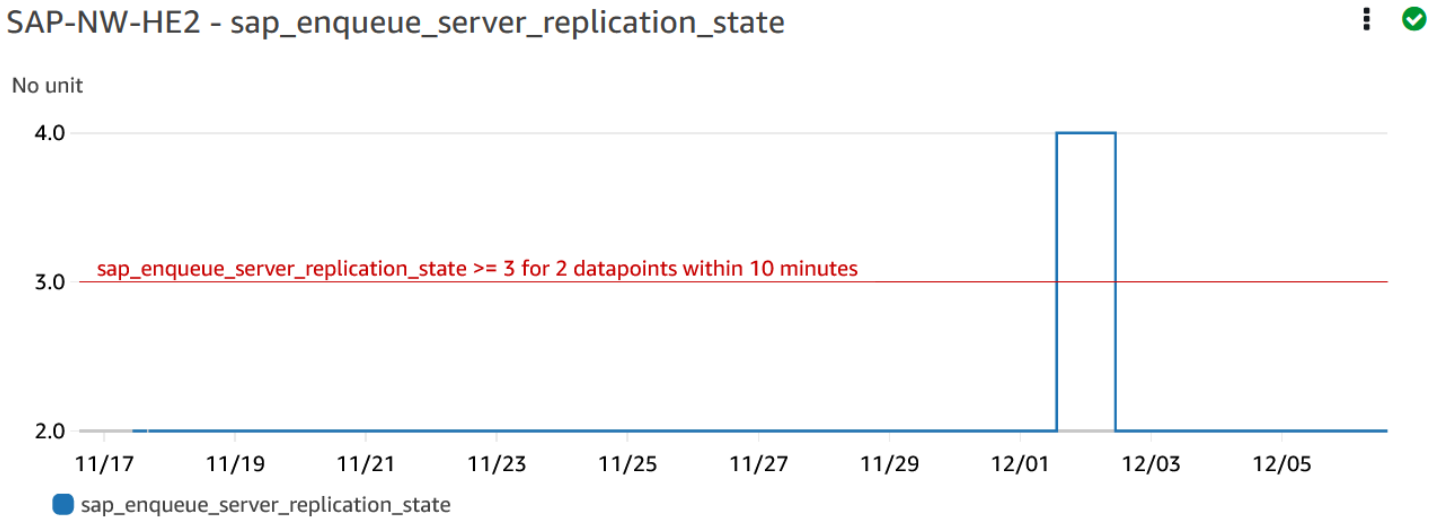


Immédiatement après Problem summary (Résumé du problème), la section Insight (Information) fournit plus de contexte sur l'erreur et indique où vous pouvez obtenir plus d'informations sur les causes du problème.

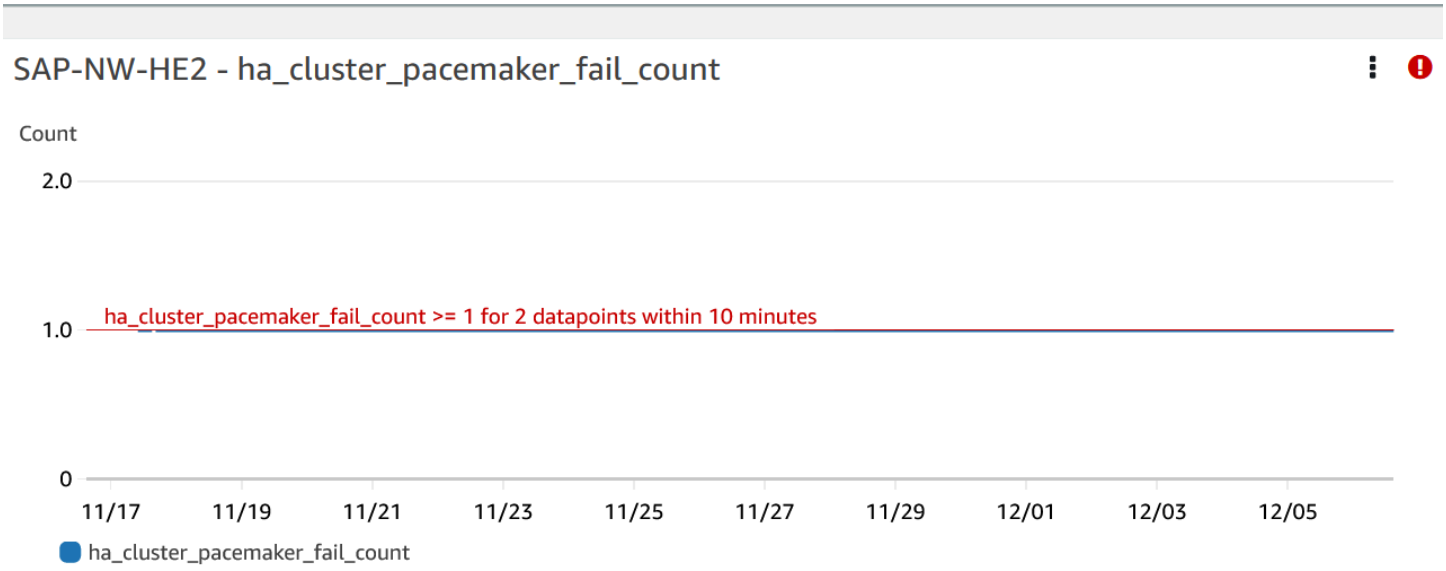
Insight Info

An issue with your SAP enqueue replication (ERS) state has been detected. Check that your enqueue replication is working with SAP transactions, such as SMENQ or the ensmon command.

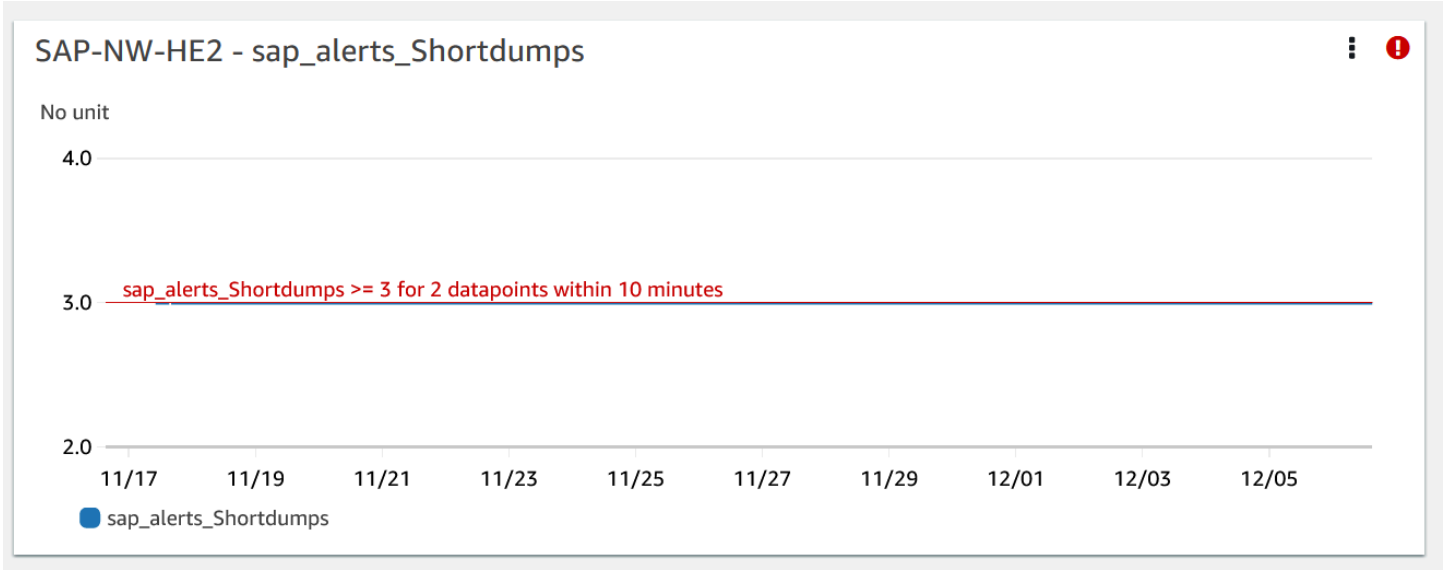
L'exemple suivant montre le tableau de bord du problème où vous visualisez des journaux et des métriques qui sont regroupés pour vous aider à isoler les causes de l'erreur. La métrique `sap_enqueue_server_replication_state` suit la valeur dans le temps. Vous pouvez utiliser le suivi historique pour établir une corrélation entre le moment où la métrique a déclenché un état d'erreur ou a franchi le seuil d'alarme.



Dans l'exemple suivant, la métrique `ha_cluster_pacemaker_fail_count` montre que le cluster pacemaker haute disponibilité a connu une défaillance de ressource. Les ressources spécifiques du pacemaker qui ont eu un nombre d'échecs supérieur ou égal à un sont identifiées dans le tableau de bord des composants.

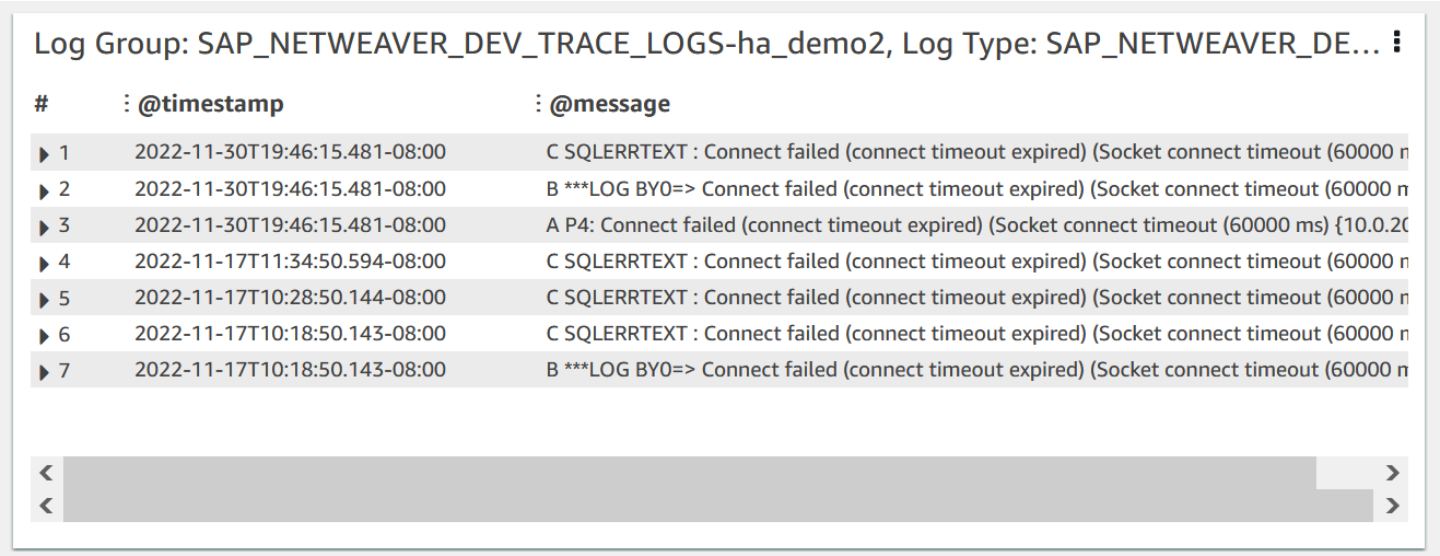


L'exemple suivant montre la métrique `sap_alerts_Shortdumps`, qui indique que les performances de l'application SAP étaient réduites lorsque le problème a été détecté.

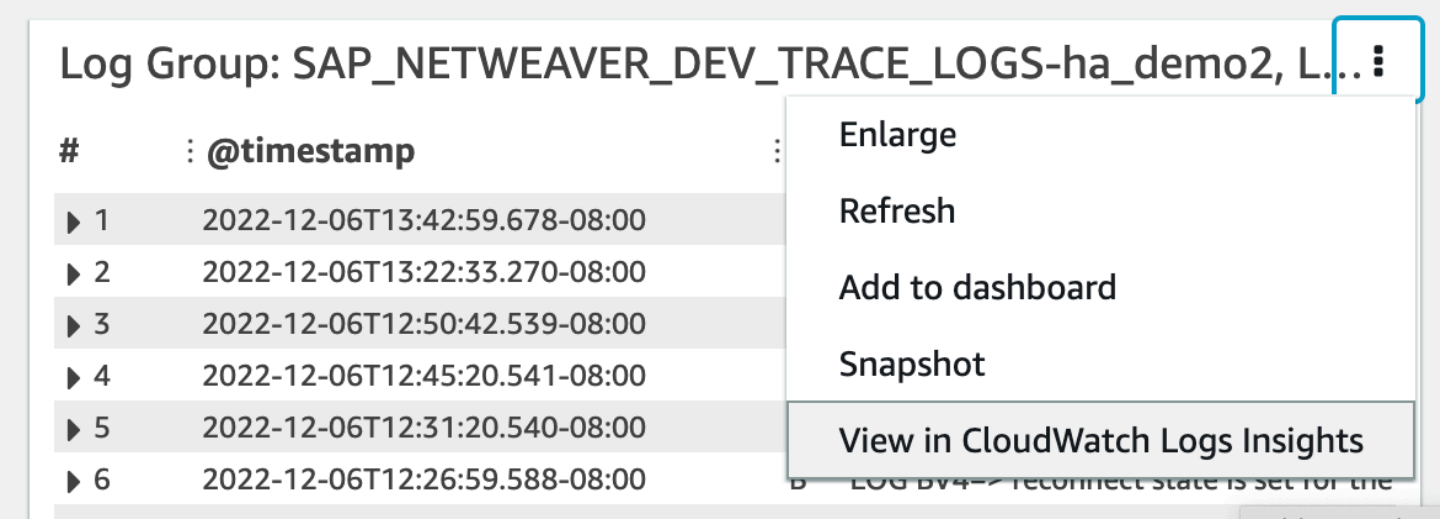


Journaux

Les entrées du journal sont utiles pour mieux comprendre les problèmes survenus au niveau de la NetWeaver couche SAP lorsque le problème a été détecté. Le widget du groupe de journaux dans le tableau de bord des problèmes indique le moment précis du problème.



Pour afficher des informations détaillées sur les journaux, sélectionnez les trois points verticaux dans le coin supérieur droit, puis sélectionnez Afficher dans CloudWatch Logs Insights.

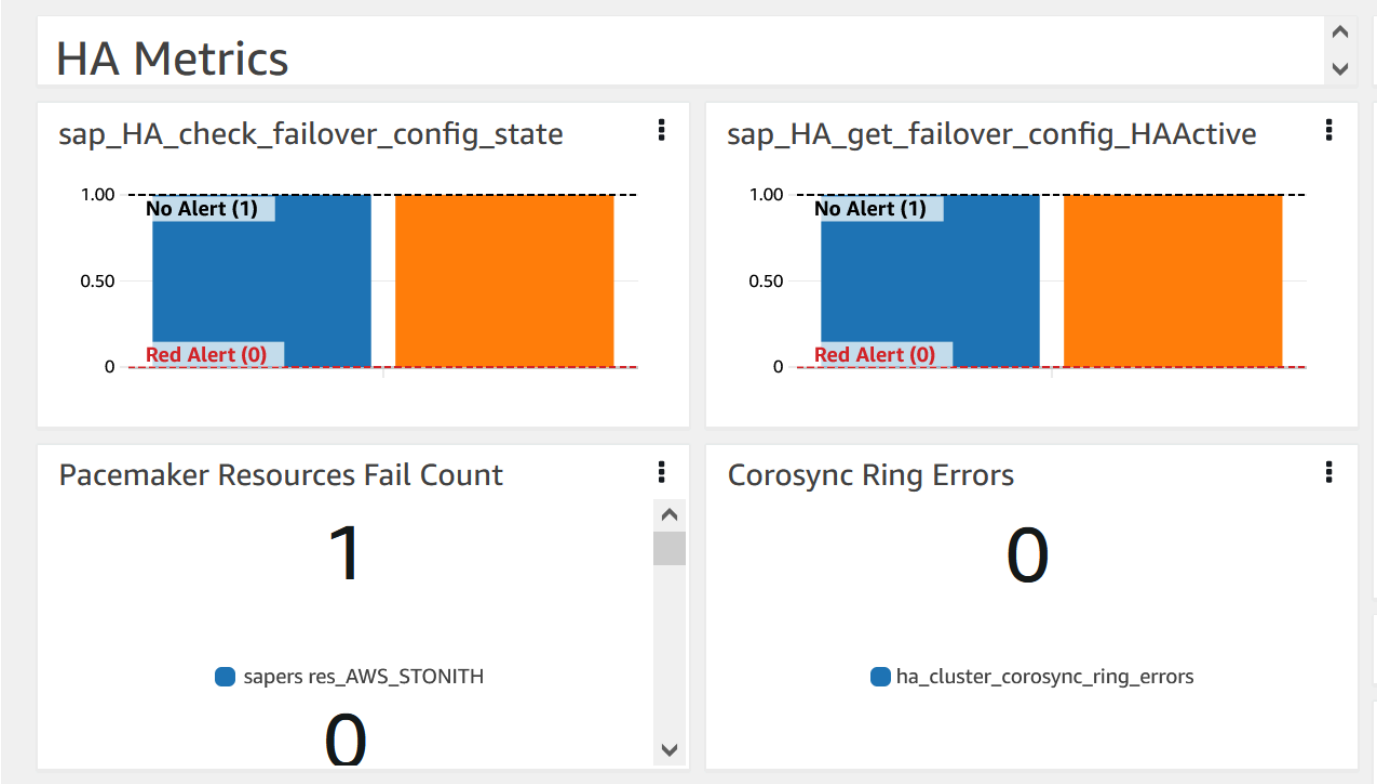


Suivez les étapes suivantes pour obtenir plus d'informations sur les métriques et les alarmes affichées dans le tableau de bord du problème.

Pour obtenir plus d'informations sur les métriques et les alarmes

1. Ouvrez la CloudWatch console.
2. Dans le panneau de navigation de gauche, sélectionnez Application Insights sous Insights. Ensuite, choisissez l'onglet List view (Vue en liste), puis sélectionnez votre application.
3. Sélectionnez l'onglet Components (Composants). Sélectionnez ensuite le NetWeaver composant SAP sur lequel vous souhaitez obtenir plus d'informations.

L'exemple suivant montre la section HA Metrics (Métriques HA) avec la métrique `ha_cluster_pacemaker_fail_count` qui a été affichée dans le tableau de bord du problème.



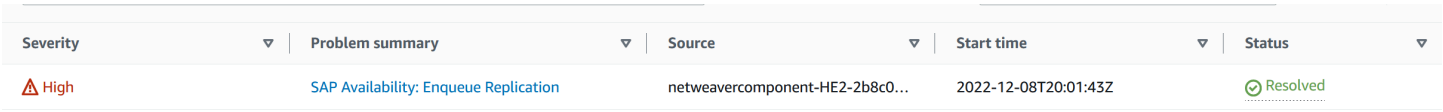
Résolution

Application Insights surveille le problème détecté toutes les heures. S'il n'y a aucune nouvelle entrée de journal associée dans vos fichiers NetWeaver journaux SAP, les anciennes entrées de journal seront considérées comme résolues. Vous devez corriger toutes les conditions d'erreur liées à ce problème.

Pour l'`sap_alerts_Shortdumps`alarme, vous devez résoudre l'alerte dans le NetWeaver système SAP en utilisant le code de transaction `RZ20 # R3Abap # Shortdumps` pour accéder à l'alerte CCMS. Pour plus d'informations sur les alertes CCMS, consultez le site Web de SAP (français non garanti). Résoudre toutes les alertes CCMS dans l'arborescence Shortdumps. Une fois que toutes les alertes ont été résolues dans le NetWeaver système SAP, la métrique CloudWatch n'est plus signalée en état d'alarme.

Lorsque toutes les erreurs du CloudWatch journal et les alarmes sont résolues, Application Insights cesse de détecter les erreurs et le problème est automatiquement résolu en moins d'une heure. Nous

vous recommandons de résoudre toutes les conditions d'erreurs de journal et les alarmes afin de disposer des derniers problèmes sur le tableau de bord des problèmes. Dans l'exemple suivant, le problème de SAP Netweaver High Availability Enqueue Replication est résolu.



Résolution des problèmes liés aux applications pour SAP NetWeaver

Cette section fournit des étapes pour vous aider à résoudre les erreurs courantes renvoyées par le tableau de bord Application Insights.

Impossible d'ajouter plus de 60 métriques de moniteur

Erreur renvoyée :`Component cannot have more than 60 monitored metrics.`

Cause profonde :`The current metric limit is 60 monitor metrics per component.`

Résolution : supprimez les métriques qui ne sont pas nécessaires pour respecter la limite.

Les métriques de SAP n'apparaissent pas sur le tableau de bord après le processus d'intégration

Cause principale : le tableau de bord des composants utilise une période métrique de cinq minutes pour agréger les points de données.

Résolution : toutes les métriques devraient apparaître sur le tableau de bord au bout de cinq minutes.

Les métriques et les alarmes de SAP n'apparaissent pas sur le tableau de bord

Suivez les étapes suivantes pour identifier pourquoi les métriques et les alarmes de SAP n'apparaissent pas sur le tableau de bord après le processus d'intégration.

Pour identifier le problème des métriques et des alarmes

1. Ouvrez la CloudWatch console.
2. Dans le panneau de navigation de gauche, sélectionnez Application Insights sous Insights. Ensuite, choisissez l'onglet List view (Vue en liste), puis sélectionnez votre application.
3. Choisissez l'onglet Configuration history (Historique de configuration).
4. Si vous voyez des points de données de métriques manquants, vérifiez les erreurs liées à `prometheus-sap_host_exporter`.

5. Si vous ne trouvez pas d'erreur à l'étape précédente, connectez-vous à votre instance Linux. Pour les déploiements en haute disponibilité, connectez-vous à l'instance Amazon EC2 du cluster principal.
6. Dans votre instance, vérifiez que l'exportateur est en cours d'exécution à l'aide de la commande suivante. La valeur par défaut du port est `9680`. Si vous utilisez un autre port, remplacez `9680` par le port que vous utilisez.

```
curl localhost:9680/metrics
```

   Si aucune donnée n'est renvoyée, l'exportateur n'a pas pu démarrer.
7. Pour trouver la convention de dénomination correcte à utiliser `WORKLOAD_SHORT_NAME` dans les deux étapes suivantes, exécutez la commande suivante.

> **Note**
>
> Application Insights ajoute un suffixe`WORKLOAD_SHORT_NAME`, au nom du service en fonction de la charge de travail en cours d'exécution. Les noms abrégés des déploiements NetWeaver distribués, standard et haute disponibilité sont `SAP_NWDSAP_NWS`, et`SAP_NWH`.

```
sudo systemctl | grep exporter
```

8. Pour vérifier l'absence d'erreurs dans les journaux de service de l'exportateur, exécutez la commande suivante :

```
sudo journalctl -e --unit=prometheus-sap_host_exporter_WORKLOAD_SHORT_NAME.service
```

9. Pour vérifier l'absence d'erreurs dans les journaux de service du gestionnaire de l'exportateur, exécutez la commande suivante :

```
sudo journalctl -e --unit=prometheus-
sap_host_exporter_manager_WORKLOAD_SHORT_NAME.service
```

> **Note**
>
> Ce service doit être opérationnel à tout moment.

Si cette commande ne renvoie aucune erreur, passez à l'étape suivante.

10. Pour démarrer manuellement l'exportateur, exécutez la commande suivante. Vérifiez ensuite la sortie de l'exportateur.

```
sudo /opt/aws/sap_host_exporter/sap_host_exporter
```

Vous pouvez quitter le processus d'exportation après avoir vérifié l'absence d'erreurs.

Cause racine : plusieurs causes possibles peuvent entraîner ce problème. Une cause commune est que l'exportateur ne peut pas se connecter à l'une des instances du serveur d'application.

Résolution

Suivez les étapes suivantes pour connecter l'exportateur aux instances du serveur d'application. Vous vérifierez que l'instance d'application SAP est en cours d'exécution et utiliserez SAPControl pour vous connecter à l'instance.

Pour connecter l'exportateur aux instances de serveur d'application

1. Dans votre instance Amazon EC2, exécutez la commande suivante pour vérifier que l'application SAP est en cours d'exécution.

```
sapcontrol -nr <App_InstNo> -function GetProcessList
```

2. Vous devez établir une connexion SAPControl fonctionnelle. Si la connexion SAPControl ne fonctionne pas, recherchez la cause racine du problème sur l'instance d'application SAP concernée.
3. Pour démarrer manuellement l'exportateur après avoir résolu le problème de connexion SAPControl, exécutez la commande suivante :

```
sudo systemctl start prometheus-sap_host_exporter.service
```

4. Si vous ne parvenez pas à résoudre le problème de connexion SAPControl, utilisez la procédure suivante comme solution temporaire.

   a. Ouvrez la AWS Systems Manager console.

   b. Dans le volet de navigation de gauche, choisissez State Manager (Gestionnaire d'états).

   c. Dans Associations, recherchez l'association du NetWeaver système SAP.

   ```
   Association Name: Equal: AWS-ApplicationInsights-
   SSMSAPHostExporterAssociationForCUSTOMSAPNW<SID>-1
   ```

   d. Sélectionnez le Association id (Identifiant de l'association).

   e. Cliquez sur l'onglet Parameters (Paramètres) et supprimez le numéro du serveur d'applications dans additionalArguments.

   f. Choisissez Apply Association Now (Appliquer l'association maintenant).

   > **Note**
   >
   > Il s'agit d'une solution temporaire. Si des mises à jour sont apportées aux configurations de surveillance du composant, l'instance sera de nouveau ajoutée.

Afficher et résoudre les problèmes détectés par Amazon CloudWatch Application Insights

Les rubriques de cette section donnent des informations détaillées sur les problèmes détectés et les informations affichées par Application Insights. Elle présente également des solutions suggérées pour les problèmes détectés concernant votre compte ou votre configuration.

Résolution des problèmes liés aux rubriques

- CloudWatch présentation de la console
- Page récapitulative des problèmes d'Application Insights
- CloudWatch échecs liés à un conflit de fusion d'agents
- Les alertes ne sont pas créées
- Commentaires
- Erreurs de configuration

CloudWatch présentation de la console

Vous trouverez un aperçu des problèmes qui ont un impact sur vos applications surveillées dans le volet CloudWatch Application Insights de la page d'aperçu de la CloudWatch console. Pour plus d'informations, consultez Commencez avec Amazon CloudWatch Application Insights.

Le volet de présentation d' CloudWatch Application Insights affiche les informations suivantes :

- La gravité des problèmes détectés : haute/moyenne/basse
- Un bref résumé du problème
- La source du problème
- L'heure à laquelle le problème a commencé
- L'état de résolution du problème
- Le Resource Group affecté

Pour consulter les détails d'un problème spécifique, sous Problem Summary (Résumé du problème), sélectionnez la description du problème. Un tableau de bord détaillé affiche les analyses du problème, les anomalies de métriques connexes et des extraits d'erreurs de journal. Vous pouvez fournir des commentaires sur la pertinence de l'analyse en indiquant si elle est utile ou non.

Si une nouvelle ressource non configurée est détectée, le résumé du problème vous permet d'accéder à l'assistant Edit configuration (Modifier la configuration) pour configurer votre nouvelle ressource. Vous pouvez afficher ou modifier la configuration de votre Resource Group en sélectionnant View/edit configuration (Afficher/modifier la configuration) dans le coin supérieur droit du tableau de bord détaillé.

Pour revenir à l'aperçu, choisissez Retour à l'aperçu, qui se trouve à côté de l'en-tête détaillé du tableau de bord CloudWatch Application Insights.

Page récapitulative des problèmes d'Application Insights

Page récapitulative des problèmes d'Application Insights

CloudWatch Application Insights fournit les informations suivantes sur les problèmes détectés sur la page récapitulative des problèmes :

- Un bref résumé du problème
- La date et l'heure de début du problème

- La gravité du problème : High/Medium/Low (Haute/Moyenne/Basse)
- Le statut du problème détecté : In-progress/Resolved (En cours/Résolu)
- Analyses : génération automatique d’analyses concernant le problème détecté et sa possible cause
- Commentaires sur les informations : commentaires que vous avez fournis sur l'utilité des informations générées par CloudWatch Application Insights
- Observations connexes : une vue détaillée des anomalies métriques et des extraits pertinents de journaux d’erreurs liés au problème, parmi différents composants d’application

CloudWatch échecs liés à un conflit de fusion d'agents

CloudWatch Application Insights installe et configure l' CloudWatch agent sur les instances du client. Cela inclut la création d'un fichier de configuration d' CloudWatch agent avec des configurations pour les métriques ou les journaux. Un conflit de fusion peut survenir si l'instance d'un client possède déjà un fichier de configuration d' CloudWatch agent avec différentes configurations définies pour les mêmes métriques ou journaux. Pour résoudre le conflit de fusion, procédez comme suit :

1. Identifiez les fichiers de configuration de l' CloudWatch agent sur votre système. Pour plus d'informations sur les emplacements de fichiers, consultez CloudWatch fichiers et emplacements des agents.
2. Supprimez les configurations de ressources que vous souhaitez utiliser dans Application Insights du fichier de configuration de l' CloudWatch agent existant. Si vous souhaitez utiliser uniquement les configurations Application Insights, supprimez les fichiers de configuration de CloudWatch l'agent existants.

Les alertes ne sont pas créées

Pour certaines métriques, Application Insights prédit le seuil d'alerte en fonction des points de données précédents pour la métrique. Pour activer cette prédiction, les critères suivants doivent être remplis.

- Points de données récents – Il doit y avoir au moins 100 points de données datant des dernières 24 heures. Les points de données n'ont pas besoin d'être continus et peuvent être répartis sur une période de 24 heures.

- Données historiques – Il doit y avoir au moins 100 points de données couvrant la période comprise entre 15 jours avant la date actuelle et 1 jour avant la date actuelle. Les points de données n'ont pas besoin d'être continus et peuvent être répartis sur une période de 15 jours.

> **Note**
>
> Pour certaines métriques, Application Insights retarde la création d'alertes jusqu'à ce que les conditions précédentes soient remplies. Dans ce cas, vous obtenez un événement d'historique de configuration indiquant que la métrique ne dispose pas de suffisamment de points de données pour établir le seuil d'alerte.

Commentaires

Commentaires

Vous pouvez fournir des commentaires sur les analyses générées automatiquement afférentes aux problèmes détectés en les qualifiant d'utiles ou d'inutiles. Les commentaires que vous laissez sur les analyses, ainsi que vos diagnostics d'application (anomalies métriques et exceptions de journaux) sont utilisés pour améliorer les futures détections de problèmes similaires.

Erreurs de configuration

CloudWatch Application Insights utilise votre configuration pour créer des télémétries de surveillance pour les composants. Quand Application Insights détecte un problème avec votre compte ou votre configuration, des informations sur la façon de résoudre le problème de configuration de votre application sont fournies dans le champ Remarks (Remarques) du résumé de l'application.

Le tableau suivant montre les solutions suggérées pour les remarques spécifiques.

Remarques	Solutions suggérées	Informations complémentaires
Le quota pour CloudFormation a déjà été atteint.	Application Insights crée une CloudFormation pile pour chaque application afin de gérer l'installation et la configuration des CloudWatch agents pour tous les	N/A

Remarques	Solutions suggérées	Informations complémentaires
	composants de l'application. Par défaut, chaque AWS compte peut avoir 2 000 piles. Consultez la section Limites AWS CloudFormation. Pour résoudre ce problème, augmentez la limite des CloudFormation piles.	
Aucun rôle d'instance SSM sur les instances suivantes.	Pour qu'Application Insights puisse installer et configure r l' CloudWatch agent sur les instances d'application, AmazonSSM ManagedIn stanceCore et CloudWatc hAgentServerPolicy les politiques doivent être associés au rôle d'instance.	Application Insights appelle l'DescribeInstanceInformation API SSM pour obtenir la liste des instances disposant d'une autorisation SSM. Une fois le rôle attaché à l'instance, SSM met du temps à inclure l'instance dans le DescribeInstanceIn formation résultat. Jusqu'à ce que le SSM inclue l'instanc e dans le résultat, l'erreur NO_SSM_INSTANCE_ROLE reste présente pour l'applica tion.
De nouveaux composants peuvent avoir besoin d'être configurés.	Application Insights détecte qu'il existe de nouveaux composants dans le Resource Group de l'application. Pour résoudre ce problème, configurez les nouveaux composants en conséquence.	N/A

Logs et statistiques pris en charge par Amazon CloudWatch Application Insights

Les listes suivantes indiquent les journaux et les métriques pris en charge pour Amazon CloudWatch Application Insights.

CloudWatch Application Insights prend en charge les journaux suivants :

- Journaux Microsoft Internet Information Services (IIS)
- Journaux d'erreurs pour SQL Server sur EC2
- Journaux d'applications .NET personnalisés, comme Log4Net
- Journaux d'événements Windows, y compris les journaux Windows (système, application et sécurité) et le journal des applications et services
- Amazon CloudWatch Logs pour AWS Lambda
- Journal des erreurs et journal lent pour RDS MySQL Aurora MySQL et MySQL sur EC2
- Journal PostgreSQL pour PostgreSQL RDS et PostgreSQL sur EC2
- Amazon CloudWatch Logs pour AWS Step Functions
- Journaux d'exécution et journaux d'accès (JSON, CSV et XML, mais pas CLF) pour les étapes d'API REST API Gateway
- Journaux Prometheus JMX Exporter (EMF)
- Journaux d'alerte et journaux d'écoute pour Oracle sur Amazon RDS et Oracle sur Amazon EC2
- Le conteneur enregistre le routage depuis les conteneurs Amazon ECS vers le pilote de journal à CloudWatch l'aide du pilote de `awslogs` journal.
- Le conteneur enregistre le routage depuis les conteneurs Amazon ECS vers CloudWatch un routeur de journaux de FireLens conteneurs.
- Routage des journaux de conteneurs depuis Amazon EKS ou Kubernetes s'exécutant sur Amazon EC2 vers le processeur de journaux Fluent Bit ou Fluentd CloudWatch avec Container Insights.
- Journaux de suivi et d'erreurs SAP HANA
- Journaux de Pacemaker HA
- Journaux du serveur SAP ASE
- Journaux du serveur de sauvegarde SAP ASE
- Journaux du serveur de réplication SAP ASE

- Journaux de l'agent RMA SAP ASE
- Journaux du SAP ASE Fault Manager
- Journaux de suivi des NetWeaver développeurs SAP
- Métriques de processus pour les processus Windows utilisant le plugin proctstat pour agent CloudWatch
- Journaux de requêtes DNS publics pour la zone hébergée
- Amazon Route 53 Resolver Journaux de requêtes DNS

CloudWatch Application Insights prend en charge les classes de journaux suivantes :

- Standard — Amazon CloudWatch Application Insights exige que les groupes de journaux soient configurés avec la classe de CloudWatch journaux Logs Standard pour permettre la surveillance.

CloudWatch Application Insights prend en charge les métriques pour les composants d'application suivants :

- Amazon Elastic Compute Cloud (EC2)
  - CloudWatch métriques intégrées
  - CloudWatch métriques relatives aux agents (serveur Windows)
  - CloudWatch métriques du processus de l'agent (serveur Windows)
  - CloudWatch métriques de l'agent (serveur Linux)
- Elastic Block Store (EBS)
- Amazon Elastic File System (Amazon EFS)
- Elastic Load Balancer (ELB)
- Application ELB
- Groupes Amazon EC2 Auto Scaling
- Amazon Simple Queue Server (SQS)
- Amazon Relational Database Service (RDS)
  - Instances base de données RDS
  - Clusters base de données RDS
- AWS Lambda fonction
- Table Amazon DynamoDB

- Compartiment Amazon S3
- AWS Step Functions
  - Niveau exécution
  - Activité
  - Fonction Lambda
  - Intégration de service
  - Step Functions API
- Étapes d'API REST API Gateway d'API
- SAP HANA
- SAP ASE
- Haute disponibilité SAP ASE sur Amazon EC2
- SAP NetWeaver
- Cluster HA
- Java
- Amazon Elastic Container Service (Amazon ECS)
  - CloudWatch métriques intégrées
  - Métriques Container Insights
  - Métriques Prometheus Container Insights
- Kubernetes activé AWS
  - Métriques Container Insights
  - Métriques Prometheus Container Insights
- Amazon FSx
- Amazon VPC
- Passerelles NAT Amazon VPC
- Surveillance de l'état Amazon Route 53
- Zone hébergée Amazon Route 53
- Amazon Route 53 Resolver point de terminaison
- AWS Network Firewall groupe de règles
- AWS Network Firewall association de groupes de règles

- Métriques avec exigences aux points de données
  - AWS/ApplicationELB
  - AW/ AutoScaling
  - AWS/EC2
  - Elastic Block Store (EBS)
  - AWS/ELB
  - AWS/RDS
  - AWS/Lambda
  - AWS/SQS
  - AWS/CWAgent
  - AWS/DynamoDB
  - AWS/S3
  - AWS/States
  - AW/ ApiGateway
  - AWS/SNS
- Métriques recommandées
- Métriques du compteur de performances

Amazon Elastic Compute Cloud (EC2)

CloudWatch Application Insights prend en charge les mesures suivantes :

Métriques

- CloudWatch métriques intégrées
- CloudWatch métriques relatives aux agents (serveur Windows)
- CloudWatch métriques du processus de l'agent (serveur Windows)
- CloudWatch métriques de l'agent (serveur Linux)

CloudWatch métriques intégrées

CPU CreditBalance

CPU CreditUsage

CPU SurplusCreditBalance

CPU SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

% EBS ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_Système

CloudWatch métriques relatives aux agents (serveur Windows)

Exceptions CLR .NET Nombre d'exceptions émises

Exceptions CLR .NET Nombre d'exceptions émises/s

Exceptions CLR .NET Nombre de filtres/s

Exceptions CLR .NET Nombre de finalements/s

Exceptions CLR .NET Lancer pour capter la profondeur/s

Interop CLR .NET Nombre de CCW

Interop CLR .NET Nombre de talons

Interop CLR .NET Nombre d'exportations TLB/s

Interop CLR .NET Nombre d'importations TLB/s

Interop CLR .NET Nombre de regroupements

Jit CLR .NET % temps en juste-à-temps

Jit CLR .NET Échecs juste-à-temps standard

Chargement CLR .NET % temps de chargement

Chargement CLR .NET Taux des échecs de chargement

Taux de LocksAndThreads contention du CLR .NET par seconde

Longueur de la file d' LocksAndThreads attente .NET CLR/sec

Mémoire CLR .NET # Nombre total d'octets engagés

Mémoire CLR .NET % de temps en GC

.NET CLR Networking 4.0.0.0 HttpWebRequest Temps d'attente moyen

Mise en réseau .NET CLR HttpWebRequests 4.0.0.0 abandonnée par seconde

Le réseau .NET CLR HttpWebRequests 4.0.0.0 a échoué/sec

Réseau .NET CLR HttpWebRequests 4.0.0.0 en file d'attente/sec

APP_POOL_WAS Total des échecs de ping de processus de travail

L'application ASP.NET redémarre

Applications ASP.NET Temps de traitement en % du processeur géré (estimé)

Applications ASP.NET Total erreurs/s

Applications ASP.NET Erreurs non traitées pendant l'exécution/s

Applications ASP.NET Demandes en file d'attente d'application

Applications ASP.NET Demandes/s

Temps d'attente de la demande ASP.NET

Demandes ASP.NET en file d'attente

Files d'attente de service HTTP CurrentQueueSize

LogicalDisk % d'espace libre

% d'octets validés de mémoire en cours d'utilisation

Mo de mémoire disponible

Pages mémoire/s

Total octets interface réseau/sec

Utilisation en % du fichier de pagination

PhysicalDisk % de temps sur le disque

PhysicalDisk Moyenne. Longueur de la file d'attente de disque

PhysicalDisk Moyenne. Disk sec/Read (Lecture/s sur disque)

PhysicalDisk Moyenne. Disk Write/sec (Écriture/s sur disque)

PhysicalDisk Octets lus sur le disque par seconde

PhysicalDisk Lectures sur disque par seconde

PhysicalDisk Octets d'écriture sur disque par seconde

PhysicalDisk Nombre d'écritures sur disque par seconde

Temps d'inactivité en % du processeur

Durée d'interruption en % du processeur

Temps de traitement en % du processeur

Temps utilisateur en % du processeur

SQLServer : Méthodes d'accès - Enregistrements transmis/sec

SQLServer : Méthodes d'accès - Analyses complètes/s

SQLServer : Pages de méthodes - Splits de pages/sec

SQLServer : Taux de réussite du cache du gestionnaire de tampon

SQLServer : Espérance de vie de la page du gestionnaire de tampon

SQLServer : Statistiques générales - Processus bloqués

SQLServer : Statistiques générales - Connexions utilisateurs

SQLServer : Moyenne loquets - Temps d'attente du loquet (ms)

SQLServer : Verrouillages - Temps d'attente moyen (ms)

SQLServer : Verrouillages - Verrouillage des temporisations/s

SQLServer : Verrouillages - Attente de verrouillage/s

SQLServer : Verrouillages - Nombre de blocages/s

SQLServer : Gestionnaire de mémoire - Octrois de mémoire en attente

SQLServer : Statistiques SQL - Requêtes par lots/sec

SQLServer : Statistiques SQL - Compilations SQL/sec

SQLServer : Statistiques SQL - Recompilations SQL/sec

Longueur de la file d'attente du processeur système

Connexions TCPv4 établies

Connexions TCPv6 établies

W3SVC_W3WP Vidages du cache de fichiers

W3SVC_W3WP Échecs du cache de fichiers

W3SVC_W3WP Demandes/s

W3SVC_W3WP Vidages de cache d'URI

W3SVC_W3WP Échecs de cache d'URI

Service Web Octets reçus/s

Service Web Octets envoyés/s

Tentatives/s de connexion au service Web

Service Web Connexions actuelles

Service Web Obtenir des requêtes/s

Service Web Demandes de publication/s

Octets reçus/s

Longueur de la file d'attente des messages normaux/s

Longueur de la file d'attente de message urgent/s

Nombre de reconnexion

Longueur de la file d'attente de message non reconnu/s

Messages en attente

Messages envoyés/s

Messages de mise à jour de la base de données/s

Messages de mise à jour/sec

Vidanges/s

Points de contrôle crypto enregistrés/s

Points de contrôle crypto restaurés/s

Points de contrôle du registre restaurés/s

Points de contrôle du registre enregistrés/s

Appels d'API de cluster/s

Appels d'API de ressource/s

Descripteurs de cluster/s

Descripteurs de ressource/s

CloudWatch métriques du processus de l'agent (serveur Windows)

Les métriques de processus sont collectées à l'aide du plugin CloudWatch agent procstat. Seules les instances Amazon EC2 exécutant des charges de travail Windows prennent en charge les métriques de processus.

procstat cpu_time_system

procstat cpu_time_user

procstat cpu_usage

procstat memory_rss

procstat memory_vms

procstat read_bytes

procstat write_bytes

.procstat read_count

procstat write_count

CloudWatch métriques de l'agent (serveur Linux)

cpu_time_active

cpu_time_guest

cpu_time_guest_nice

cpu_time_idle

cpu_time_iowait

cpu_time_irq

cpu_time_nice

cpu_time_softirq

cpu_time_steal

cpu_time_system

cpu_time_user

cpu_usage_active

cpu_usage_guest

cpu_usage_guest_nice

cpu_usage_idle

cpu_usage_iowait

cpu_usage_irq

cpu_usage_nice

cpu_usage_softirq

cpu_usage_steal

cpu_usage_system

cpu_usage_user

disk_free

disk_inodes_free

disk_inodes_used

disk_used

disk_used_percent

diskio_io_time

diskio_iops_in_progress

diskio_read_bytes

diskio_read_time

diskio_reads

diskio_write_bytes

diskio_write_time

diskio_writes

mem_active

mem_available

mem_available_percent

mem_buffered

mem_cached

mem_free

mem_inactive

mem_used

mem_used_percent

net_bytes_recv

net_bytes_sent

net_drop_in

net_drop_out

net_err_in

net_err_out

net_packets_recv

net_packets_sent

netstat_tcp_close

netstat_tcp_close_wait

netstat_tcp_closing

netstat_tcp_established

netstat_tcp_fin_wait1

netstat_tcp_fin_wait2

netstat_tcp_last_ack

netstat_tcp_listen

netstat_tcp_none

netstat_tcp_syn_recv

netstat_tcp_syn_sent

netstat_tcp_time_wait

netstat_udp_socket

processes_blocked

processes_dead

processes_idle

processes_paging

processes_running

processes_sleeping

processes_stopped

processes_total

processes_total_threads

processes_wait

processes_zombies

swap_free

swap_used

swap_used_percent

Elastic Block Store (EBS)

CloudWatch Application Insights prend en charge les mesures suivantes :

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

Amazon Elastic File System (Amazon EFS)

CloudWatch Application Insights prend en charge les mesures suivantes :

BurstCreditBalance

PercentIOLimit

PermittedThroughput

MeteredIOBytes

TotalIOBytes

DataWriteIOoctets

DataReadIOoctets

MetadataIOBytes

ClientConnections

TimeSinceLastSync

StorageBytes

Débit

PercentageOfPermittedThroughputUtilization

ThroughputIOPS

PercentThroughputDataReadIOoctet

PercentThroughputDataWriteIOoctets

PercentageOfIOPS DataRead en octets

PercentageOfIOPS DataWrite en octets

AverageDataReadIO BytesSize

AverageDataWriteIO BytesSize

Elastic Load Balancer (ELB)

CloudWatch Application Insights prend en charge les mesures suivantes :

Dalb estimé ActiveConnectionCount

EstimatedALBConsumedLCUs

Dalb estimé NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

RequestCount

UnHealthyHostCount

Application ELB

CloudWatch Application Insights prend en charge les mesures suivantes :

Dalb estimé ActiveConnectionCount

EstimatedALBConsumedLCUs

Dalb estimé NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latence

RequestCount

SurgeQueueLength

UnHealthyHostCount

Groupes Amazon EC2 Auto Scaling

CloudWatch Application Insights prend en charge les mesures suivantes :

CPU CreditBalance

CPU CreditUsage

CPU SurplusCreditBalance

CPU SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

% EBS ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

StatusCheckFailed

StatusCheckFailed_Instance

StatusCheckFailed_Système

Amazon Simple Queue Server (SQS)

CloudWatch Application Insights prend en charge les mesures suivantes :

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

Amazon Relational Database Service (RDS)

CloudWatch Application Insights prend en charge les mesures suivantes :

Métriques

- Instances base de données RDS
- Clusters base de données RDS

Instances base de données RDS

BurstBalance

CPU CreditBalance

CPUUtilization

DatabaseConnections

DiskQueueDepth

SQL défaillant ServerAgentJobsCount

FreeStorageSpace

FreeableMemory

NetworkReceiveThroughput

NetworkTransmitThroughput

ReadIOPS

ReadLatency

ReadThroughput

WriteIOPS

WriteLatency

WriteThroughput

Clusters base de données RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

BufferCacheHitRatio

CPUUtilization

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlocks

DeleteLatency

DeleteThroughput

EngineUptime

FreeLocalStorage

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Requêtes

ResultSetCacheHitRatio

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPS

VolumeWriteIOPS

AWS Lambda fonction

CloudWatch Application Insights prend en charge les mesures suivantes :

Erreurs

DeadLetterErrors

Durée

Throttles

IteratorAge

ProvisionedConcurrencySpilloverInvocations

Table Amazon DynamoDB

CloudWatch Application Insights prend en charge les mesures suivantes :

SystemErrors

UserErrors

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

Compartiment Amazon S3

CloudWatch Application Insights prend en charge les mesures suivantes :

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS Step Functions

CloudWatch Application Insights prend en charge les mesures suivantes :

Métriques

- Niveau exécution
- Activité
- Fonction Lambda
- Intégration de service
- Step Functions API

Niveau exécution

ExecutionTime

ExecutionThrottled

ExecutionsFailed

ExecutionsTimedOut

ExecutionsAborted

ExecutionsSucceeded

ExecutionsStarted

Activité

ActivityRunTime

ActivityScheduleTime

ActivityTime

ActivitiesFailed

ActivitiesHeartbeatTimedOut

ActivitiesTimedOut

ActivitiesScheduled

ActivitiesSucceeded

ActivitiesStarted

Fonction Lambda

LambdaFunctionRunTime

LambdaFunctionScheduleTime

LambdaFunctionTime

LambdaFunctionsFailed

LambdaFunctionsTimedOut

LambdaFunctionsScheduled

LambdaFunctionsSucceeded

LambdaFunctionsStarted

Intégration de service

ServiceIntegrationRunTime

ServiceIntegrationScheduleTime

ServiceIntegrationTime

ServiceIntegrationsFailed

ServiceIntegrationsTimedOut

ServiceIntegrationsScheduled

ServiceIntegrationsSucceeded

ServiceIntegrationsStarted

Step Functions API

ThrottledEvents

ProvisionedBucketSize

ProvisionedRefillRate

ConsumedCapacity

Étapes d'API REST API Gateway d'API

CloudWatch Application Insights prend en charge les mesures suivantes :

4XXError

5XXError

IntegrationLatency

Latence

CacheHitCount

CacheMissCount

SAP HANA

> **Note**
>
> CloudWatch Application Insights ne prend en charge que les environnements SID HANA uniques. Si plusieurs SID HANA sont connectés, la surveillance sera configurée uniquement pour le premier SID détecté.

CloudWatch Application Insights prend en charge les mesures suivantes :

hanadb_every_service_started_status

hanadb_daemon_service_started_status

hanadb_preprocessor_service_started_status

hanadb_webdispatcher_service_started_status

hanadb_compileserver_service_started_status

hanadb_nameserver_service_started_status

hanadb_server_startup_time_variations_seconds

hanadb_level_5_alerts_count

hanadb_level_4_alerts_count

hanadb_out_of_memory_events_count

hanadb_max_trigger_read_ratio_percent

hanadb_max_trigger_write_ratio_percent

hanadb_log_switch_wait_ratio_percent

hanadb_log_switch_race_ratio_percent

hanadb_time_since_last_savepoint_seconds

hanadb_disk_usage_highlevel_percent

hanadb_max_converter_page_number_count

hanadb_long_running_savepoints_count

hanadb_failed_io_reads_count

hanadb_failed_io_writes_count

hanadb_disk_data_unused_percent

hanadb_current_allocation_limit_used_percent

hanadb_table_allocation_limit_used_percent

hanadb_host_total_physical_memory_mb

hanadb_host_physical_memory_used_mb

hanadb_host_physical_memory_free_mb

hanadb_swap_memory_free_mb

hanadb_swap_memory_used_mb

hanadb_host_allocation_limit_mb

hanadb_host_total_memory_used_mb

hanadb_host_total_peak_memory_used_mb

hanadb_host_total_allocation_limit_mb

hanadb_host_code_size_mb

hanadb_host_shared_memory_allocation_mb

hanadb_cpu_usage_percent

hanadb_cpu_user_percent

hanadb_cpu_system_percent

hanadb_cpu_waitio_percent

hanadb_cpu_busy_percent

hanadb_cpu_idle_percent

hanadb_long_delta_merge_count

hanadb_unsuccessful_delta_merge_count

hanadb_successful_delta_merge_count

hanadb_row_store_allocated_size_mb

hanadb_row_store_free_size_mb

hanadb_row_store_used_size_mb

hanadb_temporary_tables_count

hanadb_large_non_compressed_tables_count

hanadb_total_non_compressed_tables_count

hanadb_longest_running_job_seconds

hanadb_average_commit_time_milliseconds

hanadb_suspended_sql_statements_count

hanadb_plan_cache_hit_ratio_percent

hanadb_plan_cache_lookup_count

hanadb_plan_cache_hit_count

hanadb_plan_cache_total_execution_microseconds

hanadb_plan_cache_cursor_duration_microseconds

hanadb_plan_cache_preparation_microseconds

hanadb_plan_cache_evicted_count

hanadb_plan_cache_evicted_microseconds

hanadb_plan_cache_evicted_preparation_count

hanadb_plan_cache_evicted_execution_count

hanadb_plan_cache_evicted_preparation_microseconds

hanadb_plan_cache_evicted_cursor_duration_microseconds

hanadb_plan_cache_evicted_total_execution_microseconds

hanadb_plan_cache_evicted_plan_size_mb

hanadb_plan_cache_count

hanadb_plan_cache_preparation_count

hanadb_plan_cache_execution_count

hanadb_network_collision_rate

hanadb_network_receve_rate

hanadb_network_transmit_rate

hanadb_network_packet_receve_rate

hanadb_network_packet_transmit_rate

hanadb_network_transmit_error_rate

hanadb_network_receve_error_rate

hanadb_time_until_license_expires_days

hanadb_is_license_valid_status

hanadb_local_running_connections_count

hanadb_local_idle_connections_count

hanadb_remote_running_connections_count

hanadb_remote_idle_connections_count

hanadb_last_full_data_backup_age_days

hanadb_last_data_backup_age_days

hanadb_last_log_backup_age_hours

hanadb_failed_data_backup_past_7_days_count

hanadb_failed_log_backup_past_7_days_count

hanadb_oldest_backup_in_catalog_age_days

hanadb_backup_catalog_size_mb

hanadb_hsr_replication_status

hanadb_hsr_log_shipping_delay_seconds

hanadb_hsr_secondary_failover_count

hanadb_hsr_secondary_reconnect_count

hanadb_hsr_async_buffer_used_mb

hanadb_hsr_secondary_active_status

hanadb_handle_count

hanadb_ping_time_millisecondes

hanadb_connection_count

hanadb_internal_connection_count

hanadb_external_connection_count

hanadb_idle_connection_count

hanadb_transaction_count

hanadb_internal_transaction_count

hanadb_external_transaction_count

hanadb_user_transaction_count

hanadb_blocked_transaction_count

hanadb_statement_count

hanadb_active_commit_id_range_count

hanadb_mvcc_version_count

hanadb_pending_session_count

hanadb_record_lock_count

hanadb_read_count

hanadb_write_count

hanadb_merge_count

hanadb_unload_count

hanadb_active_thread_count

hanadb_waiting_thread_count

hanadb_total_thread_count

hanadb_active_sql_executor_count

hanadb_waiting_sql_executor_count

hanadb_total_sql_executor_count

hanadb_data_write_size_mb

hanadb_data_write_time_milliseconds

hanadb_log_write_size_mb

hanadb_log_write_time_milliseconds

hanadb_data_read_size_mb

hanadb_data_read_time_milliseconds

hanadb_log_read_size_mb

hanadb_log_read_time_milliseconds

hanadb_data_backup_write_size_mb

hanadb_data_backup_write_time_milliseconds

hanadb_log_backup_write_size_mb

hanadb_log_backup_write_time_milliseconds

hanadb_mutex_collision_count

hanadb_read_write_lock_collision_count

hanadb_admission_control_admit_count

hanadb_admission_control_reject_count

hanadb_admission_control_queue_size_mb

hanadb_admission_control_wait_time_milliseconds

SAP ASE

CloudWatch Application Insights prend en charge les mesures suivantes :

asedb_database_availability

asedb_trunc_log_on_chkpt_enabled

asedb_last_db_backup_age_in_days

asedb_last_transaction_log_backup_age_in_hours

asedb_suspected_database

asedb_db_space_usage_percent

asedb_db_log_space_usage_percent

asedb_locked_login

asedb_has_mixed_log_and_data

asedb_runtime_for_open_transactions

asedb_data_cache_hit_ratio

asedb_data_cache_usage

asedb_sql_cache_hit_ratio

asedb_cache_usage

asedb_run_queue_length

asedb_number_of_rollbacks

asedb_number_of_commits

asedb_number_of_transactions

asedb_outstanding_disk_io

asedb_percent_io_busy

asedb_percent_system_busy

asedb_percent_locks_active

asedb_scheduled_jobs_failed_percent

asedb_user_connections_percent

asedb_query_logical_reads

asedb_query_physical_reads

asedb_query_cpu_time

asedb_query_memory_usage

Haute disponibilité SAP ASE sur Amazon EC2

CloudWatch Application Insights prend en charge les mesures suivantes :

asedb_ha_replication_state

asedb_ha_replication_mode

asedb_ha_replication_latency_in_minutes

SAP NetWeaver

CloudWatch Application Insights prend en charge les mesures suivantes :

Métrique	Description
alertes_sap_ ResponseTime	L'alerte de temps de réponse SAP envoyée par CCMS (RZ20) >R3Services>Dialog>. ResponseTime
alertes_sap_ ResponseTimeDialog	L'alerte de dialogue sur le temps de réponse SAP envoyée par CCMS (RZ20) >R3Servic es>Dialog>. ResponseTimeDialog
SAP_ALERTS_ RFC ResponseTimeDialog	L'alerte de temps de réponse SAP envoyée par CCMS (RZ20) > R3 Services > Dialog > RFC. ResponseTimeDialog
SAP Alerts_DB RequestTime	L'alerte de temps de réponse SAP envoyée par CCMS (RZ20) >R3Services>Dialog>DB. RequestTime
alertes_sap_ FrontendResponseTime	L'alerte de temps de réponse SAP envoyée par CCMS (RZ20) > R3Services > Dialog>. FrontEndResponseTime
sap_alerts_Database	Le système SAP a enregistré des erreurs liées à la base de données. Alerte provenant de SM21 ou de CCMS (RZ20)>R3Syslog>Da tabase.
alertes_sap_ QueueTime	L'alerte de temps d'attente SAP envoyée par CCMS (RZ20) >R3Services>Dialog>. QueueTime

Métrique	Description
alertes_sap_ AbortedJobs	Échec des tâches d'arrière-plan dans le système SAP. Alerte de (RZ20) > Services R3 > Contexte >. AbortedJobs
alertes_sap_ BasisSystem	Le système SAP a journalisé les erreurs de niveau système. Alerte de SM21 ou CCMS (RZ20) >R3Syslog>. BasisSystem
sap_alerts_Security	Le système SAP a journalisé des messages liés à la sécurité. Alerte provenant de SM21 ou CCMS (RZ20)>R3Syslog>Security.
sap_alerts_System	Le système SAP a enregistré des messages relatifs à la sécurité ou à l'audit. Alerte provenant de SM21 ou CCMS (RZ20)>Se curity>System.
alertes_sap_ LongRunners	Des programmes sont en cours d'exécuti on depuis longtemps dans votre système SAP. Alerte du CCMS (RZ20) > R3Services > Dialog>. LongRunners
alertes_sap_ SqlError	Il existe des journaux d'erreurs de la couche client de la base de données SAP. Alerte du CCMS (RZ20) > DatabaseClient >AbapSql. SqlError
sap_alerts_State	Alerte d'état provenant de CCMS (RZ20)>OS Collector>State.
sap_alerts_Shortdumps	Alerte de shortdumps à partir de ST22 et CCMS (RZ20)>R3Abap>Shortdumps.
sap_alerts_Availability	Alerte de disponibilité pour une instance de serveur d'applications SAP émise par SM21, SM50, SM51, SM66 et CCMS (RZ20) > >Disponibilité. InstanceAsTask

Métrique	Description
sap_dispatcher_queue_high	La fonction SAPControl Web Service `GetQueueStatistic` fournit le nombre élevé de files d'attente du répartiteur.
sap_dispatcher_queue_max	La fonction SAPControl Web Service `GetQueueStatistic` fournit le compte maximum de la file d'attente du répartiteur.
sap_dispatcher_queue_now	La fonction SAPControl Web Service `GetQueueStatistic` fournit le nombre de files d'attente du répartiteur maintenant.
sap_dispatcher_queue_reads	La fonction SAPControl Web Service `GetQueueStatistic` fournit le nombre de lectures de la file d'attente du répartiteur.
sap_dispatcher_queue_writes	La fonction SAPControl Web Service `GetQueueStatistic` fournit le nombre d'écritures dans la file d'attente du répartiteur.
sap_enqueue_server_arguments_high	La fonction SAPControl Web Service `EnqGetStatistic` fournit les arguments de mise en file d'attente haut.
sap_enqueue_server_arguments_max	La fonction SAPControl Web Service `EnqGetStatistic` fournit les arguments de mise en file d'attente max.
sap_enqueue_server_arguments_now	La fonction SAPControl Web Service `EnqGetStatistic` fournit les arguments de mise en file d'attente maintenant.
sap_enqueue_server_arguments_state	La fonction SAPControl Web Service `EnqGetStatistic` fournit les arguments de mise en file d'attente état.

Métrique	Description
sap_enqueue_server_backup_requests	La fonction SAPControl Web Service `EnqGetStatistic` fournit les demandes de sauvegarde de mise en file d'attente.
sap_enqueue_server_cleanup_requests	La fonction SAPControl Web Service `EnqGetStatistic` fournit les demandes de nettoyage de mise en file d'attente.
sap_enqueue_server_dequeue_all_requests	La fonction SAPControl Web Service `EnqGetStatistic` fournit les demandes de retrait de la file d'attente de tout.
sap_enqueue_server_dequeue_errors	La fonction SAPControl Web Service `EnqGetStatistic` fournit les erreurs de retrait de la file d'attente.
sap_enqueue_server_dequeue_requests	La fonction SAPControl Web Service `EnqGetStatistic` fournit les demandes de retrait de la file d'attente.
sap_enqueue_server_enqueue_errors	La fonction SAPControl Web Service `EnqGetStatistic` fournit les erreurs de mise en file d'attente.
sap_enqueue_server_enqueue_rejects	La fonction SAPControl Web Service `EnqGetStatistic` fournit les rejets de mise en file d'attente.
sap_enqueue_server_enqueue_requests	La fonction SAPControl Web Service `EnqGetStatistic` fournit les demandes de mise en file d'attente.
sap_enqueue_server_lock_time	La fonction SAPControl Web Service `EnqGetStatistic` fournit le temps de verrouillage de la mise en file d'attente.

Métrique	Description
sap_enqueue_server_lock_wait_time	La fonction SAPControl Web Service `EnqGetStatistic` fournit le temps d'attente du verrouillage de la mise en file d'attente.
sap_enqueue_server_locks_high	La fonction SAPControl Web Service `EnqGetStatistic` fournit les verrous de mise en file d'attente haut.
sap_enqueue_server_locks_max	La fonction SAPControl Web Service `EnqGetStatistic` fournit les verrous de mise en file d'attente max.
sap_enqueue_server_locks_now	La fonction SAPControl Web Service `EnqGetStatistic` fournit les verrous de mise en file d'attente maintenant.
sap_enqueue_server_locks_state	La fonction SAPControl Web Service `EnqGetStatistic` fournit les verrous de mise en file d'attente état.
sap_enqueue_server_owner_high	La fonction SAPControl Web Service `EnqGetStatistic` fournit le propriétaire de mise en file d'attente haut.
sap_enqueue_server_owner_max	La fonction SAPControl Web Service `EnqGetStatistic` fournit le propriétaire de mise en file d'attente max.
sap_enqueue_server_owner_now	La fonction SAPControl Web Service `EnqGetStatistic` fournit le propriétaire de mise en file d'attente maintenant.
sap_enqueue_server_owner_state	La fonction SAPControl Web Service `EnqGetStatistic` fournit le propriétaire de mise en file d'attente état.

Métrique	Description
sap_enqueue_server_replication_state	La fonction SAPControl Web Service `EnqGetStatistic` fournit le statut de l'état de réplication de la mise en file d'attente.
sap_enqueue_server_reporting_requests	La fonction SAPControl Web Service `EnqGetStatistic` fournit le statut des demandes de reqporting.
sap_enqueue_server_server_time	La fonction SAPControl Web Service `EnqGetStatistic` fournit l'heure du serveur de mise en file d'attente.
sap_HA_check_failover_config_state	La fonction SAPControl Web Service `HACheckFailoverConfig` fournit le statut de SAP High Availability.
sap_HA_get_failover_config_HAActive	La fonction SAPControl Web Service `HAGetFailoverConfig` fournit la configuration et le statut du cluster de SAP High Availability.
sap_start_service_processes	La fonction SAPControl Web Service `GetProcessList` fournit le statut des processus disp+work, IGS, gwrd, icman, serveur de messages et serveur de mise en file d'attente.

Cluster HA

CloudWatch Application Insights prend en charge les mesures suivantes :

ha_cluster_pacemaker_stonith_enabled

ha_cluster_corosync_quorate

hanadb_webdispatcher_service_started_status

ha_cluster_pacemaker_nodes

ha_cluster_corosync_ring_errors

ha_cluster_pacemaker_fail_count

Java

CloudWatch Application Insights prend en charge les mesures suivantes :

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_threading_daemonthreadcount

java_lang_classloading_loadedclasscount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon Elastic Container Service (Amazon ECS)

CloudWatch Application Insights prend en charge les mesures suivantes :

Métriques

- CloudWatch métriques intégrées
- Métriques Container Insights
- Métriques Prometheus Container Insights

CloudWatch métriques intégrées

CPUReservation

CPUUtilization

MemoryReservation

MemoryUtilization

GPUReservation

Métriques Container Insights

ContainerInstanceCount

CpuUtilized

CpuReserved

DeploymentCount

DesiredTaskCount

MemoryUtilized

MemoryReserved

NetworkRxBytes

NetworkTxBytes

PendingTaskCount

RunningTaskCount

ServiceCount

StorageReadBytes

StorageWriteBytes

TaskCount

TaskSetCount

instance_cpu_limit

instance_cpu_reserved_capacity

instance_cpu_usage_total

instance_cpu_utilization

instance_filesystem_utilization

instance_memory_limit

instance_memory_reserved_capacity

instance_memory_utilization

instance_memory_working_set

instance_network_total_bytes

instance_number_of_running_tasks

Métriques Prometheus Container Insights

Métriques Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Kubernetes activé AWS

CloudWatch Application Insights prend en charge les mesures suivantes :

Métriques

- Métriques Container Insights
- Métriques Prometheus Container Insights

Métriques Container Insights

cluster_failed_node_count

cluster_node_count

namespace_number_of_running_pods

node_cpu_limit

node_cpu_reserved_capacity

node_cpu_usage_total

node_cpu_utilization

node_filesystem_utilization

node_memory_limit

node_memory_reserved_capacity

node_memory_utilization

node_memory_working_set

node_network_total_bytes

node_number_of_running_containers

node_number_of_running_pods

pod_cpu_reserved_capacity

pod_cpu_utilization

pod_cpu_utilization_over_pod_limit

pod_memory_reserved_capacity

pod_memory_utilization

pod_memory_utilization_over_pod_limit

pod_network_rx_bytes

pod_network_tx_bytes

service_number_of_running_pods

Métriques Prometheus Container Insights

Métriques Java JMX

java_lang_memory_heapmemoryusage_used

java_lang_memory_heapmemoryusage_committed

java_lang_operatingsystem_openfiledescriptorcount

java_lang_operatingsystem_maxfiledescriptorcount

java_lang_operatingsystem_freephysicalmemorysize

java_lang_operatingsystem_freeswapspacesize

java_lang_threading_threadcount

java_lang_classloading_loadedclasscount

java_lang_threading_daemonthreadcount

java_lang_garbagecollector_collectiontime_copy

java_lang_garbagecollector_collectiontime_ps_scavenge

java_lang_garbagecollector_collectiontime_parnew

java_lang_garbagecollector_collectiontime_marksweepcompact

java_lang_garbagecollector_collectiontime_ps_marksweep

java_lang_garbagecollector_collectiontime_concurrentmarksweep

java_lang_garbagecollector_collectiontime_g1_young_generation

java_lang_garbagecollector_collectiontime_g1_old_generation

java_lang_garbagecollector_collectiontime_g1_mixed_generation

java_lang_operatingsystem_committedvirtualmemorysize

Amazon FSx

CloudWatch Application Insights prend en charge les mesures suivantes :

DataReadBytes

DataWriteBytes

DataReadOperations

DataWriteOperations

MetadataOperations

FreeStorageCapacity

FreeDataStorageCapacity

LogicalDiskUsage

PhysicalDiskUsage

Amazon VPC

CloudWatch Application Insights prend en charge les mesures suivantes :

NetworkAddressUsage

NetworkAddressUsagePeered

VPC FirewallQueryVolume

Passerelles NAT Amazon VPC

CloudWatch Application Insights prend en charge les mesures suivantes :

ErrorPortAllocation

IdleTimeoutCount

Surveillance de l'état Amazon Route 53

CloudWatch Application Insights prend en charge les mesures suivantes :

ChildHealthCheckHealthyCount

ConnectionTime

HealthCheckPercentageHealthy

HealthCheckStatus

SSL HandshakeTime

TimeToFirstByte

Zone hébergée Amazon Route 53

CloudWatch Application Insights prend en charge les mesures suivantes :

DNSQueries

DNSSEC InternalFailure

DNSSEC KeySigningKeysNeedingAction

DNSSEC KeySigningKeyMaxNeedingActionAge

DNSSEC KeySigningKeyAge

Amazon Route 53 Resolver point de terminaison

CloudWatch Application Insights prend en charge les mesures suivantes :

EndpointHealthyCompte ENI

EndpointUnHealthyCompte ENI

InboundQueryVolume

OutboundQueryVolume

OutboundQueryAggregateVolume

AWS Network Firewall groupe de règles

CloudWatch Application Insights prend en charge les mesures suivantes :

FirewallRuleGroupQueryVolume

AWS Network Firewall association de groupes de règles

CloudWatch Application Insights prend en charge les mesures suivantes :

FirewallRuleGroupVpcQueryVolume

Métriques avec exigences aux points de données

Pour les métriques sans seuil d'alerte évident par défaut, Application Insights attend que la métrique comporte suffisamment de points de données pour prédire un seuil d'alerte raisonnable. Les points de données métriques qu' CloudWatch Application Insights vérifie avant de créer une alarme sont les suivants :

- La métrique possède au moins 100 points de données qui datent de 2 à 15 jours.
- La métrique possède au moins 100 points de données qui datent de la veille.

Les métriques suivantes respectent ces exigences aux points de données. Notez que les métriques des CloudWatch agents nécessitent jusqu'à une heure pour créer des alarmes.

Métriques

- AWS/ApplicationELB
- AW/ AutoScaling
- AWS/EC2
- Elastic Block Store (EBS)
- AWS/ELB
- AWS/RDS
- AWS/Lambda
- AWS/SQS
- AWS/CWAgent
- AWS/DynamoDB
- AWS/S3
- AWS/States
- AW/ ApiGateway
- AWS/SNS

AWS/ApplicationELB

ActiveConnectionCount

ConsumedLCUs

HTTPCode_ELB_4XX_Count

HTTPCode_Target_2XX_Count

HTTPCode_Target_3XX_Count

HTTPCode_Target_4XX_Count

HTTPCode_Target_5XX_Count

NewConnectionCount

ProcessedBytes

TargetResponseTime

UnHealthyHostCount

AW/ AutoScaling

GroupDesiredCapacity

GroupInServiceInstances

GroupMaxSize

GroupMinSize

GroupPendingInstances

GroupStandbyInstances

GroupTerminatingInstances

GroupTotalInstances

AWS/EC2

CPU CreditBalance

CPU CreditUsage

CPU SurplusCreditBalance

CPU SurplusCreditsCharged

CPUUtilization

DiskReadBytes

DiskReadOps

DiskWriteBytes

DiskWriteOps

% EBS ByteBalance

EBSIOBalance%

EBS ReadBytes

EBS ReadOps

EBS WriteBytes

EBS WriteOps

NetworkIn

NetworkOut

NetworkPacketsIn

NetworkPacketsOut

Elastic Block Store (EBS)

VolumeReadBytes

VolumeWriteBytes

VolumeReadOps

VolumeWriteOps

VolumeTotalReadTime

VolumeTotalWriteTime

VolumeIdleTime

VolumeQueueLength

VolumeThroughputPercentage

VolumeConsumedReadWriteOps

BurstBalance

AWS/ELB

Dalb estimé ActiveConnectionCount

EstimatedALBConsumedLCUs

Dalb estimé NewConnectionCount

EstimatedProcessedBytes

HTTPCode_Backend_4XX

HTTPCode_Backend_5XX

HealthyHostCount

Latence

RequestCount

SurgeQueueLength

UnHealthyHostCount

AWS/RDS

ActiveTransactions

AuroraBinlogReplicaLag

AuroraReplicaLag

BackupRetentionPeriodStorageUsed

BinLogDiskUsage

BlockedTransactions

CPU CreditBalance

CommitLatency

CommitThroughput

DDLLatency

DDLThroughput

DMLLatency

DMLThroughput

DatabaseConnections

Deadlocks

DeleteLatency

DeleteThroughput

DiskQueueDepth

EngineUptime

FreeLocalStorage

FreeStorageSpace

FreeableMemory

InsertLatency

InsertThroughput

LoginFailures

NetworkReceiveThroughput

NetworkThroughput

NetworkTransmitThroughput

Requêtes

ReadIOPS

ReadThroughput

SelectLatency

SelectThroughput

SnapshotStorageUsed

TotalBackupStorageBilled

UpdateLatency

UpdateThroughput

VolumeBytesUsed

VolumeReadIOPS

VolumeWriteIOPS

WriteIOPS

WriteThroughput

AWS/Lambda

Erreurs

DeadLetterErrors

Durée

Throttles

IteratorAge

ProvisionedConcurrencySpilloverInvocations

AWS/SQS

ApproximateAgeOfOldestMessage

ApproximateNumberOfMessagesDelayed

ApproximateNumberOfMessagesNotVisible

ApproximateNumberOfMessagesVisible

NumberOfEmptyReceives

NumberOfMessagesDeleted

NumberOfMessagesReceived

NumberOfMessagesSent

AWS/CWAgent

LogicalDisk % d'espace libre

% d'octets validés de mémoire en cours d'utilisation

Mo de mémoire disponible

Total octets interface réseau/sec

Utilisation en % du fichier de pagination

PhysicalDisk % de temps sur le disque

PhysicalDisk Moyenne. Disk sec/Read (Lecture/s sur disque)

PhysicalDisk Moyenne. Disk Write/sec (Écriture/s sur disque)

PhysicalDisk Octets lus sur le disque par seconde

PhysicalDisk Lectures sur disque par seconde

PhysicalDisk Octets d'écriture sur disque par seconde

PhysicalDisk Nombre d'écritures sur disque par seconde

Temps d'inactivité en % du processeur

Durée d'interruption en % du processeur

Temps de traitement en % du processeur

Temps utilisateur en % du processeur

SQLServer : Méthodes d'accès - Enregistrements transmis/sec

SQLServer : Pages de méthodes - Splits de pages/sec

SQLServer : Taux de réussite du cache du gestionnaire de tampon

SQLServer : Espérance de vie de la page du gestionnaire de tampon

SQLServer : Octets du fichier de réplica de base de données reçus/seconde

SQLServer : Octets du journal de réplica de base de données reçus/seconde

SQLServer : Journal de réplica de base de données restant à annuler

SQLServer : File d'attente d'envoi du journal de réplica de base de données

SQLServer : Transactions d'écriture en miroir du réplica de base de données/seconde

SQLServer : File d'attente de récupération du réplica de base de données

SQLServer : Octets restants de rétablissement du réplica de base de données

SQLServer : Octets rétablis du réplica de base de données/seconde

SQLServer : Nombre total de journaux du réplica de base de données nécessitant une annulation

SQLServer : Retard de transaction du réplica de base de données

SQLServer : Statistiques générales - Processus bloqués

SQLServer : Statistiques SQL - Requêtes par lots/sec

SQLServer : Statistiques SQL - Compilations SQL/sec

SQLServer : Statistiques SQL - Recompilations SQL/sec

Longueur de la file d'attente du processeur système

Connexions TCPv4 établies

Connexions TCPv6 établies

AWS/DynamoDB

ConsumedReadCapacityUnits

ConsumedWriteCapacityUnits

ReadThrottleEvents

WriteThrottleEvents

TimeToLiveDeletedItemCount

ConditionalCheckFailedRequests

TransactionConflict

ReturnedRecordsCount

PendingReplicationCount

ReplicationLatency

AWS/S3

ReplicationLatency

BytesPendingReplication

OperationsPendingReplication

4xxErrors

5xxErrors

AllRequests

GetRequests

PutRequests

DeleteRequests

HeadRequests

PostRequests

SelectRequests

ListRequests

SelectScannedBytes

SelectReturnedBytes

FirstByteLatency

TotalRequestLatency

BytesDownloaded

BytesUploaded

AWS/States

ActivitiesScheduled

ActivitiesStarted

ActivitiesSucceeded

ActivityScheduleTime

ActivityRuntime

ActivityTime

LambdaFunctionsScheduled

LambdaFunctionsStarted

LambdaFunctionsSucceeded

LambdaFunctionScheduleTime

LambdaFunctionRuntime

LambdaFunctionTime

ServiceIntegrationsScheduled

ServiceIntegrationsStarted

ServiceIntegrationsSucceeded

ServiceIntegrationScheduleTime

ServiceIntegrationRuntime

ServiceIntegrationTime

ProvisionedRefillRate

ProvisionedBucketSize

ConsumedCapacity

ThrottledEvents

AW/ ApiGateway

4XXError

IntegrationLatency

Latence

DataProcessed

CacheHitCount

CacheMissCount

AWS/SNS

NumberOfNotificationsDelivered

NumberOfMessagesPublished

NumberOfNotificationsFailed

NumberOfNotificationsFilteredOut

NumberOfNotificationsFilteredOut-InvalidAttributes

NumberOfNotificationsFilteredOut-NoMessageAttributes

NumberOfNotificationsRedrivenToDlq

NumberOfNotificationsFailedToRedriveToDlq

SMS SuccessRate

Métriques recommandées

Le tableau suivant répertorie les métriques recommandées pour chaque type de composant.

Type de composant	Type de charge de travail	Métrique recommandée
Instance EC2 (serveurs Windows)	Par défaut / Personnalisée	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation LogicalDisk % d'espace libre Mo de mémoire disponible
	Active Directory	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation Mo de mémoire disponible Base de données ==> Résultats en du cache de bases de données des instances DirectoryServices DRA en attente d'opérations de réplication DirectoryServices SYNCHRONISATIONS DE RÉPLICATION DRA EN ATTENTE

Type de composant	Type de charge de travail	Métrique recommandée
		Échec de la requête récursive DNS /s LogicalDisk Moyenne. Longueur de la file d'attente de disque
	Application Java	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation Mo de mémoire disponible java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Type de composant	Type de charge de travail	Métrique recommandée
	Interface Web Microsoft IIS/.NET	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation Mo de mémoire disponible Exceptions CLR .NET Nombre d'exceptions émises/s Mémoire CLR .NET # Nombre total d'octets engagés Mémoire CLR .NET % de temps en GC Applications ASP.NET Demandes en file d'attente d'application Demandes ASP.NET en file d'attente L'application ASP.NET redémarre

Type de composant	Type de charge de travail	Métrique recommandée
	Niveau de base de données Microsoft SQL Server	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation Mo de mémoire disponible Utilisation en % du fichier de pagination Longueur de la file d'attente du processeur système Total octets interface réseau/s PhysicalDisk % de temps sur le disque SQLServer : Taux de réussite du cache du gestionnaire de tampon SQLServer : Espérance de vie de la page du gestionnaire de tampon SQLServer : Statistiques générales - Processus bloqués SQLServer : Statistiques générales - Connexions utilisateurs

Type de composant	Type de charge de travail	Métrique recommandée
		SQLServer : Verrouillages - Nombre de blocages/s SQLServer : Statistiques SQL - Requêtes par lots/sec
	MySQL	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation LogicalDisk % d'espace libre Mo de mémoire disponible

Type de composant	Type de charge de travail	Métrique recommandée
	.NET workerpool/Niveau intermédiaire	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation Mo de mémoire disponible Exceptions CLR .NET Nombre d'exceptions émises/s Mémoire CLR .NET # Nombre total d'octets engagés Mémoire CLR .NET % de temps en GC
	Niveau .NET Core	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation Mo de mémoire disponible

Type de composant	Type de charge de travail	Métrique recommandée
	Oracle	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation LogicalDisk % d'espace libre Mo de mémoire disponible
	Postgres	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation LogicalDisk % d'espace libre Mo de mémoire disponible

Type de composant	Type de charge de travail	Métrique recommandée
	SharePoint	CPUUtilization StatusCheckFailed Temps de traitement en % du processeur % d'octets validés de mémoire en cours d'utilisation Mo de mémoire disponible Tronquage des API du cache des applications ASP.NET Demandes ASP.NET rejetées Le processus ASP.NET Worker redémarre Pages mémoire/s SharePoint Cache de publication Le cache de publication est vidé par seconde SharePoint Durée d'exécution/demande de page de la Fondation SharePoint Cache sur disque Nombre total de compactages de cache SharePoint Taux de réussite du cache Blob basé sur le disque

Type de composant	Type de charge de travail	Métrique recommandée
		SharePoint Taux de remplissage du cache blob basé sur disque SharePoint Cache sur disque : le cache blob est vidé par seconde Demandes ASP.NET en file d'attente Applications ASP.NET Demandes en file d'attente d'applications L'application ASP.NET redémarre LogicalDisk Moyenne. Disk Write/sec (Écriture/s sur disque) LogicalDisk Moyenne. Disk sec/Read (Lecture/s sur disque) Durée d'interruption en % du processeur
Instance EC2 (serveurs Linux)	Par défaut / Personnalisée	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Type de composant	Type de charge de travail	Métrique recommandée
	Application Java	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize
	Niveau .NET Core ou niveau Base de données SQL Server	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Type de composant	Type de charge de travail	Métrique recommandée
	Oracle	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent
	Postgres	CPUUtilization StatusCheckFailed disk_used_percent mem_used_percent

Type de composant	Type de charge de travail	Métrique recommandée
Groupe d'instances EC2	Nœuds multiples ou nœuds uniques SAP HANA	• hanadb_server_startup_time_variations_seconds • hanadb_level_5_alerts_count • hanadb_level_4_alerts_count • hanadb_out_of_memory_events_count • hanadb_max_trigger_read_ratio_percent • hanadb_max_trigger_write_ratio_percent • hanadb_log_switch_race_ratio_percent • hanadb_time_since_last_savepoint_seconds • hanadb_disk_usage_highlevel_percent • hanadb_current_allocation_limit_used_percent • hanadb_table_allocation_limit_used_percent • hanadb_cpu_usage_percent • hanadb_plan_cache_hit_ratio_percent • hanadb_last_data_backup_age_days

Type de composant	Type de charge de travail	Métrique recommandée
Volume EBS	N'importe quel compte	VolumeReadBytes VolumeWriteBytes VolumeReadOps VolumeWriteOps VolumeQueueLength VolumeThroughputPercentage VolumeConsumedReadWriteOps BurstBalance
ELB classique	N'importe quel compte	HTTPCode_Backend_4XX HTTPCode_Backend_5XX Latence SurgeQueueLength UnHealthyHostCount
Application ELB	N'importe quel compte	HTTPCode_Target_4XX_Count HTTPCode_Target_5XX_Count TargetResponseTime UnHealthyHostCount

Type de composant	Type de charge de travail	Métrique recommandée
Base de données d'instance RDS	N’importe quel compte	CPUUtilization ReadLatency WriteLatency BurstBalance SQL défaillant ServerAgentJobsCount
Cluster de Base de données RDS	N’importe quel compte	CPUUtilization CommitLatency DatabaseConnections Deadlocks FreeableMemory NetworkThroughput VolumeBytesUsed
Fonction Lambda	N’importe quel compte	Durée Erreurs IteratorAge ProvisionedConcurrencySpilloverInvocations Throttles

Type de composant	Type de charge de travail	Métrique recommandée
File d'attente SQS	N’importe quel compte	ApproximateAgeOfOldestMessage ApproximateNumberOfMessagesVisible NumberOfMessagesSent
Table Amazon DynamoDB	N’importe quel compte	SystemErrors UserErrors ConsumedReadCapacityUnits ConsumedWriteCapacityUnits ReadThrottleEvents WriteThrottleEvents ConditionalCheckFailedRequests TransactionConflict

Type de composant	Type de charge de travail	Métrique recommandée
Compartiment Amazon S3	N’importe quel compte	Si la configuration de réplication avec Replication Time Control (RTC) est activée : ReplicationLatency BytesPendingReplication OperationsPendingReplication Si les métriques de demande sont activées : 5xxErrors 4xxErrors BytesDownloaded BytesUploaded

Type de composant	Type de charge de travail	Métrique recommandée
AWS Step Functions	N’importe quel compte	Général • ExecutionThrottled • ExecutionsAborted • ProvisionedBucketSize • ProvisionedRefillRate • ConsumedCapacity Si le type de machine d'état est **`EXPRESS`** ou le niveau du groupe de journaux est **`OFF`** • ExecutionsFailed • ExecutionsTimedOut Si la machine d'état a des fonctions Lambda • LambdaFunctionsFailed • LambdaFunctionsTimedOut Si la machine d'état a des activités • ActivitiesFailed • ActivitiesTimedOut • ActivitiesHeartbea tTimedOut Si la machine d'état a des intégrations de service • ServiceIntegrationsFailed

Type de composant	Type de charge de travail	Métrique recommandée
		• ServiceIntegration sTimedOut
Étape d'API REST API Gateway	N’importe quel compte	• 4XXErrors • 5XXErrors • Latence

Type de composant	Type de charge de travail	Métrique recommandée
Cluster ECS	N’importe quel compte	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (type de lancement EC2 uniquement) CPUUtilization (type de lancement EC2 uniquement) MemoryReservation (Type de lancement EC2 uniquement) MemoryUtilization (Type de lancement EC2 uniquement) GPUReservation (type de lancement EC2 uniquement) instance_cpu_utilization (type de lancement EC2 uniquement) instance_filesystem_utilization (type de lancement EC2 uniquement)

Type de composant	Type de charge de travail	Métrique recommandée
		instance_memory_utilization (type de lancement EC2 uniquement) instance_network_total_bytes (Type de lancement EC2 uniquement)

Type de composant	Type de charge de travail	Métrique recommandée
	Application Java	CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes CPUReservation (type de lancement EC2 uniquement) CPUUtilization (type de lancement EC2 uniquement) MemoryReservation (Type de lancement EC2 uniquement) MemoryUtilization (Type de lancement EC2 uniquement) GPUReservation (type de lancement EC2 uniquement) instance_cpu_utilization (type de lancement EC2 uniquement) instance_filesystem_utilization (type de lancement EC2 uniquement)

Type de composant	Type de charge de travail	Métrique recommandée
		instance_memory_utilization (type de lancement EC2 uniquement) instance_network_total_bytes (Type de lancement EC2 uniquement) java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Type de composant	Type de charge de travail	Métrique recommandée
Service ECS	N’importe quel compte	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes

Type de composant	Type de charge de travail	Métrique recommandée
	Application Java	CPUUtilization MemoryUtilization CpuUtilized MemoryUtilized NetworkRxBytes NetworkTxBytes RunningTaskCount PendingTaskCount StorageReadBytes StorageWriteBytes java_lang_threading_threadcount java_lang_classloading_loadedclasscount java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Type de composant	Type de charge de travail	Métrique recommandée
Cluster EKS	N’importe quel compte	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_utilization
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_utilization
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_utilization
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Type de composant	Type de charge de travail	Métrique recommandée
	Application Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserv ed_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_po d_limit pod_memory_reserve d_capacity pod_memory_utilization pod_memory_utilization_over _pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadc ount java_lang_classloading_load edclasscount

Type de composant	Type de charge de travail	Métrique recommandée
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Type de composant	Type de charge de travail	Métrique recommandée
Cluster Kubernetes sur EC2	N’importe quel compte	cluster_failed_node_count
		node_cpu_reserved_capacity
		node_cpu_utilization
		node_filesystem_utilization
		node_memory_reserv ed_capacity
		node_memory_utilization
		node_network_total_bytes
		pod_cpu_reserved_capacity
		pod_cpu_utilization
		pod_cpu_utilization_over_po d_limit
		pod_memory_reserve d_capacity
		pod_memory_utilization
		pod_memory_utilization_over _pod_limit
		pod_network_rx_bytes
		pod_network_tx_bytes

Type de composant	Type de charge de travail	Métrique recommandée
	Application Java	cluster_failed_node_count node_cpu_reserved_capacity node_cpu_utilization node_filesystem_utilization node_memory_reserv ed_capacity node_memory_utilization node_network_total_bytes pod_cpu_reserved_capacity pod_cpu_utilization pod_cpu_utilization_over_po d_limit pod_memory_reserve d_capacity pod_memory_utilization pod_memory_utilization_over _pod_limit pod_network_rx_bytes pod_network_tx_bytes java_lang_threading_threadc ount java_lang_classloading_load edclasscount

Type de composant	Type de charge de travail	Métrique recommandée
		java_lang_memory_heapmemoryusage_used java_lang_memory_heapmemoryusage_committed java_lang_operatingsystem_freephysicalmemorysize java_lang_operatingsystem_freeswapspacesize

Le tableau suivant répertorie les processus recommandés et les mesures de processus pour chaque type de composant. CloudWatch Application Insights ne recommande pas de surveiller les processus qui ne s'exécutent pas sur une instance.

Type de composant	Type de charge de travail	Processus recommandé	Métrique recommandée

Instance EC2 (serveurs Windows)	Interface Web Microsoft IIS/.NET	`w3wp`	`procstat cpu_usage ,`
			
			`procstat memory_rss ,`
			
			`procstat memory_vms ,`
			
			`procstat read_bytes ,`
			
			`procstat write_bytes`

	Niveau de base de données Microsoft SQL Server	`SQLAgent`	`procstat cpu_usage ,`

Type de composant	Type de charge de travail	Processus recommandé	Métrique recommandée
			procstat memory_rss , procstat memory_vms , procstat read_bytes , procstat write_bytes
		sqlservr	procstat cpu_usage , procstat memory_rss , procstat memory_vms , procstat read_bytes , procstat write_bytes
		sqlwriter	procstat cpu_usage , procstat memory_rss

Type de composant	Type de charge de travail	Processus recommandé	Métrique recommandée
		Reporting ServicesService	procstat cpu_usage , procstat memory_rss
		MsDtsServr	procstat cpu_usage , procstat memory_rss , procstat memory_vms , procstat read_bytes , procstat write_bytes
		Msmdsrv	procstat cpu_usage , procstat memory_rss , procstat memory_vms , procstat read_bytes , procstat write_bytes

Type de composant	Type de charge de travail	Processus recommandé	Métrique recommandée
	.NET workerpool/ Niveau intermédiaire	`w3wp`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`
	Niveau .NET Core	`w3wp`	`procstat cpu_usage` , `procstat memory_rss` , `procstat memory_vms` , `procstat read_bytes` , `procstat write_bytes`

Métriques du compteur de performances

Les métriques du compteur de performances sont recommandées pour les instances uniquement lorsque les jeux de compteur de performances correspondants sont installés sur les instances Windows.

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
Exceptions CLR .NET Nombre d'exceptions émises	Exceptions CLR .NET
Exceptions CLR .NET Nombre d'exceptions émises/s	Exceptions CLR .NET
Exceptions CLR .NET Nombre de filtres/s	Exceptions CLR .NET
Exceptions CLR .NET Nombre de finalements/s	Exceptions CLR .NET
Exceptions CLR .NET Lancer pour capter la profondeur/s	Exceptions CLR .NET
Interop CLR .NET Nombre de CCW	Interop CLR .NET
Interop CLR .NET Nombre de talons	Interop CLR .NET
Interop CLR .NET Nombre d'exportations LTB/s	Interop CLR .NET
Interop CLR .NET Nombre d'importations TLB/s	Interop CLR .NET
Interop CLR .NET Nombre de regroupements	Interop CLR .NET
Jit CLR .NET % temps en juste-à-temps	Jit CLR .NET
Jit CLR .NET Échecs juste-à-temps standard	Jit CLR .NET
Chargement CLR .NET % temps de chargement	Chargement CLR .NET
Chargement CLR .NET Taux des échecs de chargement	Chargement CLR .NET
Taux de LocksAndThreads contention .NET CLR par seconde	.NET CLR LocksAndThreads

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
Longueur de file d' attente .NET CLR par seconde	LocksAndThreads .NET CLR LocksAndThreads
Mémoire CLR .NET # Nombre total d'octets engagés	Mémoire CLR .NET
Mémoire CLR .NET % de temps en GC	Mémoire CLR .NET
.NET CLR Networking 4.0.0.0 HttpWebRequest Temps d'attente moyen	Mise en réseau CLR .NET 4.0.0.0
Mise en réseau .NET CLR HttpWebRequests 4.0.0.0 abandonnée par seconde	Mise en réseau CLR .NET 4.0.0.0
Le réseau .NET CLR HttpWebRequests 4.0.0.0 a échoué/sec	Mise en réseau CLR .NET 4.0.0.0
.NET CLR Networking HttpWebRequests 4.0.0.0 en file d'attente/sec	Mise en réseau CLR .NET 4.0.0.0
APP_POOL_WAS Total des échecs de ping de processus de travail	APP_POOL_WAS
L'application ASP.NET redémarre	ASP.NET
Demandes ASP.NET rejetées	ASP.NET
Le processus ASP.NET Worker redémarre	ASP.NET
Tronquage des API du cache des applications ASP.NET	Applications ASP.NET
Applications ASP.NET Temps de traitement en % du processeur géré (estimé)	Applications ASP.NET
Applications ASP.NET Total erreurs/s	Applications ASP.NET

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
Applications ASP.NET Erreurs non traitées pendant l'exécution/s	Applications ASP.NET
Applications ASP.NET Demandes en file d'attente d'application	Applications ASP.NET
Applications ASP.NET Demandes/s	Applications ASP.NET
Temps d'attente de la demande ASP.NET	ASP.NET
Demandes ASP.NET en file d'attente	ASP.NET
Base de données ==> Résultats en du cache de bases de données des instances	Base de données ==> Instances
Base de données ==> Latence de moyenne de lectures de bases de données I/O d'instances	Base de données ==> Instances
Base de données ==> Lectures de bases de données I/O /s	Base de données ==> Instances
Base de données ==> Latence de moyenne d'écritures de journaux I/O d'instances	Base de données ==> Instances
DirectoryServices DRA en attente d'opérations de réplication	DirectoryServices
DirectoryServices SYNCHRONISATIONS DE RÉPLICATION DRA EN ATTENTE	DirectoryServices
DirectoryServices Heure de liaison LDAP	DirectoryServices
Requêtes récursives DNS par seconde	DNS
Échec de la requête récursive DNS /s	DNS
Requête DNS TCP reçue/s	DNS

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
Total DNS Requête reçue/s	DNS
Réponse totale DNS envoyée/s	DNS
Requête DNS TCP reçue/s	DNS
Files d'attente de service HTTP CurrentQueueSize	Files d'attente de demandes de service HTTP
LogicalDisk % d'espace libre	LogicalDisk
LogicalDisk Moyenne. Disk Write/sec (Écriture/s sur disque)	LogicalDisk
LogicalDisk Moyenne. Disk sec/Read (Lecture/s sur disque)	LogicalDisk
LogicalDisk Moyenne. Longueur de la file d'attente de disque	LogicalDisk
% d'octets validés de mémoire en cours d'utilisation	Mémoire
Mo de mémoire disponible	Mémoire
Pages mémoires/s	Mémoire
Durée de vie moyenne du cache en veille à long terme de la mémoire (s)	Mémoire
Total octets interface réseau/s	Interface réseau
Octets d'interface réseau reçus/s	Interface réseau
Octets d'interface réseau envoyés/sec	Interface réseau
Bande passante actuelle de l'interface réseau	Interface réseau

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
Utilisation en % du fichier de pagination	Fichier de pagination
PhysicalDisk % de temps sur le disque	PhysicalDisk
PhysicalDisk Moyenne. Longueur de la file d'attente de disque	PhysicalDisk
PhysicalDisk Moyenne. Lecture/s sur disque	PhysicalDisk
PhysicalDisk Moyenne. Écriture/s sur disque	PhysicalDisk
PhysicalDisk Octets lus sur le disque par seconde	PhysicalDisk
PhysicalDisk Lectures sur disque par seconde	PhysicalDisk
PhysicalDisk Octets d'écriture sur disque par seconde	PhysicalDisk
PhysicalDisk Nombre d'écritures sur disque par seconde	PhysicalDisk
Temps d'inactivité en % du processeur	Processeur
Durée d'interruption en % du processeur	Processeur
Temps de traitement en % du processeur	Processeur
Temps utilisateur en % du processeur	Processeur
SharePoint Taux de remplissage du cache blob basé sur disque	SharePoint Cache sur disque
SharePoint Cache sur disque : le cache blob est vidé par seconde	SharePoint Cache sur disque
SharePoint Taux de réussite du cache Blob basé sur le disque	SharePoint Cache sur disque

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
SharePoint Cache sur disque Nombre total de compactages de cache	SharePoint Cache sur disque
SharePoint Durée d'exécution/demande de page de la Fondation	SharePoint Fondation
SharePoint Cache de publication Le cache de publication est vidé par seconde	SharePoint Cache de publication
Authentications Kerberos des statistiques de sécurité à l'échelle du système	Statistiques de sécurité à l'échelle du système
Authentications NTLM des statistiques de sécurité à l'échelle du système	Statistiques de sécurité à l'échelle du système
SQLServer : Méthodes d'accès - Enregistrements transmis/sec	SQLServer : méthodes d'accès
SQLServer : Méthodes d'accès - Analyses complètes/s	SQLServer : méthodes d'accès
SQLServer : Pages de méthodes - Splits de pages/s	SQLServer : méthodes d'accès
SQLServer : Taux de résultats du cache du gestionnaire de tampon	SQLServer : gestionnaire de mémoire tampon
SQLServer : Espérance de vie de la page du gestionnaire de tampon	SQLServer : gestionnaire de mémoire tampon
SQLServer : Octets du fichier de réplica de base de données reçus/seconde	SQLServer : réplica de base de données
SQLServer : Octets du journal de réplica de base de données reçus/seconde	SQLServer : réplica de base de données

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
SQLServer : Journal de réplica de base de données restant à annuler	SQLServer : réplica de base de données
SQLServer : File d'attente d'envoi du journal de réplica de base de données	SQLServer : réplica de base de données
SQLServer : Transactions d'écriture en miroir du réplica de base de données/seconde	SQLServer : réplica de base de données
SQLServer : File d'attente de récupération du réplica de base de données	SQLServer : réplica de base de données
SQLServer : Octets restants de rétablissement du réplica de base de données	SQLServer : réplica de base de données
SQLServer : Octets rétablis du réplica de base de données/seconde	SQLServer : réplica de base de données
SQLServer : Nombre total de journaux du réplica de base de données nécessitant une annulation	SQLServer : réplica de base de données
SQLServer : Retard de transaction du réplica de base de données	SQLServer : réplica de base de données
SQLServer : Statistiques générales - Processus bloqués	SQLServer : statistiques générales
SQLServer : Statistiques générales - Connexions utilisateurs	SQLServer : statistiques générales
SQLServer : Moyenne loquets - Temps d'attente du loquet (ms)	SQLServer : Loquets
SQLServer : Verrouillages - Temps d'attente moyen (ms)	SQLServer : Verrouillages

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
SQLServer : Verrouillages - Verrouillage des temporisations/s	SQLServer : Verrouillages
SQLServer : Verrouillages - Attente de verrouillage/s	SQLServer : Verrouillages
SQLServer : Verrouillages - Nombre de blocages/s	SQLServer : Verrouillages
SQLServer : Gestionnaire de mémoire - Octrois de mémoire en attente	SQLServer : Gestionnaire de mémoire
SQLServer : Statistiques SQL - Requêtes par lots/sec	SQLServer : statistiques SQL
SQLServer : Statistiques SQL - Compilations SQL/s	SQLServer : statistiques SQL
SQLServer : Statistiques SQL - Recompilations SQL/s	SQLServer : statistiques SQL
Longueur de la file d'attente du processeur système	Système
Connexions TCPv4 établies	TCPv4
Connexions TCPv6 établies	TCPv6
W3SVC_W3WP Vidages du cache de fichiers	W3SVC_W3WP
W3SVC_W3WP Échecs du cache de fichiers	W3SVC_W3WP
W3SVC_W3WP Demandes/s	W3SVC_W3WP
W3SVC_W3WP Vidages de cache d'URI	W3SVC_W3WP
W3SVC_W3WP Échecs de cache d'URI	W3SVC_W3WP

Nom de la métrique du compteur de performances	Nom de l'ensemble de compteurs de performances
Service Web Octets reçus/s	Services web
Service Web Octets envoyés/s	Services web
Tentatives/s de connexion au service Web	Services web
Service Web Connexions actuelles	Services web
Service Web Obtenir des requêtes/s	Services web
Demandes de publication de service Web/s	Services web

Utilisation de l'affichage de l'état des ressources dans la CloudWatch console

Vous pouvez utiliser l'affichage d'état des ressources pour découvrir, gérer et visualiser automatiquement l'état et les performances des hôtes dans leurs applications dans une vue unique. Vous pouvez visualiser l'état de leurs hôtes par une dimension de performance telle que l'UC ou la mémoire, et découper des centaines d'hôtes dans une seule vue à l'aide de filtres. Vous pouvez filtrer par identifications ou par cas d'utilisation, tels que les hôtes du même groupe Auto Scaling ou les hôtes qui utilisent le même équilibreur de charge,

Prérequis

Pour vous assurer de bénéficier pleinement de l'affichage de l'état des ressources, vérifiez que les conditions préalables suivantes sont remplies.

- Pour connaître l'utilisation de la mémoire par vos hôtes et l'utiliser comme filtre, vous devez installer l' CloudWatch agent sur vos hôtes et le configurer pour envoyer une métrique de mémoire CloudWatch dans l'espace de `CWAgent` noms par défaut. Sur les instances Linux et macOS, l' CloudWatch agent doit envoyer la `mem_used_percent` métrique. Sur les instances Windows, l'agent doit envoyer la métrique `Memory % Committed Bytes In Use`. Ces métriques sont incluses si vous utilisez l'assistant pour créer le fichier de configuration de l' CloudWatch agent et sélectionnez l'un des ensembles de métriques prédéfinis. Les métriques collectées par l'

CloudWatch agent sont facturées comme des métriques personnalisées. Pour plus d'informations, consultez Installation de l' CloudWatch agent.

Lorsque vous utilisez l' CloudWatch agent pour collecter ces métriques de mémoire à utiliser avec la vue de l'état des ressources, vous devez inclure la section suivante dans le fichier de configuration de l' CloudWatch agent. Cette section contient les paramètres de dimension par défaut et est créée par défaut. Par conséquent, ne modifiez aucune partie de cette section à quelque chose de différent de ce qui est indiqué dans l'exemple suivant.

```
"append_dimensions": {
  "ImageId": "${aws:ImageId}",
  "InstanceId": "${aws:InstanceId}",
  "InstanceType": "${aws:InstanceType}",
  "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
},
```

- Pour afficher toutes les informations disponibles dans l'affichage de l'état de la ressource, vous devez être connecté à un compte disposant des autorisations suivantes. Si vous êtes connecté avec moins d'autorisations, vous pouvez toujours utiliser la vue données d'état des ressources, mais certaines données de performances ne seront pas visibles.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:Describe*",
                "cloudwatch:Describe*",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:Describe*",
                "sns:Get*",
                "sns:List*",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeRegions"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
```

```
    ]
}
```

Pour afficher l'état des ressources dans votre compte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, choisissez Surveillance de l’infrastructure, État des ressources.

   La page d'état de la ressource apparaît et affiche un carré pour chaque hôte de votre compte. Chaque carré est coloré en fonction de l'état actuel de cet hôte, en fonction du paramètre Color by (Couleur par). Les carrés de l'hôte avec un symbole d'alerte comportent une ou plusieurs alertes actuellement en état alerte.

   Vous pouvez voir jusqu'à 500 hôtes dans une seule vue. Si vous avez plus d'hôtes dans votre compte, utilisez les paramètres de filtre à l'étape 6 de cette procédure.
3. Pour modifier les critères utilisés pour afficher l'état de santé de chaque hôte, choisissez un paramètre pour Color by (Couleur par). Vous pouvez choisir CPU Utilization (Utilisation de l'UC), puis Memory Utilization (Utilisation de la mémoire) ou Status check (Contrôle d'états). Les métriques d'utilisation de la mémoire ne sont disponibles que pour les hôtes qui exécutent l' CloudWatch agent et qui l'ont configuré pour collecter des métriques de mémoire et les envoyer vers l'espace de `CWAgent` noms par défaut. Pour plus d’informations, consultez Collectez des métriques, des journaux et des traces avec l' CloudWatch agent.
4. Pour modifier les seuils et les couleurs utilisés pour les indicateurs d'état dans la grille, choisissez l'icône d'engrenage au-dessus de la grille.
5. Pour activer ou désactiver l'affichage des alertes dans la grille de l'hôte, sélectionnez ou désactivez Show alarms across all metrics (Afficher les alertes sur toutes les métriques).
6. Pour diviser les hôtes de la carte en groupes, choisissez un critère de regroupement pour Group by (Regrouper par).
7. Pour réduire la vue à moins d'hôtes, choisissez un critère de filtre pour Filter by (Filter par). Vous pouvez filtrer par identifications et par groupes de ressources tels que le groupe Auto Scaling, le type d'instance, le groupe de sécurité, etc.
8. Pour trier les hôtes, choisissez un critère de tri pour Sort by (Trier par). Vous pouvez trier les résultats du contrôle d'état, l'état de l'instance, l'utilisation de l'UC ou de la mémoire, et le nombre d'alertes qui sont en état alerte.

9. Pour afficher plus d'informations sur un hôte, choisissez le carré qui représente cet hôte. Un panneau contextuel s'affiche. Pour ensuite approfondir les informations sur cet hôte, choisissez View dashboard (Afficher le tableau de bord) ou View on list (Afficher sur la liste).

CloudWatch observabilité entre comptes

Grâce à l'observabilité CloudWatch entre comptes Amazon, vous pouvez surveiller et dépanner les applications qui couvrent plusieurs comptes au sein d'une même région. Recherchez, visualisez et analysez en toute simplicité vos indicateurs, journaux, traces, applications Application Insights et moniteurs Internet Monitor dans tous les comptes associés, sans limites de compte.

Configurez un ou plusieurs AWS comptes en tant que comptes de surveillance et associez-les à plusieurs comptes sources. Un compte de surveillance est un compte AWS central qui peut visualiser et interagir avec les données d'observabilité générées par les 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. Les données d'observabilité partagées peuvent inclure les types de télémétrie suivants :

- Métriques sur Amazon CloudWatch. Vous pouvez choisir de partager les métriques de tous les espaces de noms avec le compte de surveillance ou de les filtrer sur un sous-ensemble d'espaces de noms.
- Groupes de journaux dans Amazon CloudWatch Logs. Vous pouvez choisir de partager tous les groupes de journaux avec le compte de surveillance ou de les filtrer sur un sous-ensemble de groupes de journaux.
- Traces dans AWS X-Ray
- Applications dans Amazon CloudWatch Application Insights
- Moniteurs dans CloudWatch Internet Monitor

Pour créer des liens entre les comptes de surveillance et les comptes sources, vous pouvez utiliser la CloudWatch console. Vous pouvez également utiliser les commandes Observability Access Manager dans l'API AWS CLI and. Pour plus d'informations, consultez Référence d'API Observability Access Manager.

Un récepteur est une ressource qui représente un point d'attache dans un compte de surveillance. Les comptes sources peuvent être liés au récepteur pour partager des données d'observabilité. Chaque compte peut avoir un récepteur par région. Chaque récepteur est géré par le compte de surveillance sur lequel il se trouve. Un lien d'observabilité est une ressource qui représente le lien établi entre un compte source et un compte de surveillance. Les liens sont gérés par le compte source.

Pour une démonstration vidéo de la configuration de l'observabilité CloudWatch entre comptes, voir la vidéo suivante.

La rubrique suivante explique comment configurer l'observabilité CloudWatch entre comptes à la fois dans les comptes de surveillance et dans les comptes sources. Pour plus d'informations sur le tableau de CloudWatch bord inter-comptes inter-régions, voir. Console multicompte et multirégion CloudWatch

Utilisation d'Organizations pour les comptes sources

Il existe deux options pour associer les comptes sources à votre compte de surveillance. Vous pouvez utiliser une ou les deux options.

- Permet AWS Organizations de lier les comptes d'une organisation ou d'une unité organisationnelle au compte de surveillance.
- Connectez des AWS comptes individuels au compte de surveillance.

Nous vous recommandons d'utiliser Organizations afin que les nouveaux AWS comptes créés ultérieurement dans l'organisation soient automatiquement intégrés à l'observabilité entre comptes en tant que comptes sources.

Détails sur la liaison des comptes de surveillance et des comptes sources

- Chaque compte de surveillance peut être lié à 100 000 comptes sources.
- Chaque compte source peut partager des données avec jusqu'à cinq comptes de surveillance.
- Vous pouvez configurer un seul compte en tant que compte de surveillance et compte source. Dans ce cas, ce compte envoie uniquement les données d'observabilité de lui-même au compte de surveillance auquel il est lié. Il ne transmet pas les données à partir de ses comptes sources.
- Un compte de surveillance spécifie les types de télémétrie qui peuvent être partagés avec lui. Un compte source spécifie les types de télémétrie qu'il veut partager.
  - S'il y a plus de types de télémétrie sélectionnés dans le compte de surveillance que dans le compte source, les comptes sont liés. Seuls les types de données sélectionnés dans les deux comptes sont partagés.
  - S'il y a plus de types de télémétrie sélectionnés dans le compte source que dans le compte de surveillance, la création du lien échoue et rien n'est partagé.

  - Le nom d'une métrique n'apparaît pas dans la console du compte de surveillance tant que cette métrique n'émet pas de nouveaux points de données après la création du lien.
- Pour supprimer un lien entre des comptes, faites-le depuis le compte source.
- Pour supprimer un récepteur d'un compte de surveillance, vous devez d'abord supprimer tous les liens vers ce récepteur du compte de surveillance.

Tarification

L'observabilité entre comptes n' CloudWatch entraîne aucun coût supplémentaire pour les journaux et les métriques, et la première copie de trace est gratuite. Pour plus d'informations sur les tarifs, consultez Amazon CloudWatch Pricing.

Table des matières

Liaison des comptes de surveillance avec les comptes sources

Les rubriques de cette section expliquent comment configurer les liens entre les comptes de surveillance et les comptes sources.

Nous vous recommandons de créer un nouveau AWS compte qui servira de compte de surveillance pour votre organisation.

Table des matières

Autorisations nécessaires

Pour créer des liens entre un compte de surveillance et un compte source, vous devez être connecté avec certaines autorisations.

- Pour configurer un compte de surveillance : vous devez avoir un accès administrateur complet dans le compte de surveillance, ou vous devez vous connecter à ce compte avec les autorisations suivantes :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSinkModification",
            "Effect": "Allow",
            "Action": [
                "oam:CreateSink",
                "oam:DeleteSink",
                "oam:PutSinkPolicy",
                "oam:TagResource"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "AllowReadOnly",
            "Effect": "Allow",
            "Action": ["oam:Get*", "oam:List*"],
            "Resource": "*"
        }
    ]
}
```

- Compte source, étendu à un compte de surveillance spécifique : pour créer, mettre à jour et gérer des liens pour un seul compte de surveillance spécifié, vous devez vous connecter au compte avec au moins les autorisations suivantes. Dans cet exemple, le compte de surveillance est 999999999999.

  Si le lien ne doit pas partager les cinq types de ressources (métriques, journaux, traces, applications Application Insights et moniteurs Internet Monitor), vous pouvez omettre`cloudwatch:Link`,, `logs:Link` `xray:Link``applicationinsights:Link`, ou `internetmonitor:Link` selon vos besoins.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink",
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:link/*"
        },
        {
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Effect": "Allow",
            "Resource": "arn:*:oam:*:*:sink/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": [
```

```
                        "999999999999"
                    ]
                }
            }
        },
        {
            "Action": "oam:ListLinks",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
         },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

- Compte source, autorisé à établir un lien vers n'importe quel compte de surveillance : pour créer un lien vers un compte de surveillance existant (récepteur) et partager des métriques, des groupes de journaux, des traces, des applications Application Insights et des moniteurs Internet Monitor, vous devez vous connecter au compte source avec les autorisations d'administrateur complètes ou vous y connecter avec les autorisations suivantes

Si le lien ne doit pas partager les cinq types de ressources (métriques, journaux, traces, applications Application Insights et moniteurs Internet Monitor), vous pouvez omettre`cloudwatch:Link`,, `logs:Link` `xray:Link``applicationinsights:Link`, ou `internetmonitor:Link` selon vos besoins.

```
{
    "Version": "2012-10-17",
    "Statement": [{
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:List*",
                "oam:Get*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Action": "cloudwatch:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "xray:Link",
```

```
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": "logs:Link",
            "Effect": "Allow",
            "Resource": "*"
        },
        {
             "Action": "applicationinsights:Link",
             "Effect": "Allow",
             "Resource": "*"
        },
        {
             "Action": "internetmonitor:Link",
             "Effect": "Allow",
             "Resource": "*"
        }
    ]
}
```

Présentation de la configuration

Les étapes de haut niveau suivantes vous montrent comment configurer l' CloudWatch observabilité entre comptes.

> **Note**
>
> Nous vous recommandons de créer un nouveau AWS compte à utiliser comme compte de surveillance pour votre organisation.

1. Configurez un compte de surveillance dédié.
2. (Facultatif) Téléchargez un AWS CloudFormation modèle ou copiez une URL pour lier les comptes source.
3. Liez les comptes sources au compte de surveillance.

Après avoir effectué ces étapes, vous pouvez utiliser le compte de surveillance pour visualiser les données d'observabilité des comptes sources.

Étape 1 : configurer un compte de surveillance

Suivez les étapes décrites dans cette section pour configurer un AWS compte en tant que compte de surveillance afin d'assurer l' CloudWatch observabilité entre comptes.

Prérequis

- Si vous configurez des comptes dans une AWS Organizations organisation en tant que comptes source, obtenez le chemin ou l'identifiant de l'organisation.
- Si vous n'utilisez pas d'organisations pour les comptes sources : obtenez les ID de compte des comptes sources.

Pour configurer un compte en tant que compte de surveillance, vous devez disposer de certaines autorisations. Pour plus d’informations, consultez Autorisations nécessaires.

Pour configurer un compte de surveillance

1. Connectez-vous au compte que vous voulez utiliser comme compte de surveillance.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation de gauche, choisissez Paramètres.
4. Dans Monitoring account configuration (Configuration du compte de surveillance), choisissez Configure (Configurer).
5. Pour Select data, choisissez si ce compte de surveillance sera en mesure d'afficher les journaux, les métriques, les traces, les informations sur les applications, et Internet Monitor - Surveille les données des comptes sources auxquels il est lié.
6. Pour List source accounts (Répertorier les comptes sources), saisissez les comptes sources que ce compte de surveillance pourra afficher. Pour identifier les comptes sources, entrez des ID de comptes individuels, des chemins d'organisation ou des ID d'organisation. Si vous saisissez un chemin d'organisation ou un ID d'organisation, ce compte de surveillance est autorisé à visualiser les données d'observabilité de tous les comptes liés dans cette organisation.

   Séparez les entrées de cette liste par des virgules.

> **Important**
>
> Lorsque vous entrez le chemin d'une organisation, suivez le format exact. L'ou-id doit se terminer par un / (une barre oblique). Par exemple : `o-a1b2c3d4e5/r-f6g7h8i9j0example/ou-def0-awsbbbb/`

7. Pour Define a label to identify your source account (Définir une étiquette pour identifier votre compte source), indiquez si vous souhaitez utiliser des noms de compte ou des adresses e-mail pour identifier les comptes sources lorsque vous utilisez le compte de surveillance pour les visualiser.
8. Choisissez Configurer.

> **Important**
>
> Le lien entre le compte de surveillance et le compte source n'est pas complet tant que vous n'avez pas configuré les comptes sources. Pour plus d'informations, consultez les sections suivantes.

Étape 2 : (Facultatif) Téléchargez un AWS CloudFormation modèle ou une URL

Pour lier les comptes sources à un compte de surveillance, nous vous recommandons d'utiliser un modèle AWS CloudFormation ou une URL.

- Si vous associez l'ensemble d'une organisation, CloudWatch fournit un AWS CloudFormation modèle.
- Si vous liez des comptes individuels, utilisez un AWS CloudFormation modèle ou une URL CloudWatch fournissant.

Pour utiliser un AWS CloudFormation modèle, vous devez le télécharger au cours de ces étapes. Une fois que vous avez lié le compte de surveillance à au moins un compte source, le AWS CloudFormation modèle n'est plus disponible au téléchargement.

Pour télécharger un AWS CloudFormation modèle ou copier une URL permettant de lier les comptes source au compte de surveillance

1. Connectez-vous au compte que vous voulez utiliser comme compte de surveillance.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation de gauche, choisissez Paramètres.
4. Dans Monitoring account configuration (Configuration du compte de surveillance), choisissez Resources to link accounts (Ressources pour lier les comptes).
5. Effectuez l'une des actions suivantes :
   - Choisissez Organisation AWS pour obtenir un modèle à utiliser pour lier les comptes d'une organisation à ce compte de surveillance.
   - Choisissez Any account (N'importe quel compte) pour obtenir un modèle ou une URL permettant de configurer des comptes individuels comme comptes sources.
6. Effectuez l'une des actions suivantes :
   - Si vous avez choisi AWS l'organisation, choisissez Télécharger le CloudFormation modèle.
   - Si vous avez choisi N'importe quel compte, choisissez Télécharger le CloudFormation modèle ou Copier l'URL.
7. (Facultatif) Répétez les étapes 5 et 6 pour télécharger le AWS CloudFormation modèle et l'URL.

Étape 3 : lier les comptes sources

Utilisez les étapes de ces sections pour lier les comptes sources à un compte de surveillance.

Pour lier les comptes de surveillance aux comptes sources, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez Autorisations nécessaires.

Utiliser un modèle AWS CloudFormation pour configurer tous les comptes d'une organisation ou d'une unité organisationnelle en tant que comptes sources

Ces étapes supposent que vous avez déjà téléchargé le AWS CloudFormation modèle nécessaire en effectuant les étapes décrites dansÉtape 2 : (Facultatif) Téléchargez un AWS CloudFormation modèle ou une URL.

Pour utiliser un AWS CloudFormation modèle pour lier les comptes d'une organisation ou d'une unité organisationnelle au compte de surveillance

1. Connectez-vous au compte de gestion de l'organisation.
2. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
3. Dans la barre de navigation de gauche, choisissez StackSets.
4. Vérifiez que vous êtes connecté à la région de votre choix, puis choisissez Create StackSet.
5. Choisissez Suivant.
6. Choisissez Template is ready (Le modèle est prêt), puis Upload a template file (Charger un fichier modèle).
7. Choisissez Choose file (Choisir un fichier), choisissez le modèle que vous avez téléchargé depuis le compte de surveillance, puis sélectionnez Open (Ouvrir).
8. Choisissez Suivant.
9. Pour Spécifier StackSet les détails, entrez un nom pour le StackSet et choisissez Next.
10. Pour Add stacks to stack set (Ajouter des piles à un ensemble de piles), sélectionnez Deploy new stacks (Déployer de nouvelles piles).
11. Pour Deployment targets (Cibles de déploiement), choisissez de déployer sur l'ensemble de l'organisation ou sur des unités organisationnelles spécifiées.
12. Pour Spécifier les régions, choisissez les régions dans lesquelles déployer l'observabilité CloudWatch entre comptes.
13. Choisissez Suivant.
14. Sur la page Review (Vérification), confirmez vos options sélectionnées et cliquez sur Submit (Soumettre).
15. Dans l'onglet Stack instances (Instances de pile), actualisez l'écran jusqu'à ce que vous voyiez que vos instances de pile ont le statut CREATE_COMPLETE.

Utilisation d'un modèle AWS CloudFormation pour configurer des comptes sources individuels

Ces étapes supposent que vous avez déjà téléchargé le AWS CloudFormation modèle nécessaire en effectuant les étapes décrites dansÉtape 2 : (Facultatif) Téléchargez un AWS CloudFormation modèle ou une URL.

Utiliser un AWS CloudFormation modèle pour configurer des comptes sources individuels pour une observabilité CloudWatch entre comptes

1. Connectez-vous au compte source.
2. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
3. Dans la barre de navigation de gauche, choisissez Stacks (Piles).
4. Vérifiez que vous êtes connecté à la région que vous voulez, puis choisissez Create stack (Créer une pile), With new resources (standard) (Avec de nouvelles ressources (standard)).
5. Choisissez Suivant.
6. Choisissez Charger un fichier de modèle.
7. Choisissez Choose file (Choisir un fichier), choisissez le modèle que vous avez téléchargé depuis le compte de surveillance, puis sélectionnez Open (Ouvrir).
8. Choisissez Suivant.
9. Pour Specify stack details (Spécifier les détails de la pile), saisissez un nom pour la pile et sélectionnez Next (Suivant).
10. Sur la page Configurer les options de pile, choisissez Suivant.
11. Sur la page Review (Vérification), choisissez Submit (Envoyer).
12. Sur la page d'état de votre pile, actualisez l'écran jusqu'à ce que vous voyiez que votre pile a le statut CREATE_COMPLETE.
13. Pour utiliser ce même modèle afin de lier d'autres comptes sources à ce compte de surveillance, déconnectez-vous de ce compte et connectez-vous au compte source suivant. Répétez ensuite les étapes 2 à 12.

Utilisation d'une URL pour configurer des comptes sources individuels

Ces étapes supposent que vous avez déjà copié l'URL nécessaire en effectuant les étapes de Étape 2 : (Facultatif) Téléchargez un AWS CloudFormation modèle ou une URL.

Pour utiliser une URL afin de lier des comptes sources individuels au compte de surveillance

1. Connectez-vous au compte que vous voulez utiliser comme compte source.
2. Saisissez l'URL que vous avez copiée à partir du compte de surveillance.

   Vous voyez la page des CloudWatch paramètres, avec certaines informations renseignées.

3. Pour Select data, choisissez si ce compte source partagera les journaux, les métriques, les traces, les informations sur les applications, et Internet Monitor - Surveille les données avec ce compte de surveillance.

   Pour les journaux et les métriques, vous pouvez choisir de partager toutes les ressources ou un sous-ensemble avec le compte de surveillance.

   a. (Facultatif) Pour partager un sous-ensemble des groupes de journaux de ce compte avec le compte de surveillance, sélectionnez Journaux, puis Filtrer les journaux. Utilisez ensuite la case Filtrer les journaux pour créer une requête afin de trouver les groupes de journaux que vous souhaitez partager. La requête utilisera le terme `LogGroupName` et un ou plusieurs des opérandes suivants.

      - = et `!=`
      - `AND`
      - `OR`
      - ^indique LIKE et `!^` NOT LIKE. Ils ne peuvent être utilisés que comme recherches de préfixes. Incluez un `%` à la fin de la chaîne que vous souhaitez rechercher et inclure.
      - `INetNOT IN`, en utilisant des parenthèses () `(  )`

      La requête complète ne doit pas comporter plus de 2 000 caractères et est limitée à cinq opérandes conditionnels. Les opérandes conditionnels sont `AND` et. `OR` Il n'y a pas de limite au nombre d'autres opérandes.

      > **Tip**
      >
      > Choisissez Afficher les exemples de requêtes pour voir la syntaxe correcte pour les formats de requête courants.

   b. (Facultatif) Pour partager un sous-ensemble des espaces de noms de métriques de ce compte avec le compte de surveillance, sélectionnez Metrics, puis Filtrer les métriques. Utilisez ensuite la zone Filtrer les métriques pour créer une requête afin de trouver les espaces de noms de métriques que vous souhaitez partager. Utilisez le terme `Namespace` et un ou plusieurs des opérandes suivants.

      - = et `!=`
      - `AND`

- `OR`
- `LIKE` et `NOT LIKE`. Ils ne peuvent être utilisés que comme recherches de préfixes. Incluez un `%` à la fin de la chaîne que vous souhaitez rechercher et inclure.
- `INetNOT IN`, en utilisant des parenthèses () ( )

La requête complète ne doit pas comporter plus de 2 000 caractères et est limitée à cinq opérandes conditionnels. Les opérandes conditionnels sont `AND` et. `OR` Il n'y a pas de limite au nombre d'autres opérandes.

> **Tip**
>
> Choisissez Afficher les exemples de requêtes pour voir la syntaxe correcte pour les formats de requête courants.

4. Ne modifiez pas l'ARN dans Enter monitoring account configuration ARN (Saisir l'ARN de configuration du compte de surveillance).
5. La section Define a label to identify your source account (Définir une étiquette pour identifier votre compte source) est pré-remplie avec le choix d'étiquette du compte de surveillance. En option, choisissez Edit (Modifier) pour le modifier.
6. Choisissez Lier.
7. Saisissez **`Confirm`** dans la case et sélectionnez Confirm (Confirmer).
8. Pour utiliser cette même URL afin de lier d'autres comptes sources à ce compte de surveillance, déconnectez-vous de ce compte et connectez-vous au compte source suivant. Répétez ensuite les étapes 2 à 7.

Gestion des comptes de surveillance et des comptes sources

Après avoir configuré vos comptes de surveillance et vos comptes sources, vous pouvez utiliser les étapes de ces sections pour les gérer.

Table des matières

Liaison de plus de comptes sources à un compte de surveillance existant

Suivez les étapes de cette section pour ajouter des liens de comptes sources supplémentaires à un compte de surveillance existant.

Chaque compte source peut être lié à cinq comptes de surveillance au maximum. Chaque compte de surveillance peut être lié à 100 000 comptes sources.

Pour gérer un compte source, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez Autorisations nécessaires.

Pour ajouter des comptes sources supplémentaires à un compte de surveillance

1. Connectez-vous au compte de surveillance.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation de gauche, choisissez Paramètres.
4. Dans Monitoring account configuration (Configuration du compte de surveillance), choisissez Manage source accounts (Gérer les comptes sources).
5. Choisissez l'onglet Configuration policy (Politique de configuration).
6. Dans la zone Configuration policy (Politique de configuration), ajoutez le nouvel ID de compte source dans la ligne Principal.

   Par exemple, supposons que la ligne Principal est actuellement la suivante :

   ```
   "Principal": {"AWS": ["111111111111", "222222222222"]}
   ```

   Pour ajouter 999999999999 comme troisième compte source, modifiez la ligne comme suit :

   ```
   "Principal": {"AWS": ["111111111111", "222222222222", "999999999999"]}
   ```
7. Choisissez Mettre à jour.
8. Choisissez l'onglet Configuration details (Détails de la configuration).
9. Choisissez l'icône de copie qui se trouve à côté de l'ARN récepteur du compte de surveillance.
10. Connectez-vous au compte que vous voulez utiliser comme nouveau compte source.
11. Collez l'ARN récepteur du compte de surveillance que vous avez copié à l'étape 9.

    Vous voyez la page des CloudWatch paramètres, avec certaines informations renseignées.

12. Pour Sélectionner les données, choisissez si ce compte source enverra les données Journaux, Métriques, Traces et Application Insights – Applications aux comptes de surveillance auxquels il est lié.
13. Ne modifiez pas l'ARN dans Enter monitoring account configuration ARN (Saisir l'ARN de configuration du compte de surveillance).
14. La section Define a label to identify your source account (Définir une étiquette pour identifier votre compte source) est pré-remplie avec le choix d'étiquette du compte de surveillance. En option, choisissez Edit (Modifier) pour le modifier.
15. Choisissez Lier.
16. Saisissez **`Confirm`** dans la case et sélectionnez Confirm (Confirmer).

Suppression du lien entre un compte de surveillance et un compte source

Suivez les étapes de cette section pour arrêter l'envoi de données d'un compte source vers un compte de surveillance.

Vous devez disposer des autorisations requises pour gérer un compte source afin de réaliser cette tâche. Pour plus d'informations, consultez Autorisations nécessaires.

Pour supprimer le lien entre un compte source et un compte de surveillance

1. Connectez-vous au compte source.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation de gauche, choisissez Paramètres.
4. Dans Source account information (Informations sur le compte source), choisissez View monitoring accounts (Afficher les comptes de surveillance).
5. Cochez la case en regard du compte de surveillance avec lequel vous voulez arrêter de partager des données.
6. Choisissez Stop sharing data (Arrêter le partage des données), Confirm (Confirmer).
7. Connectez-vous au compte de surveillance.
8. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
9. Sélectionnez Settings (Paramètres).
10. Dans Monitoring account information (Informations sur le compte de surveillance), choisissez View configuration (Afficher la configuration).

11. Dans la zone Policy (Politique), supprimez l'ID du compte source de la ligne Principal et sélectionnez Update (Mettre à jour).

Affichage des informations sur un compte de surveillance

Suivez les étapes de cette section pour afficher les paramètres inter-comptes d'un compte de surveillance.

Pour gérer un compte de surveillance, vous devez disposer de certaines autorisations. Pour plus d'informations, consultez Autorisations nécessaires.

Pour gérer un compte de surveillance

1. Connectez-vous au compte de surveillance.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le panneau de navigation de gauche, choisissez Paramètres.
4. Dans Monitoring account configuration (Configuration du compte de surveillance), choisissez Manage source accounts (Gérer les comptes sources).
5. Pour afficher la politique d'Observability Access Manager qui permet à ce compte d'être un compte de surveillance, choisissez l'onglet Configuration policy (Politique de configuration).
6. Pour afficher les comptes sources qui sont liés à ce compte de surveillance, choisissez l'onglet Linked source accounts (Comptes sources liés).
7. Pour afficher l'ARN de puits du compte de surveillance, et les types de données que ce compte de surveillance peut afficher dans les comptes sources liés, choisissez l'onglet Linked source accounts (Comptes sources liés).

Interrogation de métriques d’autres sources de données

Vous pouvez l'utiliser CloudWatch pour interroger, visualiser et créer des alarmes pour les métriques provenant d'autres sources de données. Pour ce faire, vous devez vous connecter CloudWatch aux autres sources de données. Vous bénéficiez ainsi d'une expérience de surveillance unique et consolidée au sein de la CloudWatch console. Vous pouvez bénéficier d’une vue unifiée des indicateurs de votre infrastructure et de vos applications, quel que soit l’endroit où les données sont stockées, ce qui vous permet d’identifier et de résoudre les problèmes plus rapidement.

Après vous être connecté à une source de données à l'aide d'un CloudWatch assistant, il CloudWatch crée une AWS CloudFormation pile qui déploie et configure une AWS Lambda fonction. Cette fonction Lambda s’exécute à la demande chaque fois que vous interrogez la source de données. Le générateur de CloudWatch requêtes affiche en temps réel une liste d'éléments pouvant être interrogés, tels que des métriques, des tables, des champs ou des étiquettes. Lorsque vous faites des choix, le générateur de requêtes préremplit une requête dans la langue native de la source sélectionnée.

CloudWatch fournit des assistants guidés vous permettant de vous connecter aux sources de données suivantes. Pour ces sources de données, vous fournissez des informations de base pour identifier la source de données et les informations d’identification. Vous pouvez également créer manuellement des connecteurs à d’autres sources de données en créant vos propres fonctions Lambda.

- Amazon OpenSearch Service : obtenez des métriques à partir des journaux et des traces de votre OpenSearch service.
- Amazon Managed Service for Prometheus : interrogez ces métriques à l’aide de PromQL.
- Amazon RDS for MySQL : utilisez SQL pour convertir les données stockées dans vos tables Amazon RDS en métriques.
- Amazon RDS for PostgreSQL : utilisez SQL pour convertir les données stockées dans vos tables Amazon RDS en métriques.
- Fichiers CSV Amazon S3 : affiche les données des métriques d’un fichier CSV stocké dans un compartiment Amazon S3.
- Microsoft Azure Monitor : interrogez les métriques depuis votre compte Microsoft Azure Monitor.
- Prometheus : interrogez ces métriques à l’aide de PromQL.

Après avoir créé des connecteurs vers des sources de données, veuillez consulter la rubrique Création d'un graphique de mesures à partir d'une autre source de données pour plus d'informations sur la représentation graphique d'une métrique à partir d'une source de données. Pour plus d'informations sur le réglage d'une alarme sur une métrique provenant d'une source de données, veuillez consulter al rubrique Création d'une alarme basée sur une source de données connectée.

Rubriques

Gestion de l'accès aux sources de données

CloudWatch utilise AWS CloudFormation pour créer les ressources requises dans votre compte. Nous vous recommandons d'utiliser cette `cloudformation:TemplateUrl` condition pour contrôler l'accès aux AWS CloudFormation modèles lorsque vous accordez `CreateStack` des autorisations aux utilisateurs IAM.

> **Warning**
>
> Tout utilisateur auquel vous accordez l'autorisation d'invoquer une source de données peut interroger les métriques de cette source de données même s'il ne dispose pas d'autorisations IAM directes sur la source de données. Par exemple, si vous accordez des autorisations `lambda:InvokeFunction` sur une fonction Lambda de la source de données Amazon Managed Service for Prometheus à un utilisateur, celui-ci pourra interroger les métriques de l'espace de travail Amazon Managed Service for Prometheus correspondant, même si vous ne lui avez pas accordé un accès IAM direct à cet espace de travail.

Vous trouverez des modèles d'URL pour les sources de données sur la page Créer une pile de la console des CloudWatch paramètres.

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

```
  "Statement":[
    {
      "Effect" : "Allow",
      "Action" : [ "cloudformation:CreateStack" ],
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "cloudformation:TemplateUrl" : [ data-source-template-url ]
        }
      }
    }
  ]
}
```

Pour plus d'informations sur le contrôle AWS CloudFormation d'accès, consultez Controlling access with AWS Identity and Access Management

Connexion à une source de données prédéfinie à l’aide d’un assistant

Cette rubrique fournit des instructions relatives à l'utilisation de l'assistant pour se connecter CloudWatch aux sources de données suivantes.

- Amazon OpenSearch Service
- Amazon Managed Service for Prometheus
- Amazon RDS for MySQL
- Amazon RDS for PostgreSQL
- Fichiers CSV Amazon S3
- Moniteur Microsoft Azure
- Prometheus

Plus avant dans cette section, vous trouverez des sous-sections contenant des remarques sur la gestion et l’interrogation de chacune de ces sources de données.

Pour créer un connecteur à une source de données

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

2. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
3. Choisissez l’onglet Sources de données de métriques.
4. Choisissez Create data source.
5. Sélectionnez la source de votre choix, puis choisissez Suivant.
6. Entrez un nom pour la source de données.
7. Saisissez les autres informations requises, en fonction de la source de données que vous avez choisie. Cela peut inclure des informations d’identification pour accéder à la source de données et des informations d’identification de la source de données telles que le nom de l’espace de travail Prometheus, le nom de la base de données ou le nom du compartiment Amazon S3. Pour les AWS services, l'assistant découvre les ressources et les insère dans le menu déroulant de sélection.

   Pour plus de remarques sur la source de données que vous utilisez, veuillez consulter les sections qui suivent cette procédure.
8. Pour vous CloudWatch connecter à la source de données dans un VPC, choisissez Utiliser un VPC et sélectionnez le VPC à utiliser. Sélectionnez ensuite le sous-réseau et le groupe de sécurité.
9. Choisissez Je AWS CloudFormation reconnais pour créer des ressources IAM. Cette ressource est le rôle d’exécution de la fonction Lambda.
10. Choisissez Create data source.

    La nouvelle source que vous venez d'ajouter n'apparaît pas tant que la AWS CloudFormation pile n'a pas fini de la créer. Pour vérifier la progression, vous pouvez choisir Afficher le statut de ma CloudFormation pile. Vous pouvez également choisir l’icône d’actualisation pour mettre à jour cette liste.

    Lorsque votre nouvelle source de données s’affiche dans cette liste, elle est prête à être utilisée. Vous pouvez choisir Requête parmi CloudWatch les métriques pour commencer à l'interroger. Pour plus d’informations, consultez Création d’un graphique de mesures à partir d’une autre source de données.

Amazon Managed Service for Prometheus

Mise à jour de la configuration de la source de données

- Vous pouvez mettre à jour votre source de données manuellement en procédant comme suit :

- Pour mettre à jour l'ID d'espace de travail Amazon Managed Service for Prometheus, mettez à jour la variable d'environnement `AMAZON_PROMETHEUS_WORKSPACE_ID` de la fonction Lambda du connecteur de source de données.
- Pour mettre à jour la configuration du VPC, veuillez consulter Configuration de l'accès au VPC (console) pour plus d'informations.

Interrogation de la source de données

- Lorsque vous interrogez Amazon Managed Service for Prometheus, après avoir sélectionné la source de données dansl'onglet Requête multisources et sélectionné un connecteur Amazon Managed Service for Prometheus, vous pouvez utiliser l'assistant aux requêtes pour découvrir les métriques et les étiquettes et fournir des requêtes PromQL simples. Vous pouvez également utiliser l'éditeur de requêtes PromQL pour créer une requête PromQL.
- Les requêtes multilignes ne sont pas prises en charge par les connecteurs de source de CloudWatch données. Chaque retour à la ligne est remplacé par un espace lorsque la requête est exécutée, ou lorsque vous créez une alarme ou un widget de tableau de bord avec la requête. Dans certains cas, cela peut rendre votre requête non valide. Par exemple, si votre requête contient un commentaire d'une seule ligne, elle ne sera pas valide. Si vous essayez de créer un tableau de bord ou une alarme à l'aide d'une requête multiligne à partir de la ligne de commande ou de l'infrastructure en tant que code, l'API rejettera l'action avec une erreur d'analyse.

Amazon OpenSearch Service

Création d'une source de données

Si le OpenSearch domaine est activé pour le FGAC, vous devez mapper le rôle d'exécution de la fonction Lambda du connecteur à un utilisateur OpenSearch dans Service. Pour plus d'informations, consultez la section Associer les utilisateurs aux rôles dans la section Gestion des autorisations dans la documentation du OpenSearch service.

Si votre OpenSearch domaine n'est accessible que dans un Virtual Private Cloud (VPC), vous devez inclure manuellement une nouvelle variable d'environnement dans la fonction Lambda appelée. `AMAZON_OPENSEARCH_ENDPOINT` La valeur de cette variable doit être le domaine racine du OpenSearch point de terminaison. Vous pouvez obtenir ce domaine racine en supprimant `https://` et `<region>.es.amazonaws.com` depuis le point de terminaison de domaine répertorié dans la console OpenSearch de service. Par exemple, si le point de terminaison de votre

domaine esthttps://sample-domain.us-east-1.es.amazonaws.com, le domaine racine seraitsample-domain.

Mise à jour d’une source de données

- Vous pouvez mettre à jour votre source de données manuellement en procédant comme suit :
  - Pour mettre à jour le domaine OpenSearch de service, mettez à jour la variable d'AMAZON_OPENSEARCH_DOMAIN_NAMEenvironnement de la fonction Lambda du connecteur de source de données.
  - Pour mettre à jour la configuration du VPC, veuillez consulter Configuration de l’accès au VPC (console) pour plus d’informations.

Interrogation de la source de données

- Lorsque vous interrogez OpenSearch Service, après avoir sélectionné la source de données dans l'onglet Requête multi-sources, procédez comme suit :
  - Sélectionnez l’index à interroger.
  - Sélectionnez le nom de la métrique (n’importe quel champ numérique du document) et Stat.
  - Sélectionnez l’axe temporel (n’importe quel champ de date dans le document).
  - Sélectionnez les filtres à appliquer (n’importe quel champ de chaîne du document).
  - Choisissez Requête graphique.

Amazon RDS for PostgreSQL et Amazon RDS for MySQL

Création d’une source de données

- Si votre source de données n’est accessible que dans un VPC, vous devez inclure la configuration VPC du connecteur, comme décrit dans Connexion à une source de données prédéfinie à l’aide d’un assistant. Si la source de données doit se connecter au VPC pour obtenir des informations d’identification, le point de terminaison doit être configuré dans le VPC. Pour plus d'informations, consultez la section Utilisation d'un point de AWS Secrets Manager terminaison VPC.

  En outre, vous devez créer un point de terminaison VPC pour le service Amazon RDS. Pour plus d'informations, consultez l'API Amazon RDS et les points de terminaison AWS PrivateLink VPC d'interface ().

Mise à jour d’une source de données

- Vous pouvez mettre à jour votre source de données manuellement en procédant comme suit :
  - Pour mettre à jour l’instance de base de données, mettez à jour la variable d’environnement `RDS_INSTANCE` de la fonction Lambda du connecteur de source de données.
  - Pour mettre à jour le nom d’utilisateur et le mot de passe utilisés pour se connecter à Amazon RDS, utilisez AWS Secrets Manager. Vous pouvez trouver l’ARN du secret utilisé pour la source de données dans la variable d’environnement `RDS_SECRET` de la fonction Lambda de la source de données. Pour plus d’informations sur la mise à jour du secret dans AWS Secrets Manager, veuillez consulter la rubrique Modifier un secret AWS Secrets Manager.
  - Pour mettre à jour la configuration du VPC, veuillez consulter Configuration de l’accès au VPC (console) pour plus d’informations.

Interrogation de la source de données

- Lorsque vous interrogez Amazon RDS, après avoir sélectionné la source de données dans l’onglet Requête multisources et sélectionné un connecteur Amazon RDS, vous pouvez utiliser le découvreur de base de données pour afficher les bases de données, les tables et les colonnes disponibles. Vous pouvez également utiliser l’éditeur SQL pour créer une requête SQL.

  Vous pouvez utiliser les variables suivantes dans la requête :

  - `$start.iso` : l’heure de début au format de date ISO
  - `$end.iso` : l’heure de fin au format de date ISO
  - `$period` : la période sélectionnée en secondes

  Par exemple, vous pouvez effectuer la requête `SELECT value, timestamp FROM table WHERE timestamp BETWEEN $start.iso and $end.iso`
- Les requêtes multilignes ne sont pas prises en charge par les connecteurs de source de CloudWatch données. Chaque retour à la ligne est remplacé par un espace lorsque la requête est exécutée, ou lorsque vous créez une alarme ou un widget de tableau de bord avec la requête. Dans certains cas, cela peut rendre votre requête non valide. Par exemple, si votre requête contient un commentaire d’une seule ligne, elle ne sera pas valide. Si vous essayez de créer un tableau de bord ou une alarme à l’aide d’une requête multiligne à partir de la ligne de commande ou de l’infrastructure en tant que code, l’API rejettera l’action avec une erreur d’analyse.

> **Note**
>
> Si aucun champ de date n'est trouvé dans les résultats, les valeurs de chaque champ numérique sont additionnées en valeurs uniques et tracées sur l'intervalle de temps spécifié. Si les horodatages ne correspondent pas à la période sélectionnée dans CloudWatch, les données sont automatiquement agrégées en utilisant `SUM` et alignées sur la période en. CloudWatch

Fichiers CSV Amazon S3

Interrogation de la source de données

- Lorsque vous interrogez des fichiers CSV Amazon S3, après avoir sélectionné la source de données dans l'onglet Requête multisources et sélectionné un connecteur Amazon S3, vous sélectionnez le compartiment et la clé Amazon S3.

  Le fichier CSV doit être formaté de la manière suivante :

  - L'horodatage doit être la première colonne.
  - Le tableau doit comporter une ligne d'en-tête. Les en-têtes sont utilisés pour nommer vos indicateurs. Le titre de la colonne d'horodatage sera ignoré, seuls les titres des colonnes de mesures sont utilisés.
  - Les horodatages doivent être au format de date ISO.
  - Les métriques doivent être des champs numériques.

```
Timestamp, Metric-1, Metric-2, ...
```

  Voici un exemple :

timestamp	Processeur (%)	Memory (%) (Mémoire (%))	Stockage (%)
2023-11-23T17:09:41+00:00	1	2	3

timestamp	Processeur (%)	Memory (%) (Mémoire (%))	Stockage (%)
2023-11-23T17:04:41+00:00	4	5	6
2023-11-23T16:59:41+00:00	7	8	9
2023-11-23T16:54:41+00:00	10	11	12

> **Note**
>
> Si aucun horodatage n'est fourni, les valeurs de chaque métrique sont additionnées en valeurs uniques et tracées sur l'intervalle de temps spécifié. Si les horodatages ne correspondent pas à la période sélectionnée dans CloudWatch, les données sont automatiquement agrégées en utilisant `SUM` et alignées sur la période en. CloudWatch

Moniteur Microsoft Azure

Création d'une source de données

- Vous devez fournir votre ID client, et votre secret client pour vous connecter à Microsoft Azure Monitor. Les informations d'identification seront stockées dans AWS Secrets Manager. Pour plus d'informations, veuillez consulter la rubrique Créer une application et un principal de service Microsoft Entra pouvant accéder aux ressources de la documentation Microsoft.

Mise à jour d'une source de données

- Vous pouvez mettre à jour votre source de données manuellement en procédant comme suit :
  - Pour mettre à jour l'ID du locataire, l'ID du client et le secret du client utilisés pour se connecter à Azure Monitor, vous pouvez trouver l'ARN du secret utilisé pour la source de données en tant que variable d'environnement `AZURE_CLIENT_SECRET` sur la fonction Lambda de la source de données. Pour plus d'informations sur la mise à jour du secret dans AWS Secrets Manager, voir Modifier un AWS Secrets Manager secret.

Interrogation de la source de données

- Lorsque vous interrogez Azure Monitor, après avoir sélectionné la source de données dans l'onglet Requête multisources et sélectionné un connecteur Azure Monitor, vous spécifiez l'abonnement Azure, le groupe de ressources et la ressource. Vous pouvez ensuite sélectionner l'espace de noms, la métrique et l'agrégation des métriques, puis les filtrer par dimensions.

Prometheus

Création d'une source de données

- Vous devez fournir le point de terminaison Prometheus ainsi que l'utilisateur et le mot de passe requis pour interroger Prometheus. Les informations d'identification seront stockées dans AWS Secrets Manager.
- Si votre source de données n'est accessible que dans un VPC, vous devez inclure la configuration VPC du connecteur, comme décrit dans Connexion à une source de données prédéfinie à l'aide d'un assistant. Si la source de données doit se connecter pour obtenir des informations d'identification, le point de terminaison doit être configuré dans le VPC. Pour plus d'informations, consultez la section Utilisation d'un point de AWS Secrets Manager terminaison VPC.

Mise à jour de la configuration de la source de données

- Vous pouvez mettre à jour votre source de données manuellement en procédant comme suit :
  - Pour mettre à jour le point de terminaison Prometheus, spécifiez le nouveau point de terminaison comme variable d'environnement `PROMETHEUS_API_ENDPOINT` dans la fonction Lambda de la source de données.
  - Pour mettre à jour le nom d'utilisateur et le mot de passe utilisés pour se connecter à Prometheus, vous pouvez trouver l'ARN du secret utilisé pour la source de données comme variable d'environnement `PROMETHEUS_API_SECRET` dans la fonction Lambda de la source de données. Pour plus d'informations sur la mise à jour du secret dans AWS Secrets Manager, voir Modifier un AWS Secrets Manager secret.
  - Pour mettre à jour la configuration du VPC, veuillez consulter Configuration de l'accès au VPC (console) pour plus d'informations.

Interrogation de la source de données

> **Important**
>
> Les types de métriques Prometheus sont différents des métriques et CloudWatch de nombreuses métriques disponibles via Prometheus sont cumulatives par conception. Lorsque vous interrogez les métriques de Prometheus CloudWatch , aucune transformation supplémentaire n'est appliquée aux données : si vous spécifiez uniquement le nom ou le libellé de la métrique, la valeur affichée sera cumulative. Pour plus d'informations, veuillez consulter la rubrique Metric types dans la documentation Prometheus.
> Pour voir les données des métriques Prometheus sous forme de valeurs discrètes, CloudWatch comme les métriques, vous devez modifier la requête avant de l'exécuter. A titre d'exemple, vous pouvez avoir besoin d'ajouter un appel à la fonction rate sur le nom de votre métrique Prometheus. Pour de la documentation sur la fonction rate et les autres fonctions de Prometheus, veuillez consulter la rubrique rate() dans la documentation de Prometheus.

Les requêtes multilignes ne sont pas prises en charge par les connecteurs de source de CloudWatch données. Chaque retour à la ligne est remplacé par un espace lorsque la requête est exécutée, ou lorsque vous créez une alarme ou un widget de tableau de bord avec la requête. Dans certains cas, cela peut rendre votre requête non valide. Par exemple, si votre requête contient un commentaire d'une seule ligne, elle ne sera pas valide. Si vous essayez de créer un tableau de bord ou une alarme à l'aide d'une requête multiligne à partir de la ligne de commande ou de l'infrastructure en tant que code, l'API rejettera l'action avec une erreur d'analyse.

Notification des mises à jour disponibles

De temps à autre, Amazon peut vous informer que nous vous recommandons de mettre à jour vos connecteurs avec une version plus récente disponible et vous fournira des instructions sur la manière de procéder.

Création d'un connecteur personnalisé à une source de données

Pour connecter une source de données personnalisée à CloudWatch, deux options s'offrent à vous :

- Commencez par utiliser un exemple de modèle qui CloudWatch fournit. Vous pouvez utiliser l'un JavaScript ou l'autre ou Python avec ce modèle. Ces modèles incluent un exemple de code Lambda qui vous sera utile lors de la création de votre fonction Lambda. Vous pouvez ensuite modifier la fonction Lambda à partir du modèle pour vous connecter à votre source de données personnalisée.

- Créez une AWS Lambda fonction à partir de zéro qui implémente le connecteur de source de données, la requête de données et la préparation des séries chronologiques à utiliser par CloudWatch. Cette fonction doit pré-agréger ou fusionner des points de données si nécessaire, et également aligner la période et les horodatages pour être compatible avec. CloudWatch

Table des matières

Utilisation d’un modèle

L’utilisation d’un modèle crée un exemple de fonction Lambda et peut vous aider à créer votre connecteur personnalisé plus rapidement. Ces exemples de fonctions fournissent des exemples de code pour de nombreux scénarios courants liés à la création d’un connecteur personnalisé. Vous pouvez examiner le code Lambda après avoir créé un connecteur avec un modèle, puis le modifier pour l’utiliser pour vous connecter à votre source de données.

De plus, si vous utilisez le modèle, CloudWatch prend soin de créer la politique d'autorisations Lambda et d'associer des balises de ressources à la fonction Lambda.

Pour utiliser le modèle afin de créer un connecteur vers une source de données personnalisée

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
3. Choisissez l’onglet Sources de données de métriques.
4. Choisissez Create data source.

5. Choisissez le bouton radio Personnalisé – modèle de démarrage, puis Choisissez Suivant.
6. Entrez un nom pour la source de données.
7. Sélectionnez l'un des modèles répertoriés.
8. Sélectionnez Node.js ou Python.
9. Choisissez Create data source.

   La nouvelle source personnalisée que vous venez d'ajouter n'apparaît pas tant que la AWS CloudFormation pile n'a pas fini de la créer. Pour vérifier la progression, vous pouvez choisir Afficher le statut de ma CloudFormation pile. Vous pouvez également choisir l'icône d'actualisation pour mettre à jour cette liste.

   Lorsque votre nouvelle source de données apparaît dans cette liste, elle est prête à être testée dans la console et modifiée.
10. (Facultatif) Pour interroger les données de test de cette source dans la console, suivez les instructions de la rubrique Création d'un graphique de mesures à partir d'une autre source de données.
11. Modifiez la fonction Lambda en fonction de vos besoins.

    a. Dans le panneau de navigation, sélectionnez Settings (Paramètres).

    b. Choisissez l'onglet Sources de données de métriques.

    c. Choisissez Afficher dans la console Lambda pour la source que vous souhaitez modifier.

    Vous pouvez désormais modifier la fonction pour accéder à votre source de données. Pour plus d'informations, consultez Étape 1 : créer la fonction.

    > **Note**
    >
    > En utilisant le modèle, lorsque vous écrivez votre fonction Lambda, vous n'avez pas besoin de suivre les instructions contenues dans Étape 2 : créer une stratégie d'autorisations Lambda ou. Étape 3 : attacher une balise de ressource à la fonction Lambda Ces étapes ont été effectuées CloudWatch parce que vous avez utilisé le modèle.

Création d’une source de données personnalisée de toutes pièces

Suivez les étapes décrites dans cette section pour créer une fonction Lambda qui se connecte CloudWatch à une source de données.

Étape 1 : créer la fonction

Un connecteur de source de données personnalisé doit prendre en charge `GetMetricData` les événements provenant de CloudWatch. Vous pouvez également éventuellement implémenter un `DescribeGetMetricData` événement pour fournir aux utilisateurs de la CloudWatch console une documentation expliquant comment utiliser le connecteur. La `DescribeGetMetricData` réponse peut également être utilisée pour définir les valeurs par défaut utilisées dans le générateur de requêtes CloudWatch personnalisé.

CloudWatch fournit des extraits de code sous forme d'exemples pour vous aider à démarrer. Pour plus d'informations, consultez le référentiel d'échantillons à l'[adresse https://github.com/aws-samples/cloudwatch-data-source-samples](https://github.com/aws-samples/cloudwatch-data-source-samples).

Contraintes

- La réponse de Lambda doit être inférieure à 6 Mo. Si la réponse dépasse 6 Mo, la réponse `GetMetricData` marque la fonction Lambda comme `InternalError` et aucune donnée n’est renvoyée.
- La fonction Lambda doit être exécutée dans les 10 secondes à des fins de visualisation et de tableau de bord, ou dans les 4,5 secondes pour l’utilisation des alarmes. Si le temps d’exécution dépasse ce délai, la réponse `GetMetricData` marque la fonction Lambda comme `InternalError` et aucune donnée n’est renvoyée.
- La fonction Lambda doit envoyer sa sortie en utilisant des horodatages d’époque en secondes.
- Si la fonction Lambda ne rééchantillonne pas les données mais renvoie des données qui ne correspondent pas à l'heure de début et à la durée de la période demandées par l' CloudWatch utilisateur, ces données sont ignorées par. CloudWatch Les données supplémentaires sont supprimées de toute visualisation ou alarme. Toutes les données qui ne se situent pas entre l’heure de début et de fin sont également supprimées.

  Par exemple, si un utilisateur demande des données entre 10 h et 11 h avec une période de 5 minutes, les intervalles de temps « 10:00:00 à 10:04:59 » et « 10:05:00 à 10:09:59 » sont les intervalles de temps valides pour le renvoi des données. Vous devez renvoyer une série chronologique qui inclut `10:00 value1`, `10:05 value2`, etc. Si la fonction renvoie `10:03`

`valueX`, par exemple, elle est supprimée, car 10:03 ne correspond pas à l'heure de début et à la période demandées.

- Les requêtes multilignes ne sont pas prises en charge par les connecteurs de source de CloudWatch données. Chaque retour à la ligne est remplacé par un espace lorsque la requête est exécutée, ou lorsque vous créez une alarme ou un widget de tableau de bord avec la requête. Dans certains cas, cela peut rendre votre requête non valide.

GetMetricData événement

Charge utile de la requête

Voici un exemple de charge utile de la requête `GetMetricData` envoyée en entrée à la fonction Lambda.

```
{
  "EventType": "GetMetricData",
  "GetMetricDataRequest": {
    "StartTime": 1697060700,
    "EndTime": 1697061600,
    "Period": 300,
    "Arguments": ["serviceregistry_external_http_requests{host_cluster!=\"prod\"}"]
  }
}
```

- StartTime— L'horodatage spécifiant les premières données à renvoyer. Le type est horodatage, époque, secondes.
- EndTime— L'horodatage spécifiant les dernières données à renvoyer. Le type est horodatage, époque, secondes.
- Période : nombre de secondes que représente chaque agrégation des données de métriques. La valeur minimale est 60 secondes. Le type est secondes.
- Arguments : tableau d'arguments à transmettre à l'expression mathématique de la métrique Lambda. Pour plus d'informations sur le passage d'arguments, veuillez consulter la rubrique Comment transmettre des arguments à votre fonction Lambda.

Charge utile de la réponse

Voici un exemple de charge utile de la réponse `GetMetricData` renvoyée par la fonction Lambda.

```
{
   "MetricDataResults": [
      {
         "StatusCode": "Complete",
         "Label": "CPUUtilization",
         "Timestamps": [ 1697060700, 1697061000, 1697061300 ],
         "Values": [ 15000, 14000, 16000 ]
      }
   ]
}
```

La charge utile de la réponse contiendra soit un champ `MetricDataResults`, soit un champ `Error`, mais pas les deux.

Un champ `MetricDataResults` est une liste de champs de séries temporelles de type `MetricDataResult`. Chacun de ces champs de série temporelle peut comprendre les champs suivants.

- StatusCode— (Facultatif) `Complete` indique que tous les points de données de la plage de temps demandée ont été renvoyés. `PartialData`signifie qu'un ensemble incomplet de points de données a été renvoyé. Si cet argument est omis, la valeur par défaut est `Complete`.

  Valeurs Valides: `Complete` | `InternalError` | `PartialData` | `Forbidden`
- Messages : liste facultative de messages contenant des informations supplémentaires sur les données renvoyées.

  Type : tableau d'MessageDataobjets avec des `Value` chaînes `Code` et.
- Label : étiquette lisible par l'homme associée aux données.

  Type : chaîne
- Timestamps : horodatages des points de données, formatés en fonction de l'époque. Le nombre d'horodatages correspond toujours au nombre de valeurs et la valeur de `Timestamps[x]` est `Values[x]`.

  Type : tableau d'horodatage
- Values : valeurs des points de données de la métrique, correspondant à `Timestamps`. Le nombre de valeurs correspond toujours au nombre d'horodatages et la valeur de `Timestamps[x]` est `Values[x]`.

Type : tableau de doubles

Pour en savoir plus sur les objets `Error`, veuillez consulter les sections suivantes.

Formats de réponse aux erreurs

Vous pouvez éventuellement utiliser la réponse d'erreur pour fournir plus d'informations sur les erreurs. Nous vous recommandons de renvoyer une erreur lors de la validation du code lorsqu'une erreur de validation se produit, par exemple lorsqu'un paramètre est manquant ou de type incorrect.

Voici un exemple de réponse lorsque la fonction Lambda souhaite déclencher une exception de validation `GetMetricData`.

```
{
   "Error": {
      "Code": "Validation",
      "Value": "Invalid Prometheus cluster"
   }
}
```

Voici un exemple de réponse lorsque la fonction Lambda indique qu'elle n'est pas en mesure de renvoyer des données en raison d'un problème d'accès. La réponse est traduite en une seule série chronologique avec un code d'état `Forbidden`.

```
{
   "Error": {
      "Code": "Forbidden",
      "Value": "Unable to access ..."
   }
}
```

Voici un exemple de cas où la fonction Lambda déclenche une exception globale `InternalError`, qui est traduite en une série temporelle unique avec un code d'état `InternalError` et un message. Chaque fois qu'un code d'erreur a une valeur autre que `Validation` ou`Forbidden`, CloudWatch suppose qu'il s'agit d'une erreur interne générique.

```
{
   "Error": {
      "Code": "PrometheusClusterUnreachable",
      "Value": "Unable to communicate with the cluster"
```

```
    }
}
```

DescribeGetMetricData événement

Charge utile de la requête

L'exemple qui suit illustre la charge utile d'une requête `DescribeGetMetricData`.

```
{
  "EventType": "DescribeGetMetricData"
}
```

Charge utile de la réponse

L'exemple qui suit illustre la charge utile d'une réponse `DescribeGetMetricData`.

```
{
    "Description": "Data source connector",
    "ArgumentDefaults": [{
        Value: "default value"
     }]
}
```

- Description : description de l'utilisation du connecteur de source de données. Cette description apparaîtra dans la CloudWatch console. Markdown est pris en charge.

  Type : chaîne
- ArgumentDefaults— Le tableau facultatif des valeurs par défaut des arguments utilisés préremplit le générateur de sources de données personnalisé.

  Si`[{ Value: "default value 1"}, { Value: 10}]`, est renvoyé, le générateur de requêtes de la CloudWatch console affiche deux entrées, la première avec la « valeur par défaut 1 » et la seconde avec 10.

  Si `ArgumentDefaults` n'est pas fourni, une seule entrée est affichée avec le type par défaut défini sur `String`.

  Type : tableau d'objets contenant Value et Type.
- Error : (facultatif) un champ d'erreur peut être inclus dans n'importe quelle réponse. Vous pouvez voir des exemples dans GetMetricData événement.

Considérations importantes relatives aux CloudWatch alarmes

Si vous comptez utiliser la source de données pour définir des CloudWatch alarmes, vous devez la configurer pour qu'elle rapporte les données avec des horodatages toutes les minutes à. CloudWatch Pour plus d'informations et d'autres considérations relatives à la création d'alarmes sur des métriques provenant de sources de données connectées, veuillez consulter la rubrique Création d'une alarme basée sur une source de données connectée.

(Facultatif) AWS Secrets Manager À utiliser pour stocker les informations d'identification

Si votre fonction Lambda doit utiliser des informations d'identification pour accéder à la source de données, nous vous recommandons de les stocker AWS Secrets Manager au lieu de les coder en dur dans votre fonction Lambda. Pour plus d'informations sur l'utilisation AWS Secrets Manager avec Lambda, voir Utiliser des AWS Secrets Manager secrets dans AWS Lambda les fonctions.

(Facultatif) Connexion à une source de données dans un VPC

Si votre source de données se trouve dans un VPC géré par Amazon Virtual Private Cloud, vous devez configurer votre fonction Lambda pour y accéder. Pour plus d'informations, veuillez consulter la rubrique Connexion des réseaux sortants aux ressources d'un VPC.

Vous devrez peut-être également configurer les points de terminaison du service VPC pour accéder à des services tels qu' AWS Secrets Manager. Pour plus d'informations, consultez Accéder à un AWS service à l'aide d'un point de terminaison VPC d'interface.

Étape 2 : créer une stratégie d'autorisations Lambda

Vous devez utiliser create une déclaration de politique qui CloudWatch autorise l'utilisation de la fonction Lambda que vous avez créée. Vous pouvez utiliser la console AWS CLI ou la console Lambda pour créer la déclaration de politique.

Pour utiliser le AWS CLI pour créer la déclaration de politique

- Entrez la commande suivante. Remplacez *123456789012* par votre identifiant de compte, remplacez par le nom *my-data-source-function*de votre fonction Lambda et remplacez *MyDataSource- DataSourcePermission* 1234 par une valeur unique arbitraire.

```
aws lambda add-permission --function-name my-data-source-function --statement-
id MyDataSource-DataSourcePermission1234 --action lambda:InvokeFunction --principal
 lambda.datasource.cloudwatch.amazonaws.com --source-account 123456789012
```

Étape 3 : attacher une balise de ressource à la fonction Lambda

La CloudWatch console détermine quelles fonctions Lambda sont des connecteurs de source de données à l'aide d'une balise. Lorsque vous créez une source de données à l'aide de l'un des assistants, la balise est automatiquement appliquée par la AWS CloudFormation pile qui la configure. Lorsque vous créez vous-même une source de données, vous pouvez utiliser la balise suivante pour votre fonction Lambda. Cela fait apparaître votre connecteur dans la liste déroulante des sources de données de la CloudWatch console lorsque vous interrogez des métriques.

- Une étiquette avec la clé `cloudwatch:datasource` et la valeur `custom`.

Utilisation de votre source de données personnalisée

Après avoir créé une source de données, vous pouvez l'utiliser pour interroger les données de cette source afin de les visualiser et de définir des alarmes. Si vous avez utilisé le modèle pour créer votre connecteur de source de données personnalisé ou si vous avez ajouté la balise répertoriée dans Étape 3 : attacher une balise de ressource à la fonction Lambda, vous pouvez suivre les étapes décrites dans Création d'un graphique de mesures à partir d'une autre source de données pour l'interroger.

Vous pouvez également utiliser la fonction mathématique de métrique `LAMBDA` pour l'interroger, comme expliqué dans la section suivante.

Pour plus d'informations sur la création d'alarmes sur les métriques de votre source de données, veuillez consulter la rubrique Création d'une alarme basée sur une source de données connectée.

Comment transmettre des arguments à votre fonction Lambda

La méthode recommandée pour transmettre des arguments à votre source de données personnalisée consiste à utiliser le générateur de requêtes de la CloudWatch console lorsque vous interrogez la source de données.

Vous pouvez également utiliser votre fonction Lambda pour récupérer des données de votre source de données en utilisant la nouvelle `LAMBDA` expression en mathématiques CloudWatch métriques.

```
LAMBDA("LambdaFunctionName" [, optional-arg]*)
```

`optional-arg` contient jusqu'à 20 chaînes, nombres ou booléens. Par exemple, `param`, `3.14` ou `true`.

> **Note**
>
> Les chaînes multilignes ne sont pas prises en charge par les connecteurs de source de CloudWatch données. Chaque retour à la ligne est remplacé par un espace lorsque la requête est exécutée, ou lorsque vous créez une alarme ou un widget de tableau de bord avec la requête. Dans certains cas, cela peut rendre votre requête non valide.

Lorsque vous utilisez la fonction mathématique de métrique LAMBDA, vous pouvez fournir le nom de la fonction (`"MyFunction"`). Si votre politique de ressources le permet, vous pouvez également utiliser une version spécifique de la fonction (`"MyFunction:22"`) ou un alias de fonction Lambda (`"MyFunction:MyAlias"`). Vous ne pouvez pas utiliser *

Vous trouverez ci-dessous quelques exemples d'appel de la fonction LAMBDA.

```
LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query")
```

```
LAMBDA("MyCustomDataSource", true, "fuzzy", 99.9)
```

La fonction mathématique de métrique LAMBDA renvoie une liste de séries temporelles qui peuvent être renvoyées au demandeur ou combinées avec d'autres fonctions mathématiques de métriques. Voici un exemple de combinaison de LAMBDA avec d'autres fonctions mathématiques de métriques.

```
FILL(LAMBDA("AmazonOpenSearchDataSource", "MyDomain", "some-query"), 0)
```

Suppression d'un connecteur à une source de données

Pour supprimer un connecteur à une source de données, suivez les instructions de cette section.

Pour supprimer un connecteur à une source de données

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
3. Choisissez l'onglet Sources de données de métriques.
4. Choisissez Gérer CloudFormation dans la ligne de la source de données que vous souhaitez supprimer.

Vous êtes redirigé vers la AWS CloudFormation console.

5. Dans la section portant le nom de votre source de données, choisissez Supprimer.
6. Dans la fenêtre de confirmation, choisissez Supprimer.

Collectez des métriques, des journaux et des traces avec l' CloudWatch agent

L' CloudWatch agent unifié vous permet d'effectuer les opérations suivantes :

- Collecter des métriques au niveau interne du système à partir d'instances Amazon EC2 entre systèmes d'exploitation. Les métriques peuvent inclure des métriques invitées, en plus des métriques pour les instances EC2. Les métriques supplémentaires qui peuvent être collectées sont répertoriées dans Métriques collectées par l' CloudWatchagent.
- Collecte des métriques au niveau du système à partir de serveurs sur site. Il peut s'agir de serveurs dans un environnement hybride ainsi que de serveurs non gérés par AWS.
- Récupérez des mesures personnalisées à partir de vos applications ou services à l'aide des protocoles `StatsD` et `collectd`. `StatsD` est pris en charge à la fois sur les serveurs Linux et les serveurs exécutant Windows Server. `collectd` est pris en charge uniquement sur les serveurs Linux.
- Collecter des journaux à partir d'instances Amazon EC2 et des serveurs sur site, exécutant Linux ou Windows Server.

  > **Note**
  >
  > L' CloudWatch agent ne prend pas en charge la collecte de journaux à partir de tubes FIFO.

- Les versions 1.300031.0 et ultérieures peuvent être utilisées pour activer CloudWatch les signaux d'application. Pour plus d’informations, consultez Application Signals.
- Les versions 1.300025.0 et ultérieures peuvent collecter des traces à partir des SDK des clients OpenTelemetryX-Ray et les envoyer à X-Ray.

  L'utilisation de l' CloudWatch agent vous permet de collecter des traces sans avoir à exécuter un démon de collecte de traces distinct, ce qui contribue à réduire le nombre d'agents que vous exécutez et gérez.

Vous pouvez stocker et consulter les métriques que vous collectez avec l' CloudWatch agent CloudWatch comme vous le pouvez avec n'importe quelle autre CloudWatch métrique. L'espace de

noms par défaut pour les métriques collectées par l' CloudWatch agent est`CWAgent`, bien que vous puissiez spécifier un espace de noms différent lorsque vous configurez l'agent.

Les journaux collectés par l' CloudWatch agent unifié sont traités et stockés dans Amazon CloudWatch Logs, tout comme les journaux collectés par l'ancien agent CloudWatch Logs. Pour plus d'informations sur la tarification des CloudWatch journaux, consultez Amazon CloudWatch Pricing.

Les métriques collectées par l' CloudWatch agent sont facturées comme des métriques personnalisées. Pour plus d'informations sur la tarification des CloudWatch métriques, consultez Amazon CloudWatch Pricing.

L' CloudWatch agent est open source sous licence MIT et est hébergé sur GitHub. Si vous souhaitez créer, personnaliser ou contribuer à l' CloudWatch agent, consultez le GitHub référentiel pour obtenir les dernières instructions. Si vous pensez avoir découvert un problème de sécurité potentiel, ne le publiez pas sur GitHub un forum public. Suivez plutôt les instructions de la section Signalement des vulnérabilités ou de la section relative à AWS la sécurité des e-mails directement.

Les étapes décrites dans cette section expliquent comment installer l' CloudWatch agent unifié sur les instances Amazon EC2 et les serveurs sur site. Pour plus d'informations sur les métriques que l' CloudWatch agent peut collecter, consultezMétriques collectées par l' CloudWatchagent.

Systèmes d'exploitation pris en charge

L' CloudWatch agent est pris en charge sur l'architecture x86-64 sur les systèmes d'exploitation suivants. Il est également pris en charge sur toutes les mises à jour de versions mineures pour chacune des versions principales répertoriées ici.

- Amazon Linux 2023
- Amazon Linux 2
- Versions 23.10, 22.04, 20.04, 18.04, 16.04 et 14.04 du serveur Ubuntu
- CentOS versions 9, 8 et 7
- Red Hat Enterprise Linux (RHEL) versions 9, 8 et 7
- Versions 12, 11 et 10 de Debian
- SUSE Linux Enterprise Server (SLES) versions 15 et 12
- Versions 9, 8 et 7 d'Oracle Linux
- AlmaLinux versions 9 et 8

- Rocky Linux versions 9 et 8
- Les ordinateurs macOS suivants : les instances Mac1 EC2 M1 et les ordinateurs exécutant macOS 14 (Sonoma), macOS 13 (Ventura) et macOS 12 (Monterey)
- Versions 64 bits de Windows Server 2022, Windows Server 2019 et Windows Server 2016
- Windows 10 64 bits

L'agent est pris en charge sur l'architecture d'ARM64 sur les systèmes d'exploitation suivants. Il est également pris en charge sur toutes les mises à jour de versions mineures pour chacune des versions principales répertoriées ici.

- Amazon Linux 2023
- Amazon Linux 2
- Versions 23.10, 22.04, 20.04, 18.04 et 16.04 du serveur Ubuntu
- CentOS versions 9 et 8
- Red Hat Enterprise Linux (RHEL) versions 9, 8 et 7
- Versions 12, 11 et 10 de Debian
- SUSE Linux Enterprise Server 15
- Les ordinateurs macOS suivants : macOS 14 (Sonoma), macOS 13 (Ventura) et macOS 12 (Monterey)

Présentation du processus d'installation

Vous pouvez télécharger et installer l' CloudWatch agent manuellement à l'aide de la ligne de commande, ou vous pouvez l'intégrer à SSM. Le déroulement général de l'installation de l' CloudWatch agent à l'aide de l'une ou l'autre méthode est le suivant :

1. Créez des rôles ou des utilisateurs IAM qui permettent à l'agent de collecter des métriques à partir du serveur et éventuellement de les AWS Systems Manager intégrer.
2. Téléchargez le package d'agent.
3. Modifiez le fichier de configuration de l' CloudWatch agent et spécifiez les métriques que vous souhaitez collecter.
4. Installez et lancez l'agent sur vos serveurs. À mesure que vous installez l'agent sur une instance EC2, vous attachez le rôle IAM que vous avez créé à l'étape 1. À mesure que vous installez l'agent

sur un serveur sur site, vous spécifiez un profil nommé qui contient les informations d'identification de l'utilisateur IAM que vous avez créé à l'étape 1.

Table des matières

Installation de l' CloudWatch agent

L' CloudWatch agent est disponible sous forme de package dans Amazon Linux 2023 et Amazon Linux 2. Si vous utilisez l'un de ces systèmes d'exploitation, vous pouvez installer le package en saisissant la commande suivante. Vous devez également vous assurer que le rôle IAM attaché à l'instance possède le rôle CloudWatchAgentServerPolicyattaché. Pour plus d'informations, consultez Créez des rôles IAM à utiliser avec l' CloudWatch agent sur les instances Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Sur tous les systèmes d'exploitation pris en charge, y compris Linux et Windows Server, vous pouvez télécharger et installer l' CloudWatch agent en utilisant la ligne de commande avec un lien de téléchargement Amazon S3, en utilisant Amazon EC2 Systems Manager ou en utilisant AWS CloudFormation un modèle. Consultez les sections suivantes pour obtenir des détails.

Table des matières

Installation de l' CloudWatch agent à l'aide de la ligne de commande

Utilisez les rubriques suivantes pour télécharger, configurer et installer le package de l' CloudWatch agent.

Rubriques

Téléchargez et configurez l' CloudWatchagent à l'aide de la ligne de commande

Procédez comme suit pour télécharger le package de l' CloudWatch agent, créer des rôles ou des utilisateurs IAM et éventuellement modifier le fichier de configuration commun.

Téléchargez le package de CloudWatch l'agent

> **Note**
>
> Pour télécharger l' CloudWatch agent, votre connexion doit utiliser le protocole TLS 1.2 ou version ultérieure.

L' CloudWatch agent est disponible sous forme de package dans Amazon Linux 2023 et Amazon Linux 2. Si vous utilisez ce système d'exploitation, vous pouvez installer le package en entrant la commande suivante. Vous devez également vous assurer que le rôle IAM attaché à l'instance possède le rôle CloudWatchAgentServerPolicyattaché. Pour plus d'informations, consultez Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

```
sudo yum install amazon-cloudwatch-agent
```

Sur tous les systèmes d'exploitation pris en charge, vous pouvez télécharger et installer l' CloudWatch agent à l'aide de la ligne de commande.

Pour chaque lien de téléchargement, vous pouvez trouver un lien général, ainsi que des liens pour chaque Région. Par exemple, pour Amazon Linux 2023, Amazon Linux 2 et l'architecture x86-64, les trois liens de téléchargement valides sont les suivants :

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Vous pouvez également télécharger un fichier README sur les dernières modifications apportées à l'agent, ainsi qu'un fichier qui indique le numéro de version disponible pour le téléchargement. Ces fichiers se trouvent aux emplacements suivants :

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	Amazon Linux 2023 et Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux /amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/ amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/	https://amazoncloudwatch-agent.s3.amazonaws.com/

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
		centos/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaw s.com/centos/amd64 /latest/* .rpm amazon-cloudwatch-agent	centos/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaw s.com/centos/amd64/ latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/ redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaw s.com/redhat/amd64 /latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ redhat/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaw s.com/redhat/amd64/ latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/ suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaw s.com/suse/amd64/ latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ suse/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaw s.com/suse/amd64/l atest/* .rpm.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/debian/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/debian/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/oracle_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/oracle_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/amd64/latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/amd64/latest/* .pkg.sig amazon-cloudwatch-agent
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/windows/amd64/latest/* .msi amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/windows/amd64/latest/* .msi.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
ARM64	Amazon Linux 2023 et Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/arm64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/arm64/latest/* .deb.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	MacOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/arm64/latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/arm64/latest/ amazon-cloudwatch-agent .pkg .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/arm64/latest/* .pkg.sig amazon-cloudwatch-agent

Pour télécharger et installer le package de l' CloudWatch agent à l'aide de la ligne de commande

1. Téléchargez l' CloudWatch agent.

   Sur un serveur Linux, saisissez ce qui suit. Pour *lien de téléchargement*, utilisez le lien de téléchargement approprié du tableau précédent.

   ```
   wget download-link
   ```

   Sur un serveur exécutant Windows Server, téléchargez le fichier suivant :

```
https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
```

2. Une fois que vous avez téléchargé le package, vous pouvez, si vous le souhaitez, vérifier la signature du package. Pour de plus amples informations, consultez Vérification de la signature du package de l' CloudWatch agent.
3. Installez le package . Si vous avez téléchargé un package RPM sur un serveur Linux, accédez au répertoire contenant le package et saisissez ce qui suit :

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

   Si vous avez téléchargé un package DEB sur un serveur Linux, accédez au répertoire contenant le package et saisissez ce qui suit :

   ```
   sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
   ```

   Si vous avez téléchargé un package MSI sur un serveur exécutant Windows Server, accédez au répertoire contenant le package et saisissez ce qui suit :

   ```
   msiexec /i amazon-cloudwatch-agent.msi
   ```

   Cette commande fonctionne également de l'intérieur PowerShell. Pour plus d'informations sur les options de ligne de commande MSI, consultez Command-Line Options (Options de ligne de commande) dans la documentation Microsoft Windows.

   Si vous avez téléchargé un package PKG sur un serveur macOS, accédez au répertoire contenant le package et saisissez ce qui suit :

   ```
   sudo installer -pkg ./amazon-cloudwatch-agent.pkg -target /
   ```

Créer et modifier le fichier de configuration d'agent

Après avoir téléchargé l' CloudWatch agent, vous devez créer le fichier de configuration avant de démarrer l'agent sur n'importe quel serveur. Pour plus d’informations, consultez Création du fichier de configuration de CloudWatch l'agent.

Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch

L'accès aux AWS ressources nécessite des autorisations. Vous créez un rôle IAM, un utilisateur IAM, ou les deux pour accorder les autorisations dont l' CloudWatch agent a besoin pour écrire des métriques. CloudWatch Si vous allez utiliser l'agent sur des instances Amazon EC2, vous devez créer un rôle IAM. Si vous allez utiliser l'agent sur des serveurs sur site, vous devez créer un utilisateur IAM.

> **Note**
>
> Nous avons récemment modifié les procédures suivantes à l'aide des nouvelles politiques `CloudWatchAgentServerPolicy` et `CloudWatchAgentAdminPolicy` créées par Amazon, plutôt que d'exiger des clients qu'ils créent ces stratégies eux-mêmes. Pour écrire des fichiers et télécharger des fichiers depuis le Parameter Store, les stratégies créées par Amazon prennent en charge uniquement les fichiers dont le nom commence par `AmazonCloudWatch-`. Si vous disposez d'un fichier de configuration d' CloudWatch agent dont le nom ne commence pas par`AmazonCloudWatch-`, ces règles ne peuvent pas être utilisées pour écrire le fichier dans Parameter Store ou le télécharger depuis Parameter Store.

Si vous souhaitez exécuter l' CloudWatch agent sur des instances Amazon EC2, suivez les étapes ci-dessous pour créer le rôle IAM nécessaire. Ce rôle fournit des autorisations pour lire les informations de l'instance et les y écrire CloudWatch.

Pour créer le rôle IAM nécessaire pour exécuter l' CloudWatch agent sur les instances EC2

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation gauche, choisissez Roles (Rôles), puis Create role (Créer un rôle).
3. Vérifiez que AWS service (Service) est sélectionné sous Trusted entity type (Type d'entité de confiance).
4. Pour Use case (Un cas d'utilisation), choisissez EC2 sous Common use cases (Cas d'utilisation courants),
5. Choisissez Suivant.

6. Dans la liste des politiques, cochez la case située à côté de CloudWatchAgentServerPolicy. Si nécessaire, utilisez la zone de recherche pour trouver la politique.
7. (Facultatif) Si l'agent envoie des traces à X-Ray, vous devez également attribuer la AWSXRayDaemonWriteAccesspolitique au rôle. Pour ce faire, recherchez cette politique dans la liste et cochez la case à côté.
8. Choisissez Suivant.
9. Dans Nom du rôle, entrez le nom du rôle, tel que *CloudWatchAgentServerRole*. Vous pouvez également lui donner une description. Puis choisissez Create role (Créer un rôle).

   Le rôle est maintenant créé.
10. (Facultatif) Si l'agent doit envoyer des CloudWatch journaux à Logs et que vous souhaitez qu'il soit en mesure de définir des politiques de conservation pour ces groupes de journaux, vous devez ajouter l'`logs:PutRetentionPolicy`autorisation au rôle. Pour plus d'informations, consultez Autoriser l' CloudWatch agent à définir une politique de conservation des journaux.

Si vous comptez exécuter l' CloudWatch agent sur des serveurs locaux, suivez les étapes ci-dessous pour créer l'utilisateur IAM nécessaire.

> **Warning**
>
> Ce scénario nécessite que les utilisateurs IAM disposent d'un accès programmatique et d'informations d'identification à long terme, ce qui présente un risque de sécurité. Pour atténuer ce risque, nous vous recommandons de ne fournir à ces utilisateurs que les autorisations dont ils ont besoin pour effectuer la tâche et de supprimer ces utilisateurs lorsqu'ils ne sont plus nécessaires. Les clés d'accès peuvent être mises à jour si nécessaire. Pour plus d'informations, consultez la section Mise à jour des clés d'accès dans le guide de l'utilisateur IAM.

Pour créer l'utilisateur IAM nécessaire à l'exécution de l' CloudWatch agent sur des serveurs locaux

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, sélectionnez Users (Utilisateurs), puis Add users (Ajouter des utilisateurs).
3. Saisissez le nom d'utilisateur du nouvel utilisateur.

4. Sélectionnez Access key - Programmatic access (Clé d'accès - Accès programmatique), puis choisissez Next: Permissions (Suivant : Autorisations).
5. Choisissez Attach existing policies directly (Attacher directement les politiques existantes).
6. Dans la liste des politiques, cochez la case située à côté de CloudWatchAgentServerPolicy. Si nécessaire, utilisez la zone de recherche pour trouver la politique.
7. (Facultatif) Si l'agent doit effectuer un suivi jusqu'à X-Ray, vous devez également attribuer la AWSXRayDaemonWriteAccesspolitique au rôle. Pour ce faire, recherchez cette politique dans la liste et cochez la case à côté.
8. Choisissez Suivant : Balises.
9. Vous pouvez éventuellement créer des identifications pour le nouvel utilisateur IAM, puis choisissez Next: Review (Suivant : Vérification).
10. Confirmez que la politique affichée est correcte et sélectionnez Create user (Créer un utilisateur).
11. En regard du nom du nouvel utilisateur, choisissez Show (Afficher). Copiez la clé d'accès et la clé secrète dans un fichier afin de pouvoir les utiliser lors de l'installation de l'agent. Choisissez Close (Fermer).

Autoriser l' CloudWatch agent à définir une politique de conservation des journaux

Vous pouvez configurer l' CloudWatch agent pour définir la politique de rétention pour les groupes de journaux auxquels il envoie des événements de journal. Si vous faites cela, vous devez accorder `logs:PutRetentionPolicy` au rôle IAM ou à l'utilisateur que l'agent utilise. L'agent utilise un rôle IAM pour s'exécuter sur des instances Amazon EC2 et utilise un utilisateur IAM pour les serveurs sur site.

Pour accorder au rôle IAM de l' CloudWatch agent l'autorisation de définir des politiques de conservation des journaux

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation de gauche, choisissez Rôles.
3. Dans le champ de recherche, tapez le début du nom du rôle IAM de l' CloudWatch agent. Vous avez choisi ce nom lorsque vous avez créé le rôle. Il pourrait être nommé `CloudWatchAgentServerRole`.

   Lorsque vous voyez le rôle, choisissez le nom du rôle.

4. Sous l'onglet Permissions (Autorisations), sélectionnez Add permissions (Ajouter des autorisations), Create inline policy (Créer une politique en ligne).
5. Cliquez sur l'onglet JSON et copiez la politique suivante dans la zone, en remplaçant le JSON par défaut dans la zone :

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

6. Choisissez Examiner une politique.
7. Pour Name (Nom), saisissez **`CloudWatchAgentPutLogsRetention`** ou un nom similaire, et choisissez Create Policy (Créer une politique).

Pour autoriser l'utilisateur IAM de l' CloudWatch agent à définir des politiques de conservation des journaux

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le volet de navigation de gauche, choisissez Utilisateurs.
3. Dans le champ de recherche, tapez le début du nom de l'utilisateur IAM de l' CloudWatch agent. Vous avez choisi ce nom lorsque vous avez créé l'utilisateur.

   Lorsque vous voyez l'utilisateur, choisissez le nom de l'utilisateur.
4. Sous l'onglet Permissions (Autorisations), choisissez Add inline policy (Ajouter une politique en ligne).
5. Cliquez sur l'onglet JSON et copiez la politique suivante dans la zone, en remplaçant le JSON par défaut dans la zone :

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

```
    {
      "Effect": "Allow",
      "Action": "logs:PutRetentionPolicy",
      "Resource": "*"
    }
  ]
}
```

6. Choisissez Examiner une politique.
7. Pour Name (Nom), saisissez **`CloudWatchAgentPutLogsRetention`** ou un nom similaire, et choisissez Create Policy (Créer une politique).

Installation et exécution de l' CloudWatch agent sur vos serveurs

Une fois que vous avez créé le fichier de configuration de l'agent que vous voulez et que vous avez créé un rôle IAM ou un utilisateur IAM, procédez aux étapes suivantes afin d'installer et exécuter l'agent sur vos serveurs, à l'aide de cette configuration. Tout d'abord, associez un rôle IAM ou un utilisateur IAM au serveur qui exécute l'agent. Ensuite, sur ce serveur, téléchargez le package d'agent et démarrez-le à l'aide de la configuration d'agent que vous avez créée.

Téléchargez le package de CloudWatch l'agent à l'aide d'un lien de téléchargement S3

> **Note**
>
> Pour télécharger l' CloudWatch agent, votre connexion doit utiliser le protocole TLS 1.2 ou version ultérieure.

Vous devez installer l'agent sur chaque serveur où vous allez exécuter l'agent.

AMI Amazon Linux

L' CloudWatch agent est disponible sous forme de package dans Amazon Linux 2023 et Amazon Linux 2. Si vous utilisez ce système d'exploitation, vous pouvez installer le package en entrant la commande suivante. Vous devez également vous assurer que le rôle IAM attaché à l'instance possède le rôle CloudWatchAgentServerPolicyattaché. Pour plus d'informations, veuillez consulter Créez des rôles IAM à utiliser avec l' CloudWatch agent sur les instances Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Tous les systèmes d'exploitation

Sur tous les systèmes d'exploitation pris en charge, vous pouvez télécharger et installer l' CloudWatch agent à l'aide de la ligne de commande avec un lien de téléchargement Amazon S3, comme décrit dans les étapes suivantes.

Pour chaque lien de téléchargement, vous pouvez trouver un lien général, ainsi que des liens pour chaque Région. Par exemple, pour Amazon Linux 2023, Amazon Linux 2 et l'architecture x86-64, les trois liens de téléchargement valides sont les suivants :

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	Amazon Linux 2023 et Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux /amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/ amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm .sig

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
		https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/centos/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/centos/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/debian/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/debian/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/oracle_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/oracle_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/amd64/latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/amd64/latest/* .pkg.sig amazon-cloudwatch-agent
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/windows/amd64/latest/* .msi amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/windows/amd64/latest/* .msi.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
ARM64	Amazon Linux 2023 et Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/arm64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/arm64/latest/* .deb.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Pour utiliser la ligne de commande pour installer l' CloudWatch agent sur une instance Amazon EC2

1. Téléchargez l' CloudWatch agent. Saisissez ce qui suit pour un serveur Linux. Pour *lien de téléchargement*, utilisez le lien de téléchargement approprié du tableau précédent.

   ```
   wget download-link
   ```

   Pour un serveur exécutant Windows Server, téléchargez le fichier suivant :

   ```
   https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/amazon-cloudwatch-agent.msi
   ```

2. Une fois que vous avez téléchargé le package, vous pouvez, si vous le souhaitez, vérifier la signature du package. Pour de plus amples informations, consultez Vérification de la signature du package de l' CloudWatch agent.

3. Installez le package . Si vous avez téléchargé un package RPM sur un serveur Linux, accédez au répertoire contenant le package et saisissez ce qui suit :

   ```
   sudo rpm -U ./amazon-cloudwatch-agent.rpm
   ```

   Si vous avez téléchargé un package DEB sur un serveur Linux, accédez au répertoire contenant le package et saisissez ce qui suit :

```
sudo dpkg -i -E ./amazon-cloudwatch-agent.deb
```

Si vous avez téléchargé un package MSI sur un serveur exécutant Windows Server, accédez au répertoire contenant le package et saisissez ce qui suit :

```
msiexec /i amazon-cloudwatch-agent.msi
```

Cette commande fonctionne également de l'intérieur PowerShell. Pour plus d'informations sur les options de ligne de commande MSI, consultez Command-Line Options (Options de ligne de commande) dans la documentation Microsoft Windows.

(Installation d'une instance EC2) Attacher un rôle IAM

Pour permettre à l' CloudWatch agent d'envoyer des données depuis l'instance, vous devez associer un rôle IAM à l'instance. Le rôle à attribuer est CloudWatchAgentServerRole. Vous devez avoir créé ce rôle au préalable. Pour plus d'informations, consultez Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

Pour plus d'informations sur l'attachement d'un rôle IAM à une instance, consultez Attaching an IAM Role to an Instance (Attachement d'un rôle IAM à une instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

(Installation sur un serveur local) Spécifiez les informations d'identification IAM et la région AWS

Pour permettre à l' CloudWatch agent d'envoyer des données depuis un serveur local, vous devez spécifier la clé d'accès et la clé secrète de l'utilisateur IAM que vous avez créé précédemment. Pour plus d'informations sur la création de cet utilisateur, consultez la page Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

Vous devez également spécifier la AWS région à laquelle envoyer les métriques, en utilisant le `region` champ de la `[AmazonCloudWatchAgent]` section du fichier de AWS configuration, comme dans l'exemple suivant.

```
[profile AmazonCloudWatchAgent]
region = us-west-1
```

Voici un exemple d'utilisation de la `aws configure` commande pour créer un profil nommé pour l' CloudWatch agent. Cet exemple suppose que vous utilisez le nom de profil par défaut de `AmazonCloudWatchAgent`.

Pour créer le AmazonCloudWatchAgent profil de l' CloudWatch agent

1. Si ce n'est pas déjà fait, installez-le AWS Command Line Interface sur le serveur. Pour plus d'informations, consultez Installing the AWS CLI(Installation de).
2. Sur les serveurs Linux, saisissez la commande suivante, puis suivez les invites :

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   Sur Windows Server, ouvrez PowerShell en tant qu'administrateur, entrez la commande suivante et suivez les instructions.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

Vérifier l'accès Internet

Vos instances Amazon EC2 doivent disposer d'un accès Internet sortant pour envoyer des données ou des journaux. CloudWatch 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.

Les points de terminaison et les ports à configurer sur votre proxy sont les suivants :

- Si vous utilisez l'agent pour collecter des métriques, vous devez ajouter les CloudWatch points de terminaison des régions appropriées à la liste d'autorisation. Ces points de terminaison sont répertoriés dans la section CloudWatch Points de terminaison et quotas Amazon.
- Si vous utilisez l'agent pour collecter des journaux, vous devez ajouter les points de terminaison CloudWatch des journaux pour les régions appropriées à la liste d'autorisation. Ces points de terminaison sont répertoriés dans les points de terminaison et quotas Amazon CloudWatch Logs.
- Si vous utilisez Systems Manager pour installer l'agent ou Parameter Store pour stocker votre fichier de configuration, vous devez ajouter les points de terminaison Systems Manager pour les régions appropriées afin d'autoriser la liste. Ces points de terminaison sont répertoriés dans la rubrique AWS Systems Manager endpoints and quotas.

(Facultatif) Modifiez la Configuration commune pour les informations de proxy ou de région

L' CloudWatch agent inclut un fichier de configuration appelé`common-config.toml`. Le cas échéant, vous pouvez utiliser ce fichier pour spécifier le proxy et les informations de région.

Sur un serveur exécutant Linux, ce fichier se trouve dans le répertoire `/opt/aws/amazon-cloudwatch-agent/etc`. Sur un serveur exécutant Windows Server, ce fichier se trouve dans le répertoire `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

> **Note**
>
> Nous vous recommandons d'utiliser le `common-config.toml` fichier pour fournir une configuration et des informations d'identification partagées lorsque vous exécutez l' CloudWatch agent en mode sur site. Il peut également être utile lorsque vous utilisez Amazon EC2 et que vous souhaitez réutiliser les profils et fichiers d'identification partagés existants. L'activer via le `common-config.toml` présente l'avantage supplémentaire que si votre fichier d'informations d'identification partagé est remplacé par des informations d'identification renouvelées après leur expiration, les nouvelles informations d'identification sont automatiquement récupérées par l'agent sans qu'il soit nécessaire de redémarrer.

Le `common-config.toml` par défaut est comme suit.

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for the on-premises case by
 default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
```

```
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Toutes les lignes comportent initialement un commentaire. Pour définir le profil des informations d'identification ou les paramètres de proxy, retirez # de cette ligne et indiquez une valeur. Vous pouvez éditer ce fichier manuellement ou en utilisant la fonctionnalité Exécuter la commande de `RunShellScript` dans Systems Manager :

- `shared_credential_profile`— Pour les serveurs locaux, cette ligne indique le profil d'identification de l'utilisateur IAM à utiliser pour envoyer des données. CloudWatch Si vous conservez les commentaires de cette ligne, `AmazonCloudWatchAgent` est utilisé. Pour plus d'informations sur la création de ce profil, consultez la page (Installation sur un serveur local) Spécifiez les informations d'identification IAM et la région AWS.

  Sur une instance EC2, vous pouvez utiliser cette ligne pour que l' CloudWatch agent envoie des données depuis cette instance vers CloudWatch une autre AWS région. Pour ce faire, spécifiez un profil nommé qui inclut un champ `region` en spécifiant le nom de la région destinataire.

  Si vous spécifiez une ligne `shared_credential_profile`, vous devez également supprimer le # au début de la ligne `[credentials]`.
- `shared_credential_file` – Pour que l'agent cherche des informations d'identification dans un fichier situé sur un chemin autre que le chemin par défaut, spécifiez ici le chemin d'accès complet et le nom de fichier. Le chemin d'accès par défaut est `/root/.aws` sous Linux et `C:\\Users\\Administrator\\.aws` sous Windows Server.

  Le premier exemple ci-après montre la syntaxe d'une ligne `shared_credential_file` valide pour les serveurs Linux et le deuxième exemple est valide pour Windows Server. Sous Windows Server, vous devez utiliser le caractère d'échappement \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Si vous spécifiez une ligne `shared_credential_file`, vous devez également supprimer le # au début de la ligne `[credentials]`.
- Paramètres de proxy – Si vos serveurs utilisent des proxys HTTP ou HTTPS pour contacter les services AWS , indiquez ces proxys dans les champs `http_proxy` et `https_proxy`. Si des

adresses URL doivent être exclues de la redirection de demande, indiquez-les dans le champ `no_proxy`, en les séparant par des virgules.

Démarrez l' CloudWatch agent à l'aide de la ligne de commande

Procédez comme suit pour utiliser la ligne de commande pour démarrer l' CloudWatch agent sur un serveur.

Pour démarrer l' CloudWatch agent sur un serveur à l'aide de la ligne de commande

1. Copiez le fichier de configuration de l'agent que vous souhaitez utiliser pour le serveur dans lequel vous allez exécuter l'agent. Notez le nom du chemin d'accès où vous le copiez.
2. Dans cette commande, `-a fetch-config` l'agent charge la dernière version du fichier de configuration de l' CloudWatch agent et `-s` démarre l'agent.

   Entrez l'une des commandes suivantes. Remplacez *`configuration-file-path`*par le chemin d'accès au fichier de configuration de l'agent. Ce fichier a pour nom `config.json` si vous l'avez créé avec l'assistant, et peut être appelé `amazon-cloudwatch-agent.json` si vous l'avez créé manuellement.

   Saisissez la commande suivante sur une instance EC2 exécutant Linux :

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:configuration-file-path
   ```

   Saisissez ce qui suit sur un serveur sur site exécutant Linux :

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m onPremise -s -c file:configuration-file-path
   ```

   Sur une instance EC2 exécutant Windows Server, entrez ce qui suit depuis la PowerShell console :

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
   a fetch-config -m ec2 -s -c file:configuration-file-path
   ```

   Sur un serveur local exécutant Windows Server, entrez ce qui suit depuis la PowerShell console :

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Installation de l' CloudWatch agent à l'aide de AWS Systems Manager

Utilisez les rubriques suivantes pour installer et exécuter l' CloudWatch agent à l'aide de AWS Systems Manager.

Rubriques

Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch

L'accès aux AWS ressources nécessite des autorisations. Vous pouvez créer des rôles et des utilisateurs IAM qui incluent les autorisations dont vous avez besoin pour que l' CloudWatch agent puisse écrire des métriques CloudWatch et pour communiquer avec Amazon AWS Systems Manager EC2 et. CloudWatch Vous utilisez des rôles IAM sur des instances Amazon EC2 et vous utilisez les utilisateurs IAM avec des serveurs sur site.

Un rôle ou un utilisateur permet d'installer l' CloudWatch agent sur un serveur et d'envoyer des métriques à CloudWatch. L'autre rôle ou utilisateur est nécessaire pour stocker la configuration de votre CloudWatch agent dans le magasin de paramètres de Systems Manager. Parameter Store permet à plusieurs serveurs d'utiliser une seule configuration d' CloudWatch agent.

La possibilité d'écrire dans le Parameter Store est une autorisation vaste et puissante. Elle ne doit être utilisée que lorsque vous en avez besoin, et ne doit pas être attachée à plusieurs instances dans votre déploiement. Si vous stockez la configuration de votre CloudWatch agent dans Parameter Store, nous vous recommandons ce qui suit :

- Configurez une instance où vous effectuez cette configuration.
- Utilisez le rôle IAM avec des autorisations pour écrire dans le Parameter Store uniquement sur cette instance.

- Utilisez le rôle IAM avec les autorisations nécessaires pour écrire dans Parameter Store uniquement lorsque vous travaillez avec le fichier de configuration de l' CloudWatch agent et que vous l'enregistrez.

> **Note**
>
> Nous avons récemment modifié les procédures suivantes à l'aide des nouvelles politiques `CloudWatchAgentServerPolicy` et `CloudWatchAgentAdminPolicy` créées par Amazon, plutôt que d'exiger des clients qu'ils créent ces stratégies eux-mêmes. Pour utiliser ces stratégies pour écrire le fichier de configuration de l'agent dans le Parameter Store puis le télécharger à partir du Parameter Store, le nom de votre fichier de configuration d'agent doit commencer par `AmazonCloudWatch-`. Si vous disposez d'un fichier de configuration d' CloudWatch agent dont le nom ne commence pas par`AmazonCloudWatch-`, ces règles ne peuvent pas être utilisées pour écrire le fichier dans Parameter Store ou pour télécharger le fichier depuis Parameter Store.

Créez des rôles IAM à utiliser avec l' CloudWatch agent sur les instances Amazon EC2

La première procédure crée le rôle IAM que vous devez associer à chaque instance Amazon EC2 qui exécute CloudWatch l'agent. Ce rôle fournit des autorisations pour lire les informations de l'instance et les y écrire CloudWatch.

La deuxième procédure crée le rôle IAM que vous devez associer à l'instance Amazon EC2 utilisée pour créer le fichier de configuration de CloudWatch l'agent. Cette étape est nécessaire si vous prévoyez de stocker ce fichier dans le Parameter Store du Systems Manager afin que d'autres serveurs puissent l'utiliser. Ce rôle fournit des autorisations pour écrire dans Parameter Store, en plus des autorisations pour lire les informations de l'instance et les y écrire CloudWatch. Ce rôle inclut des autorisations suffisantes pour exécuter l' CloudWatch agent ainsi que pour écrire dans Parameter Store.

> **Note**
>
> Le Parameter Store prend en charge les paramètres des niveaux Standard et Avancé. Ces niveaux de paramètres ne sont pas liés aux niveaux de détail de base, standard et avancé disponibles avec les ensembles de mesures prédéfinis par l' CloudWatch agent.

Pour créer le rôle IAM nécessaire à chaque serveur pour exécuter l'agent CloudWatch

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles), puis Create role (Créer un rôle).
3. Sous Select type of trusted entity (Sélectionner le type d'entité approuvée), choisissez service AWS .
4. Directement sous Common use cases (Cas d'utilisation courants), choisissez EC2, puis Next: Permissions (Suivant : Autorisations).
5. Dans la liste des politiques, cochez la case située à côté de CloudWatchAgentServerPolicy. Si nécessaire, utilisez la zone de recherche pour trouver la politique.
6. Pour utiliser Systems Manager afin d'installer ou de configurer l' CloudWatch agent, cochez la case à côté d'AmazonSSM ManagedInstanceCore. Cette politique AWS gérée permet à une instance d'utiliser les fonctionnalités principales du service Systems Manager. Si nécessaire, utilisez la zone de recherche pour trouver la politique. Cette politique n'est pas nécessaire si vous lancez et configurez l'agent uniquement via la ligne de commande.
7. Sélectionnez Suivant : Étiquettes.
8. (Facultatif) Ajoutez une ou plusieurs paires clé-valeur de balise afin d'organiser, de suivre ou de contrôler l'accès pour ce rôle, puis choisissez Next: Review (Suivant : Vérifier).
9. Pour Role name (Nom du rôle), entrez un nom pour votre nouveau rôle, par exemple **`CloudWatchAgentServerRole`** ou autre, en fonction de vos préférences.
10. (Facultatif) Pour Role description (Description du rôle), entrez une description.
11. Confirmez-le CloudWatchAgentServerPolicyet AmazonSSM ManagedInstanceCore apparaîtra éventuellement à côté des Politiques.
12. Choisissez Create role (Créer un rôle).

    Le rôle est maintenant créé.

La procédure suivante crée le rôle IAM que vous pouvez également écrire dans le Parameter Store. Vous pouvez utiliser ce rôle pour stocker le fichier de configuration de l'agent dans le Parameter Store afin que d'autres serveurs puissent le récupérer.

Les autorisations d'écriture dans le Parameter Store offrent un accès étendu. Ce rôle ne doit pas être attaché à tous vos serveurs et seuls les administrateurs doivent l'utiliser. Une fois le fichier

de configuration d'agent créé et copié dans le Parameter Store, vous devez détacher ce rôle de l'instance et utiliser `CloudWatchAgentServerRole` à la place.

Pour créer le rôle IAM afin de permettre à un administrateur d'écrire dans le Parameter Store

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles), puis Create role (Créer un rôle).
3. Sous Select type of trusted entity (Sélectionner le type d'entité approuvée), choisissez service AWS .
4. Immédiatement sous Choisir le service qui utilisera ce rôle, choisissez EC2, puis Next: Permissions (Suivant : Autorisations).
5. Dans la liste des politiques, cochez la case située à côté de CloudWatchAgentAdminPolicy. Si nécessaire, utilisez la zone de recherche pour trouver la politique.
6. Pour utiliser Systems Manager afin d'installer ou de configurer l' CloudWatch agent, cochez la case à côté d'AmazonSSM ManagedInstanceCore. Cette politique AWS gérée permet à une instance d'utiliser les fonctionnalités principales du service Systems Manager. Si nécessaire, utilisez la zone de recherche pour trouver la politique. Cette politique n'est pas nécessaire si vous lancez et configurez l'agent uniquement via la ligne de commande.
7. Sélectionnez Suivant : Étiquettes.
8. (Facultatif) Ajoutez une ou plusieurs paires clé-valeur de balise afin d'organiser, de suivre ou de contrôler l'accès pour ce rôle, puis choisissez Next: Review (Suivant : Vérifier).
9. Pour Role name (Nom du rôle), entrez un nom pour votre nouveau rôle, par exemple **`CloudWatchAgentAdminRole`** ou autre, en fonction de vos préférences.
10. (Facultatif) Pour Role description (Description du rôle), entrez une description.
11. Confirmez-le CloudWatchAgentAdminPolicyet AmazonSSM ManagedInstanceCore apparaîtra éventuellement à côté des Politiques.
12. Choisissez Create role (Créer un rôle).

    Le rôle est maintenant créé.

Création d'utilisateurs IAM à utiliser avec l' CloudWatch agent sur des serveurs locaux

La première procédure crée l'utilisateur IAM dont vous avez besoin pour exécuter l' CloudWatch agent. Cet utilisateur fournit les autorisations nécessaires pour envoyer des données à CloudWatch.

La seconde procédure crée l'utilisateur IAM que vous pouvez utiliser lors de la création du fichier de configuration de l' CloudWatchagent. Utilisez cette procédure pour stocker ce fichier dans le Parameter Store du Systems Manager afin que d'autres serveurs puissent l'utiliser. Cet utilisateur fournit des autorisations d'écriture dans Parameter Store, en plus des autorisations d'écriture de données CloudWatch.

> **Note**
>
> Le Parameter Store prend en charge les paramètres des niveaux Standard et Avancé. Ces niveaux de paramètres ne sont pas liés aux niveaux de détail de base, standard et avancé disponibles avec les ensembles de mesures prédéfinis par l' CloudWatch agent.

Pour créer l'utilisateur IAM nécessaire à l' CloudWatch agent pour écrire des données dans CloudWatch

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Users (Utilisateurs), puis Add user (Ajouter un utilisateur).
3. Saisissez le nom d'utilisateur du nouvel utilisateur.
4. Pour Access type (Type d'accès), choisissez Programmatic access (Accès programmatique), puis Next: Permissions (Suivant : Autorisations).
5. Pour Set permissions (Définir les autorisations), sélectionnez Attach existing policies directly (Attacher directement les politiques existantes).
6. Dans la liste des politiques, cochez la case située à côté de CloudWatchAgentServerPolicy. Si nécessaire, utilisez la zone de recherche pour trouver la politique.
7. Pour utiliser Systems Manager afin d'installer ou de configurer l' CloudWatch agent, cochez la case à côté d'AmazonSSM ManagedInstanceCore. Cette politique AWS gérée permet à une instance d'utiliser les fonctionnalités principales du service Systems Manager. Si nécessaire, utilisez la zone de recherche pour trouver la stratégie. Cette stratégie n'est pas nécessaire si vous lancez et configurez l'agent uniquement via la ligne de commande.)
8. Sélectionnez Suivant : Étiquettes.
9. (Facultatif) Ajoutez une ou plusieurs paires clé-valeur de balise afin d'organiser, de suivre ou de contrôler l'accès pour ce rôle, puis choisissez Next: Review (Suivant : Vérifier).

10. Confirmez que les stratégies affichées sont correctes et sélectionnez Create user (Créer un utilisateur).
11. Sur la ligne correspondant au nouvel utilisateur, choisissez Show (Afficher). Copiez la clé d'accès et la clé secrète dans un fichier afin de pouvoir les utiliser lors de l'installation de l'agent. Choisissez Close (Fermer).

La procédure suivante crée l'utilisateur IAM qui peut également écrire dans le Parameter Store. Vous devez utiliser cet utilisateur IAM si vous comptez stocker le fichier de configuration d'agent dans le Parameter Store afin que d'autres serveurs puissent l'utiliser. Cet utilisateur IAM fournit des autorisations pour l'écriture dans le Parameter Store. Cet utilisateur fournit également les autorisations nécessaires pour lire les informations de l'instance et les y écrire CloudWatch. Les autorisations d'écriture dans le Parameter Store du Systems Manager offrent un accès étendu. Cet utilisateur IAM ne doit pas être attaché à tous vos serveurs et seuls les administrateurs doivent l'utiliser. Vous devez utiliser cet utilisateur IAM uniquement lorsque vous stockez le fichier de configuration d'agent dans le Parameter Store.

Pour créer l'utilisateur IAM nécessaire pour stocker le fichier de configuration dans Parameter Store et envoyer des informations à CloudWatch

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Users (Utilisateurs), puis Add user (Ajouter un utilisateur).
3. Saisissez le nom d'utilisateur du nouvel utilisateur.
4. Pour Access type (Type d'accès), choisissez Programmatic access (Accès programmatique), puis Next: Permissions (Suivant : Autorisations).
5. Pour Set permissions (Définir les autorisations), sélectionnez Attach existing policies directly (Attacher directement les politiques existantes).
6. Dans la liste des politiques, cochez la case située à côté de CloudWatchAgentAdminPolicy. Si nécessaire, utilisez la zone de recherche pour trouver la politique.
7. Pour utiliser Systems Manager afin d'installer ou de configurer l' CloudWatch agent, cochez la case à côté d'AmazonSSM ManagedInstanceCore. Cette politique AWS gérée permet à une instance d'utiliser les fonctionnalités principales du service Systems Manager. Si nécessaire, utilisez la zone de recherche pour trouver la stratégie. Cette stratégie n'est pas nécessaire si vous lancez et configurez l'agent uniquement via la ligne de commande.)

8. Sélectionnez Suivant : Étiquettes.
9. (Facultatif) Ajoutez une ou plusieurs paires clé-valeur de balise afin d'organiser, de suivre ou de contrôler l'accès pour ce rôle, puis choisissez Next: Review (Suivant : Vérifier).
10. Confirmez que les stratégies affichées sont correctes et sélectionnez Create user (Créer un utilisateur).
11. Sur la ligne correspondant au nouvel utilisateur, choisissez Show (Afficher). Copiez la clé d'accès et la clé secrète dans un fichier afin de pouvoir les utiliser lors de l'installation de l'agent. Choisissez Close (Fermer).

Téléchargez et configurez l' CloudWatch agent

Cette section explique comment utiliser Systems Manager afin de télécharger l'agent, puis comment créer votre fichier de configuration de l'agent. Avant d'utiliser Systems Manager pour télécharger l'agent, vous devez vous assurer que l'instance est configurée correctement pour Systems Manager.

Installation ou mise à jour de l'agent SSM Agent

Sur une instance Amazon EC2, l' CloudWatch agent doit exécuter la version 2.2.93.0 ou ultérieure. Avant d'installer l' CloudWatch agent, mettez à jour ou installez l'agent SSM sur l'instance si ce n'est pas déjà fait.

Pour plus d'informations sur l'installation ou la mise à jour de l'agent SSM sur une instance exécutant Linux, consultez Installation et configuration de l'agent SSM Agent sur les instances Linux dans le Guide de l'utilisateur AWS Systems Manager .

Pour plus d'informations sur l'installation ou la mise à jour de l'agent, consultez Utilisation de l'agent SSM Agent dans le Guide de l'utilisateur AWS Systems Manager .

(Facultatif) Vérification des prérequis de Systems Manager

Vérifier l'accès Internet

Vos instances Amazon EC2 doivent disposer d'un accès Internet sortant pour envoyer des données ou des journaux. CloudWatch 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.

Les points de terminaison et les ports à configurer sur votre proxy sont les suivants :

- Si vous utilisez l'agent pour collecter des métriques, vous devez autoriser la liste des CloudWatch points de terminaison pour les régions appropriées. Ces points de terminaison sont répertoriés sur Amazon CloudWatch dans le Référence générale d'Amazon Web Services.
- Si vous utilisez l'agent pour collecter des journaux, vous devez autoriser la liste des points de terminaison des CloudWatch journaux pour les régions appropriées. Ces points de terminaison sont répertoriés dans Amazon CloudWatch Logs dans le Référence générale d'Amazon Web Services.
- Si vous utilisez Systems Manager pour installer l'agent ou le Parameter Store pour stocker votre fichier de configuration, vous devez permettre d'énumérer les points de terminaison Systems Manager pour les régions appropriées. Ces points de terminaison sont répertoriés dans la rubrique AWS Systems Manager du document Référence générale d'Amazon Web Services.

Procédez comme suit pour télécharger le package de l' CloudWatch agent à l'aide de Systems Manager.

Pour télécharger l' CloudWatch agent à l'aide de Systems Manager

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le panneau de navigation, choisissez Run Command (Fonctionnalité Exécuter la commande).

   -ou-

   Si la page d' AWS Systems Manager accueil s'ouvre, faites défiler la page vers le bas et choisissez Explore Run Command.
3. Sélectionnez Run Command (Exécuter la commande).
4. Dans la liste du document de commande, choisissez AWSPackageAWS-Configure.
5. Dans la zone Cibles, choisissez l'instance sur laquelle installer l' CloudWatchagent. Si vous ne voyez pas d'instance spécifique, c'est qu'elle n'est peut-être pas configurée en tant qu'instance gérée à utiliser avec Systems Manager. Pour plus d'informations, consultez la section Configuration AWS Systems Manager pour les environnements hybrides dans le guide de AWS Systems Manager l'utilisateur.
6. Dans la liste Action, choisissez Install (Installer).
7. Dans le champ Nom, saisissez *AmazonCloudWatchAgent*.
8. Conservez la définition de Version sur Dernière pour installer la dernière version de l'agent.

9. Cliquez sur Exécuter.
10. Il est également possible, dans les zones Targets and outputs (Cibles et sorties) de sélectionner le bouton en regard d'un nom d'instance et de choisir View output (Afficher les sorties). Systems Manager doit indiquer que l'agent a été correctement installé.

Créer et modifier le fichier de configuration d'agent

Après avoir téléchargé l' CloudWatch agent, vous devez créer le fichier de configuration avant de démarrer l'agent sur n'importe quel serveur.

Si vous allez enregistrer votre fichier de configuration d'agent dans le Parameter Store du Systems Manager, vous devez utiliser une instance EC2 pour enregistrer le Parameter Store. En outre, vous devez d'abord attacher à cette instance le rôle IAM `CloudWatchAgentAdminRole`. Pour plus d'informations sur l'attachement d'un rôle à une instance, consultez Attaching an IAM Role to an Instance (Attachement d'un rôle IAM à une instance) dans le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

Pour plus d'informations sur la création du fichier de configuration de l' CloudWatch agent, consultezCréation du fichier de configuration de CloudWatch l'agent.

Installation de l' CloudWatchagent sur les instances EC2 à l'aide de la configuration de votre agent

Une fois la configuration de CloudWatch l'agent enregistrée dans Parameter Store, vous pouvez l'utiliser lorsque vous installez l'agent sur d'autres serveurs.

Rubriques

- Attachement d'un rôle IAM à l'instance
- Téléchargez le package CloudWatch d'agent sur une instance Amazon EC2
- (Facultatif) Modifiez la configuration commune et le profil nommé de CloudWatch l'agent
- Démarrez l' CloudWatch agent

Attachement d'un rôle IAM à l'instance

Vous devez associer le rôle CloudWatchAgentServerRoleIAM à l'instance EC2 pour pouvoir exécuter l' CloudWatch agent sur l'instance. Ce rôle permet à l' CloudWatch agent d'effectuer des actions sur

l'instance. Vous devez avoir créé ce rôle au préalable. Pour plus d'informations, consultez Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

Pour plus d'informations, consultez les informations relatives à l'attachement d'un rôle IAM à une instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Windows.

Téléchargez le package CloudWatch d'agent sur une instance Amazon EC2

Vous devez installer l'agent sur chaque serveur où vous allez exécuter l'agent. L' CloudWatch agent est disponible sous forme de package dans Amazon Linux 2023 et Amazon Linux 2. Si vous utilisez ce système d'exploitation, vous pouvez installer le package en entrant la commande suivante. Vous devez également vous assurer que le rôle IAM attaché à l'instance possède le rôle CloudWatchAgentServerPolicyattaché. Pour plus d'informations, consultez Créez des rôles IAM à utiliser avec l' CloudWatch agent sur les instances Amazon EC2.

```
sudo yum install amazon-cloudwatch-agent
```

Sur tous les systèmes d'exploitation pris en charge, vous pouvez télécharger le package de l' CloudWatch agent à l'aide de la commande Systems Manager Run ou d'un lien de téléchargement Amazon S3. Pour de plus amples informations sur l'utilisation d'un lien de téléchargement Amazon S3, consultez Téléchargez le package de CloudWatch l'agent.

> **Note**
>
> Lorsque vous installez ou mettez à jour l' CloudWatch agent, seule l'option de désinstallation et de réinstallation est prise en charge. Vous ne pouvez pas utiliser l'option In-place update (Mises à jour sur place).

Téléchargez l' CloudWatch agent sur une instance Amazon EC2 à l'aide de Systems Manager

Avant de pouvoir utiliser Systems Manager pour installer l' CloudWatch agent, vous devez vous assurer que l'instance est correctement configurée pour Systems Manager.

Installation ou mise à jour de l'agent SSM Agent

Sur une instance Amazon EC2, l' CloudWatch agent doit exécuter la version 2.2.93.0 ou ultérieure. Avant d'installer l' CloudWatch agent, mettez à jour ou installez l'agent SSM sur l'instance si ce n'est pas déjà fait.

Pour plus d'informations sur l'installation ou la mise à jour de SSM Agent sur une instance exécutant Linux, consultez Installation et configuration de SSM Agent sur les instances Linux dans le Guide de l'utilisateur AWS Systems Manager .

Pour plus d'informations sur l'installation ou la mise à jour de de l'agent SSM sur une instance exécutant Windows Server, consultez Installation et configuration de l'agent SSM Agent sur les instances Windows dans le Guide de l'utilisateur AWS Systems Manager .

(Facultatif) Vérification des prérequis de Systems Manager

Avant d'utiliser la commande Run Command de Systems Manager pour installer et configurer l' CloudWatch agent, vérifiez que vos instances répondent aux exigences minimales de Systems Manager. Pour plus d'informations, consultez Configuration d' AWS Systems Manager dans le Guide de l'utilisateur AWS Systems Manager .

Vérifier l'accès Internet

Vos instances Amazon EC2 doivent disposer d'un accès Internet sortant pour envoyer des données ou des journaux. CloudWatch 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.

Téléchargez le package de CloudWatch l'agent

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

Pour télécharger l' CloudWatch agent à l'aide de la commande Exécuter

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le panneau de navigation, choisissez Run Command (Fonctionnalité Exécuter la commande).

   -ou-

   Si la page d' AWS Systems Manager accueil s'ouvre, faites défiler la page vers le bas et choisissez Explore Run Command.

3. Sélectionnez Run Command (Exécuter la commande).
4. Dans la liste du document de commande, choisissez AWSPackageAWS-Configure.
5. Dans la zone Cibles, choisissez l'instance sur laquelle vous souhaitez installer l' CloudWatch agent. Si vous ne voyez pas d'instance spécifique, c'est qu'elle n'est peut-être pas configurée pour la fonctionnalité Exécuter la commande. Pour plus d'informations, consultez Configuration de AWS Systems Manager dans des environnements hybrides dans le Guide de l'utilisateur AWS Systems Manager .
6. Dans la liste Action, choisissez Install (Installer).
7. Dans la case Nom, saisissez *AmazonCloudWatchAgent*.
8. Conservez la définition de Version sur Dernière pour installer la dernière version de l'agent.
9. Cliquez sur Exécuter.
10. Il est également possible, dans les zones Targets and outputs (Cibles et sorties) de sélectionner le bouton en regard d'un nom d'instance et de choisir View output (Afficher les sorties). Systems Manager doit indiquer que l'agent a été correctement installé.

(Facultatif) Modifiez la configuration commune et le profil nommé de CloudWatch l'agent

L' CloudWatch agent inclut un fichier de configuration appelé`common-config.toml`. Le cas échéant, vous pouvez utiliser ce fichier pour spécifier le proxy et les informations de région.

Sur un serveur exécutant Linux, ce fichier se trouve dans le répertoire `/opt/aws/amazon-cloudwatch-agent/etc`. Sur un serveur exécutant Windows Server, ce fichier se trouve dans le répertoire `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

Le `common-config.toml` par défaut est comme suit :

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
```

```
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Toutes les lignes comportent initialement un commentaire. Pour définir le profil des informations d'identification ou les paramètres de proxy, retirez # de cette ligne et indiquez une valeur. Vous pouvez éditer ce fichier manuellement ou en utilisant la fonctionnalité Exécuter la commande `RunShellScript` dans Systems Manager :

- `shared_credential_profile`— Pour les serveurs locaux, cette ligne indique le profil d'identification de l'utilisateur IAM à utiliser pour envoyer des données. CloudWatch Si vous conservez les commentaires de cette ligne, `AmazonCloudWatchAgent` est utilisé.

  Sur une instance EC2, vous pouvez utiliser cette ligne pour que l' CloudWatch agent envoie des données depuis cette instance vers CloudWatch une autre AWS région. Pour ce faire, spécifiez un profil nommé qui inclut un champ `region` en spécifiant le nom de la région destinataire.

  Si vous spécifiez une ligne `shared_credential_profile`, vous devez également supprimer le # au début de la ligne `[credentials]`.
- `shared_credential_file` – Pour que l'agent cherche des informations d'identification dans un fichier situé sur un chemin autre que le chemin par défaut, spécifiez ici le chemin d'accès complet et le nom de fichier. Le chemin d'accès par défaut est `/root/.aws` sous Linux et `C:\\Users\\Administrator\\.aws` sous Windows Server.

  Le premier exemple ci-après montre la syntaxe d'une ligne `shared_credential_file` valide pour les serveurs Linux et le deuxième exemple est valide pour Windows Server. Sous Windows Server, vous devez utiliser le caractère d'échappement \.

  ```
  shared_credential_file= "/usr/username/credentials"
  ```

  ```
  shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
  ```

  Si vous spécifiez une ligne `shared_credential_file`, vous devez également supprimer le # au début de la ligne `[credentials]`.

- Paramètres de proxy – Si vos serveurs utilisent des proxys HTTP ou HTTPS pour contacter les services AWS , indiquez ces proxys dans les champs `http_proxy` et `https_proxy`. Si des adresses URL doivent être exclues de la redirection de demande, indiquez-les dans le champ `no_proxy`, en les séparant par des virgules.

Démarrez l' CloudWatch agent

Vous pouvez démarrer l'agent à l'aide de la fonctionnalité Exécuter la commande de Systems Manager ou de la ligne de commande.

Démarrez l' CloudWatch agent à l'aide de la commande Run de Systems Manager

Procédez comme suit pour démarrer l'agent à l'aide de la commande Exécuter la commande Systems Manager.

Pour démarrer l' CloudWatch agent à l'aide de la commande Exécuter

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le panneau de navigation, choisissez Run Command (Fonctionnalité Exécuter la commande).

   -ou-

   Si la page d' AWS Systems Manager accueil s'ouvre, faites défiler la page vers le bas et choisissez Explore Run Command.
3. Sélectionnez Run Command (Exécuter la commande).
4. Dans la liste du document de commande, choisissez AmazonCloudWatch- ManageAgent.
5. Dans la zone Cibles, choisissez l'instance sur laquelle vous avez installé l' CloudWatch agent.
6. Dans la liste Action, choisissez configure (configurer).
7. Dans la liste Optional Configuration Source (Source de configuration facultative), choisissez ssm.
8. Dans la zone Emplacement de configuration facultative, saisissez le nom du paramètre Systems Manager du fichier de configuration d'agent que vous avez créé et enregistré dans le Parameter Store du Systems Manager, tel que décrit dans Création du fichier de configuration de CloudWatch l'agent.
9. Dans la liste Optional Restart (Redémarrage facultatif), choisissez oui pour démarrer l'agent une fois que vous avez terminé ces étapes.

10. Cliquez sur Exécuter.
11. Il est également possible, dans les zones Targets and outputs (Cibles et sorties) de sélectionner le bouton en regard d'un nom d'instance et de choisir View output (Afficher les sorties). Systems Manager doit indiquer que l'agent a été correctement démarré.

Démarrez l' CloudWatch agent sur une instance Amazon EC2 à l'aide de la ligne de commande

Suivez ces étapes pour utiliser la ligne de commande afin d'installer l' CloudWatch agent sur une instance Amazon EC2.

Pour utiliser la ligne de commande pour démarrer l' CloudWatch agent sur une instance Amazon EC2

- Dans cette commande, `-a fetch-config` l'agent charge la dernière version du fichier de configuration de l' CloudWatch agent et `-s` démarre l'agent.

  Linux et macOS : saisissez la commande suivante si vous avez enregistré le fichier dans le Parameter Store du Systems Manager :

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Linux et macOS : saisissez la commande suivante si vous avez enregistré le fichier sur l'ordinateur local. Remplacez *`configuration-file-path`*par le chemin d'accès au fichier de configuration de l'agent. Ce fichier a pour nom `config.json` si vous l'avez créé avec l'assistant, et peut être appelé `amazon-cloudwatch-agent.json` si vous l'avez créé manuellement.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m ec2 -s -c file:configuration-file-path
  ```

  Windows Server : si vous avez enregistré le fichier de configuration de l'agent dans le magasin de paramètres de Systems Manager, entrez ce qui suit depuis la PowerShell console :

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
  a fetch-config -m ec2 -s -c ssm:configuration-parameter-store-name
  ```

  Windows Server : si vous avez enregistré le fichier de configuration de l'agent sur l'ordinateur local, entrez ce qui suit depuis la PowerShell console :

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\config.json"
```

Installation de l' CloudWatch agent sur des serveurs locaux

Si vous avez téléchargé l' CloudWatch agent sur un ordinateur et créé le fichier de configuration de l'agent souhaité, vous pouvez utiliser ce fichier de configuration pour installer l'agent sur d'autres serveurs locaux.

Téléchargez l' CloudWatch agent sur un serveur local

Vous pouvez télécharger le package de l' CloudWatch agent à l'aide de la commande Run de Systems Manager ou d'un lien de téléchargement Amazon S3. Pour de plus amples informations sur l'utilisation d'un lien de téléchargement Amazon S3, consultez Téléchargez le package de CloudWatch l'agent.

Téléchargement à l'aide de Systems Manager

Pour utiliser la fonctionnalité Exécuter la commande de Systems Manager, vous devez enregistrer votre serveur sur site auprès d'Amazon EC2 Systems Manager. Pour plus d'informations, consultez Setting Up Systems Manager in Hybrid Environments (Configuration de Systems Manager dans des environnements hybrides) dans le Guide de l'utilisateur AWS Systems Manager .

Si vous avez déjà enregistré votre serveur, mettez à jour l'agent SSM Agent vers la dernière version.

Pour plus d'informations sur la mise à jour de l'agent SSM sur un serveur exécutant Linux, consultez Installation de l'agent SSM Agent pour un environnement hybride (Linux) dans le Guide de l'utilisateur AWS Systems Manager .

Pour plus d'informations sur la mise à jour de l'agent SSM sur un serveur exécutant Windows Server, consultez Installation de l'agent SSM Agent pour un environnement hybride (Windows) dans le Guide de l'utilisateur AWS Systems Manager .

Pour utiliser l'agent SSM pour télécharger le package de l' CloudWatch agent sur un serveur local

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.

2. Dans le panneau de navigation, choisissez Run Command (Fonctionnalité Exécuter la commande).

   -ou-

   Si la page d' AWS Systems Manager accueil s'ouvre, faites défiler la page vers le bas et choisissez Explore Run Command.
3. Sélectionnez Run Command (Exécuter la commande).
4. Dans la liste du document de commande, sélectionnez le bouton à côté de AWSPackageAWS-Configure.
5. Dans la zone Cibles, sélectionnez le serveur sur lequel installer l' CloudWatch agent. Si vous ne voyez pas de serveur spécifique, c'est qu'il n'est peut-être pas configuré pour la fonctionnalité Exécuter la commande. Pour plus d'informations, consultez Configuration de AWS Systems Manager dans des environnements hybrides dans le Guide de l'utilisateur AWS Systems Manager .
6. Dans la liste Action, choisissez Install (Installer).
7. Dans la case Nom, saisissez *AmazonCloudWatchAgent*.
8. Conservez Version vide pour installer la dernière version de l'agent.
9. Cliquez sur Exécuter.

   Le package d'agents est téléchargé et les étapes suivantes permettent de le configurer et de le démarrer.

(Installation sur un serveur local) Spécifiez les informations d'identification IAM et la région AWS

Pour permettre à l' CloudWatch agent d'envoyer des données depuis un serveur local, vous devez spécifier la clé d'accès et la clé secrète de l'utilisateur IAM que vous avez créé précédemment. Pour plus d'informations sur la création de cet utilisateur, consultez la page Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

Vous devez également spécifier la AWS région à laquelle envoyer les métriques, à l'aide du `region` champ.

Voici un exemple de ce fichier.

```
[AmazonCloudWatchAgent]
aws_access_key_id=my_access_key
aws_secret_access_key=my_secret_key
```

```
region = us-west-1
```

Pour *my_access_key* et *my_secret_key*, utilisez les clés provenant de l'utilisateur IAM qui ne possède pas les autorisations pour écrire sur le Parameter Store du Systems Manager. Pour plus d'informations sur les utilisateurs IAM nécessaires à l' CloudWatch agent, consultezCréation d'utilisateurs IAM à utiliser avec l' CloudWatch agent sur des serveurs locaux.

Aucune autre action n'est requise de votre part si vous nommez ce profil `AmazonCloudWatchAgent`. Le cas échéant, vous pouvez lui attribuer un nom différent et spécifier ce nom comme valeur pour `shared_credential_profile` dans le fichier `common-config.toml`, ce qui est détaillé dans la section suivante.

Voici un exemple d'utilisation de la aws configure commande pour créer un profil nommé pour l' CloudWatch agent. Cet exemple suppose que vous utilisez le nom de profil par défaut de `AmazonCloudWatchAgent`.

Pour créer le AmazonCloudWatchAgent profil de l' CloudWatch agent

1. Si ce n'est pas déjà fait, installez-le AWS Command Line Interface sur le serveur. Pour plus d'informations, consultez Installing the AWS CLI(Installation de).
2. Sur les serveurs Linux, saisissez la commande suivante, puis suivez les invites :

   ```
   sudo aws configure --profile AmazonCloudWatchAgent
   ```

   Sur Windows Server, ouvrez PowerShell en tant qu'administrateur, entrez la commande suivante et suivez les instructions.

   ```
   aws configure --profile AmazonCloudWatchAgent
   ```

(Facultatif) Modification de la configuration commune et du profil nommé de CloudWatch l'agent

L' CloudWatch agent inclut un fichier de configuration appelé`common-config.toml`. Le cas échéant, vous pouvez utiliser ce fichier pour spécifier le proxy et les informations de région.

Sur un serveur exécutant Linux, ce fichier se trouve dans le répertoire `/opt/aws/amazon-cloudwatch-agent/etc`. Sur un serveur exécutant Windows Server, ce fichier se trouve dans le répertoire `C:\ProgramData\Amazon\AmazonCloudWatchAgent`.

Le `common-config.toml` par défaut est comme suit :

```
# This common-config is used to configure items used for both ssm and cloudwatch access


## Configuration for shared credential.
## Default credential strategy will be used if it is absent here:
##            Instance role is used for EC2 case by default.
##            AmazonCloudWatchAgent profile is used for onPremise case by default.
# [credentials]
#    shared_credential_profile = "{profile_name}"
#    shared_credential_file= "{file_name}"

## Configuration for proxy.
## System-wide environment-variable will be read if it is absent here.
## i.e. HTTP_PROXY/http_proxy; HTTPS_PROXY/https_proxy; NO_PROXY/no_proxy
## Note: system-wide environment-variable is not accessible when using ssm run-command.
## Absent in both here and environment-variable means no proxy will be used.
# [proxy]
#    http_proxy = "{http_url}"
#    https_proxy = "{https_url}"
#    no_proxy = "{domain}"
```

Toutes les lignes comportent initialement un commentaire. Pour définir le profil des informations d'identification ou les paramètres de proxy, retirez # de cette ligne et indiquez une valeur. Vous pouvez éditer ce fichier manuellement ou en utilisant la fonctionnalité Exécuter la commande `RunShellScript` dans Systems Manager :

- `shared_credential_profile`— Pour les serveurs locaux, cette ligne indique le profil d'identification de l'utilisateur IAM à utiliser pour envoyer des données. CloudWatch Si vous conservez les commentaires de cette ligne, `AmazonCloudWatchAgent` est utilisé. Pour plus d'informations sur la création de ce profil, consultez la page (Installation sur un serveur local) Spécifiez les informations d'identification IAM et la région AWS.

  Sur une instance EC2, vous pouvez utiliser cette ligne pour que l' CloudWatch agent envoie des données depuis cette instance vers CloudWatch une autre AWS région. Pour ce faire, spécifiez un profil nommé qui inclut un champ `region` en spécifiant le nom de la région destinataire.

  Si vous spécifiez une ligne `shared_credential_profile`, vous devez également supprimer le # au début de la ligne `[credentials]`.
- `shared_credential_file` – Pour que l'agent cherche des informations d'identification dans un fichier situé sur un chemin autre que le chemin par défaut, spécifiez ici le chemin d'accès complet

et le nom de fichier. Le chemin d'accès par défaut est `/root/.aws` sous Linux et `C:\\Users\\Administrator\\.aws` sous Windows Server.

Le premier exemple ci-après montre la syntaxe d'une ligne `shared_credential_file` valide pour les serveurs Linux et le deuxième exemple est valide pour Windows Server. Sous Windows Server, vous devez utiliser le caractère d'échappement \.

```
shared_credential_file= "/usr/username/credentials"
```

```
shared_credential_file= "C:\\Documents and Settings\\username\\.aws\\credentials"
```

Si vous spécifiez une ligne `shared_credential_file`, vous devez également supprimer le # au début de la ligne `[credentials]`.

- Paramètres de proxy – Si vos serveurs utilisent des proxys HTTP ou HTTPS pour contacter les services AWS , indiquez ces proxys dans les champs `http_proxy` et `https_proxy`. Si des adresses URL doivent être exclues de la redirection de demande, indiquez-les dans le champ `no_proxy`, en les séparant par des virgules.

Démarrage de l'agent CloudWatch

Vous pouvez démarrer l' CloudWatch agent à l'aide de la commande Run Command de Systems Manager ou de la ligne de commande.

Pour utiliser l'agent SSM pour démarrer l' CloudWatch agent sur un serveur local

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le panneau de navigation, choisissez Run Command (Fonctionnalité Exécuter la commande).

   -ou-

   Si la page d' AWS Systems Manager accueil s'ouvre, faites défiler la page vers le bas et choisissez Explore Run Command.
3. Sélectionnez Run Command (Exécuter la commande).
4. Dans la liste du document de commande, sélectionnez le bouton à côté de AmazonCloudWatch-ManageAgent.

5. Dans la zone Targets (Cibles), sélectionnez l'instance où vous avez installé l'agent.
6. Dans la liste Action, choisissez configure (configurer).
7. Dans la liste Mode, choisissez onPremise (sur site).
8. Dans la zone Optional Configuration Location (Emplacement de configuration facultative), saisissez le nom du fichier de configuration d'agent que vous avez créé avec l'assistant et stocké dans le Parameter Store.
9. Cliquez sur Exécuter.

   L'agent commence avec la configuration que vous avez spécifiée dans le fichier de configuration.

Pour utiliser la ligne de commande pour démarrer l' CloudWatch agent sur un serveur local

- Dans cette commande, `-a fetch-config` l'agent charge la dernière version du fichier de configuration de l' CloudWatch agent et `-s` démarre l'agent.

  Linux : saisissez la commande suivante si vous avez enregistré le fichier dans le Parameter Store du Systems Manager :

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m onPremise -s -c ssm:configuration-parameter-store-name
  ```

  Linux – Saisissez la commande suivante si vous avez enregistré le fichier sur l'ordinateur local : Remplacez *`configuration-file-path`*par le chemin d'accès au fichier de configuration de l'agent. Ce fichier a pour nom `config.json` si vous l'avez créé avec l'assistant, et peut être appelé `amazon-cloudwatch-agent.json` si vous l'avez créé manuellement.

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
  config -m onPremise -s -c file:configuration-file-path
  ```

  Windows Server : si vous avez enregistré le fichier de configuration de l'agent dans le magasin de paramètres de Systems Manager, entrez ce qui suit depuis la PowerShell console :

  ```
  & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
  a fetch-config -m onPremise -s -c ssm:configuration-parameter-store-name
  ```

  Windows Server : si vous avez enregistré le fichier de configuration de l'agent sur l'ordinateur local, entrez ce qui suit depuis la PowerShell console. Remplacez *`configuration-file-`*

*path*par le chemin d'accès au fichier de configuration de l'agent. Ce fichier a pour nom `config.json` si vous l'avez créé avec l'assistant, et peut être appelé `amazon-cloudwatch-agent.json` si vous l'avez créé manuellement.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1" -
a fetch-config -m onPremise -s -c file:configuration-file-path
```

Installation de l' CloudWatchagent sur de nouvelles instances à l'aide de AWS CloudFormation

Amazon a téléchargé plusieurs AWS CloudFormation modèles pour vous aider GitHub à installer et à mettre à jour l' CloudWatch agent sur les nouvelles instances Amazon EC2. Pour plus d'informations sur l'utilisation AWS CloudFormation, voir Qu'est-ce que c'est AWS CloudFormation ? .

L'emplacement du modèle est Déployer l' CloudWatch agent Amazon sur les instances EC2 à l'aide AWS CloudFormation de. Cet emplacement inclut à la fois les répertoires `inline` et `ssm`. Chacun de ces répertoires contient des modèles pour les instances Linux et Windows.

- La configuration de l' CloudWatch agent est intégrée aux AWS CloudFormation modèles du `inline` répertoire. Par défaut, les modèles Linux collectent les métriques `mem_used_percent` et `swap_used_percent` et les modèles Windows collectent `Memory % Committed Bytes In Use` et `Paging File % Usage`.

  Modifiez la section suivante du modèle afin de modifier ces modèles pour la collecte de différentes métriques. L'exemple suivant est issu du modèle pour les serveurs Linux. Respectez le format et la syntaxe du fichier de configuration de l'agent pour effectuer ces modifications. Pour de plus amples informations, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

  ```
  {
     "metrics":{
        "append_dimensions":{
           "AutoScalingGroupName":"${!aws:AutoScalingGroupName}",
           "ImageId":"${!aws:ImageId}",
           "InstanceId":"${!aws:InstanceId}",
           "InstanceType":"${!aws:InstanceType}"
        },
        "metrics_collected":{
  ```

```
        "mem":{
           "measurement":[
              "mem_used_percent"
           ]
        },
        "swap":{
           "measurement":[
              "swap_used_percent"
           ]
        }
     }
  }
}
```

> **Note**
>
> Dans les modèles en ligne, toutes les variables d'espace réservé doivent commencer par un point d'exclamation (!) comme caractère d'échappement. Vous pouvez le constater dans l'exemple de modèle. Si vous ajoutez d'autres variables d'espace réservé, veillez à ajouter un point d'exclamation avant le nom.

- Les modèles dans le répertoire `ssm` chargent un fichier de configuration d'agent à partir du Parameter Store. Pour utiliser ces modèles, vous devez d'abord créer un fichier de configuration et le charger dans le Parameter Store. Ensuite, vous indiquez le nom du Parameter Store du fichier dans le modèle. Vous pouvez créer le fichier de configuration manuellement ou à l'aide de l'assistant. Pour plus d'informations, consultez Création du fichier de configuration de CloudWatch l'agent.

Vous pouvez utiliser les deux types de modèles pour installer l' CloudWatch agent et pour mettre à jour la configuration de l'agent.

Tutoriel : Installation et configuration de l' CloudWatch agent à l'aide d'un AWS CloudFormation modèle intégré

Ce didacticiel explique comment AWS CloudFormation installer l' CloudWatch agent sur une nouvelle instance Amazon EC2. Ce didacticiel s'installe sur une nouvelle instance qui exécute Amazon Linux 2 à l'aide de modèles en ligne qui ne nécessitent pas l'utilisation de fichier de configuration JSON ou du Parameter Store. Le modèle en ligne inclut la configuration de l'agent dans le modèle. Dans ce tutoriel, vous utilisez la configuration de l'agent par défaut contenue dans le modèle.

Après la procédure d'installation de l'agent, le tutoriel vous explique comment mettre à jour l'agent.

À utiliser AWS CloudFormation pour installer l' CloudWatch agent sur une nouvelle instance

1. Téléchargez le modèle depuis GitHub. Dans ce didacticiel, téléchargez le modèle en ligne pour Amazon Linux 2 comme suit :

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/inline/amazon_linux.template
   ```

2. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
3. Sélectionnez Créer la pile.
4. Pour Choisir un modèle, sélectionnez Télécharger un modèle sur Amazon S3, choisissez le modèle téléchargé, puis Next (Suivant).
5. Sur la page Spécifier les détails, renseignez les paramètres suivants, puis choisissez Next (Suivant) :

   - Nom de pile : Choisissez un nom de pile pour votre AWS CloudFormation pile.
   - IAMRole : Choisissez un rôle IAM autorisé à écrire des CloudWatch métriques, des journaux et des traces. Pour plus d'informations, consultez Créez des rôles IAM à utiliser avec l' CloudWatch agent sur les instances Amazon EC2.
   - InstanceAMI : choisissez une AMI valide dans la région dans laquelle vous allez lancer votre pile.
   - InstanceType: Choisissez un type d'instance valide.
   - KeyName: Pour activer l'accès SSH à la nouvelle instance, choisissez une paire de clés Amazon EC2 existante. Si vous ne possédez pas déjà une paire de clés Amazon EC2, vous pouvez en créer une dans l' AWS Management Console. Pour de plus amples informations, consultez Paires de clés Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.
   - SSHLocation : spécifie la plage d'adresses IP pouvant être utilisée pour se connecter à l'instance à l'aide de SSH. La valeur par défaut autorise l'accès depuis n'importe quelle adresse IP.
6. Dans la page Options, vous pouvez étiqueter les ressources de votre pile. Choisissez Next (Suivant).

7. Sur la page Review (Vérification), vérifiez vos informations, reconnaissez que la pile peut créer des ressources IAM, puis choisissez Create (Créer).

   Si vous actualisez la console, vous voyez que la nouvelle pile présente l'état `CREATE_IN_PROGRESS`.

8. Une fois l'instance créée, vous pouvez la voir dans la console Amazon EC2. Ensuite, vous pouvez vous connecter à l'hôte et vérifier la progression.

   Utilisez la commande suivante pour confirmer que l'agent est installé :

   ```
   rpm -qa amazon-cloudwatch-agent
   ```

   Utilisez la commande suivante pour confirmer que l'agent est en cours d'exécution :

   ```
   ps aux | grep amazon-cloudwatch-agent
   ```

La procédure suivante explique comment mettre AWS CloudFormation à jour l' CloudWatch agent à l'aide d'un modèle intégré. Par défaut, le modèle en ligne collecte la métrique `mem_used_percent`. Dans ce tutoriel, vous modifiez la configuration de l'agent pour arrêter la collecte de cette métrique.

À utiliser AWS CloudFormation pour mettre à jour l' CloudWatch agent

1. Dans le modèle que vous avez téléchargé lors de la procédure précédente, supprimez les lignes suivantes, puis enregistrez le modèle :

   ```
   "mem": {

        "measurement": [
            "mem_used_percent"
          ]
    },
   ```

2. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
3. Sur le AWS CloudFormation tableau de bord, sélectionnez la pile que vous avez créée et choisissez Update Stack.

4. Pour Select Template (Sélectionner un modèle), sélectionnez Télécharger un modèle sur Amazon S3, choisissez le modèle modifié, puis Next (Suivant).
5. Sur la page Options, choisissez Next (Suivant), puis Next.
6. Dans la page Review (Révision), passez en revue vos informations et choisissez Update (Mettre à jour).

   Après un certain temps, `UPDATE_COMPLETE` s'affiche.

Tutoriel : Installation de l' CloudWatch agent à l'aide AWS CloudFormation d'un magasin de paramètres

Ce didacticiel explique comment AWS CloudFormation installer l' CloudWatch agent sur une nouvelle instance Amazon EC2. Ce didacticiel s'installe sur une nouvelle instance qui exécute Amazon Linux 2 à l'aide d'un fichier de configuration de l'agent que vous créez et enregistrez dans le Parameter Store.

Après la procédure d'installation de l'agent, le tutoriel vous explique comment mettre à jour l'agent.

À utiliser AWS CloudFormation pour installer l' CloudWatch agent sur une nouvelle instance à l'aide d'une configuration provenant du Parameter Store

1. Si ce n'est pas déjà fait, téléchargez le package de l' CloudWatch agent sur l'un de vos ordinateurs afin de créer le fichier de configuration de l'agent. Pour plus d'informations et pour télécharger l'agent à l'aide du Parameter Store, consultez Téléchargez et configurez l' CloudWatch agent. Pour plus d'informations sur le téléchargement du package à l'aide de la ligne de commande, consultez Téléchargez et configurez l' CloudWatchagent à l'aide de la ligne de commande.
2. Créez le fichier de configuration d'agent et enregistrez-le dans le Parameter Store. Pour plus d'informations, consultez Création du fichier de configuration de CloudWatch l'agent.
3. Téléchargez le modèle GitHub comme suit :

   ```
   curl -O https://raw.githubusercontent.com/awslabs/aws-cloudformation-templates/
   master/aws/solutions/AmazonCloudWatchAgent/ssm/amazon_linux.template
   ```

4. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
5. Sélectionnez Créer la pile.

6. Pour Choisir un modèle, sélectionnez Télécharger un modèle sur Amazon S3, choisissez le modèle que vous avez téléchargé, puis Next (Suivant).
7. Sur la page Specify Details (Spécifier les détails), renseignez les paramètres suivants en conséquence, puis choisissez Next (Suivant).

   - Nom de pile : Choisissez un nom de pile pour votre AWS CloudFormation pile.
   - IAMRole : Choisissez un rôle IAM autorisé à écrire des CloudWatch métriques, des journaux et des traces. Pour plus d'informations, consultez Créez des rôles IAM à utiliser avec l' CloudWatch agent sur les instances Amazon EC2.
   - InstanceAMI : choisissez une AMI valide dans la région dans laquelle vous allez lancer votre pile.
   - InstanceType: Choisissez un type d'instance valide.
   - KeyName: Pour activer l'accès SSH à la nouvelle instance, choisissez une paire de clés Amazon EC2 existante. Si vous ne possédez pas déjà une paire de clés Amazon EC2, vous pouvez en créer une dans l' AWS Management Console. Pour de plus amples informations, consultez Paires de clés Amazon EC2 dans le Amazon EC2 Guide de l'utilisateur pour les instances Linux.
   - SSHLocation : spécifie la plage d'adresses IP pouvant être utilisée pour se connecter à l'instance à l'aide de SSH. La valeur par défaut autorise l'accès depuis n'importe quelle adresse IP.
   - SSMKey : Spécifie le fichier de configuration de l'agent que vous avez créé et enregistré dans le Parameter Store.
8. Dans la page Options, vous pouvez étiqueter les ressources de votre pile. Choisissez Next (Suivant).
9. Sur la page Review (Vérification), vérifiez vos informations, reconnaissez que la pile peut créer des ressources IAM, puis choisissez Create (Créer).

   Si vous actualisez la console, vous voyez que la nouvelle pile présente l'état `CREATE_IN_PROGRESS`.
10. Une fois l'instance créée, vous pouvez la voir dans la console Amazon EC2. Ensuite, vous pouvez vous connecter à l'hôte et vérifier la progression.

    Utilisez la commande suivante pour confirmer que l'agent est installé :

    ```
    rpm -qa amazon-cloudwatch-agent
    ```

Utilisez la commande suivante pour confirmer que l'agent est en cours d'exécution :

```
ps aux | grep amazon-cloudwatch-agent
```

La procédure suivante montre comment mettre AWS CloudFormation à jour l' CloudWatch agent à l'aide d'une configuration d'agent que vous avez enregistrée dans Parameter Store.

À utiliser pour mettre AWS CloudFormation à jour l' CloudWatch agent à l'aide d'une configuration dans Parameter Store

1. Modifiez le fichier de configuration de l'agent stocké dans le Parameter Store avec la nouvelle configuration de votre choix.
2. Dans le AWS CloudFormation modèle que vous avez téléchargé dans la the section called “Tutoriel : Installation de l' CloudWatch agent à l'aide AWS CloudFormation d'un magasin de paramètres” rubrique, modifiez le numéro de version. Par exemple, vous pouvez remplacer `VERSION=1.0` par `VERSION=2.0`.
3. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
4. Sur le AWS CloudFormation tableau de bord, sélectionnez la pile que vous avez créée et choisissez Update Stack.
5. Pour Sélectionner un modèle, sélectionnez Télécharger un modèle sur Amazon S3, sélectionnez le modèle que vous venez de modifier, puis Next (Suivant).
6. Sur la page Options, choisissez Next (Suivant), puis Next.
7. Dans la page Review (Révision), passez en revue vos informations et choisissez Update (Mettre à jour).

   Après un certain temps, `UPDATE_COMPLETE` s'affiche.

Résolution des problèmes d'installation de l' CloudWatch agent avec AWS CloudFormation

Cette section vous aide à résoudre les problèmes liés à l'installation et à la mise à jour de l' CloudWatch agent à l'aide AWS CloudFormation de.

Détection de l'échec d'une mise à jour

Si vous avez l' AWS CloudFormation habitude de mettre à jour la configuration de votre CloudWatch agent et que vous utilisez une configuration non valide, l'agent arrête d'envoyer des métriques à CloudWatch. Pour vérifier rapidement si la mise à jour de la configuration d'un agent a réussi, consultez le fichier `cfn-init-cmd.log`. Sur un serveur Linux, le fichier se situe à l'emplacement `/var/log/cfn-init-cmd.log`. Sur une instance Windows, le fichier se situe à l'emplacement `C:\cfn\log\cfn-init-cmd.log`.

Métriques manquantes

Si vous ne voyez pas les métriques prévues après l'installation ou la mise à jour de l'agent, vérifiez que l'agent est configuré pour collecter cette métrique. Pour cela, vérifiez le fichier `amazon-cloudwatch-agent.json` pour vous assurer que la métrique est répertoriée, et que vous recherchez dans le bon espace de noms de métrique. Pour plus d'informations, consultez CloudWatch fichiers et emplacements des agents.

CloudWatch préférence d'identification de l'agent

Cette section décrit la chaîne de fournisseurs d'informations d'identification que l' CloudWatch agent utilise pour obtenir des informations d'identification lorsqu'il communique avec d'autres AWS services et API. La commande est la suivante. Les préférences répertoriées aux numéros deux à cinq de la liste suivante suivent le même ordre de préférence que celui défini dans le AWS SDK. Pour plus d'informations, consultez la section Spécification des informations d'identification dans la documentation du SDK.

1. Fichiers de configuration et d'informations d'identification partagés tels que définis dans le `common-config.toml` fichier de l' CloudWatch agent. Pour plus d'informations, consultez (Facultatif) Modifiez la Configuration commune pour les informations de proxy ou de région.
2. AWS Variables d'environnement du SDK

> **Important**
>
> Sous Linux, si vous exécutez l' CloudWatch agent à l'aide du `amazon-cloudwatch-agent-ctl` script, celui-ci démarre l'agent en tant que `systemd` service. Dans ce cas, les variables d'environnement telles que `HOMEAWS_ACCESS_KEY_ID`, et ne `AWS_SECRET_ACCESS_KEY` sont pas accessibles par l'agent.

3. Fichiers de configuration et d'informations d'identification partagés trouvés dans `$HOME/%USERPROFILE%`

> **Note**
>
> L' CloudWatch agent `$HOME` recherche `.aws/credentials` dans Linux et macOS et dans `%USERPROFILE%` Windows. Contrairement au AWS SDK, l' CloudWatch agent ne dispose pas de méthodes de secours pour déterminer le répertoire de base si les variables d'environnement sont inaccessibles. Cette différence de comportement vise à maintenir la rétrocompatibilité avec les implémentations antérieures du AWS SDK.
> En outre, contrairement aux informations d'identification partagées présentes dans`common-config.toml`, si les informations d'identification partagées AWS dérivées du SDK expirent et font l'objet d'une rotation, les informations d'identification renouvelées ne sont pas automatiquement récupérées par l' CloudWatch agent et nécessitent un redémarrage de l'agent pour ce faire.

4. AWS Identity and Access Management Rôle pour les tâches en présence d'une application utilisant une définition de tâche Amazon Elastic Container Service ou une opération d' RunTask API.
5. Un profil d'instance IAM attaché à une instance Amazon EC2

Il est recommandé de spécifier les informations d'identification dans l'ordre suivant lorsque vous utilisez l' CloudWatch agent.

1. Utilisez des rôles IAM pour les tâches si votre application utilise une définition de tâche Amazon Elastic Container Service ou une opération d' RunTask API.
2. Utilisez les rôles IAM si votre application s'exécute sur une instance Amazon EC2.
3. Utilisez le `common-config.toml` fichier CloudWatch d'agent pour spécifier le fichier d'informations d'identification. Ce fichier d'informations d'identification est le même que celui utilisé par les autres AWS SDK et le AWS CLI. Si vous utilisez déjà un fichier d'informations d'identification partagé, vous pouvez également l'utiliser à cette fin. Si vous les fournissez à l'aide du `common-config.toml` fichier de l' CloudWatch agent, vous vous assurez que l'agent consommera les informations d'identification modifiées lorsqu'elles expireront et seront remplacées sans que vous ayez à redémarrer l'agent.
4. Utilisez des variables d'environnement. La définition de variables d'environnement est utile si vous effectuez des travaux de développement sur un ordinateur autre qu'une instance Amazon EC2.

> **Note**
>
> Si vous envoyez des données télémétriques à un autre compte comme expliqué dans la sectionEnvoi de métriques, de journaux et de traces à un autre compte, l' CloudWatch agent utilise la chaîne de fournisseurs d'informations d'identification décrite dans cette section pour obtenir l'ensemble initial d'informations d'identification. Il utilise ensuite ces informations d'identification lorsqu'il assume le rôle IAM spécifié `role_arn` dans le fichier de configuration de l' CloudWatch agent.

Vérification de la signature du package de l' CloudWatch agent

Les fichiers de signature GPG sont inclus pour les packages CloudWatch d'agents sur les serveurs Linux. Vous pouvez utiliser une clé publique pour vérifier que le fichier de téléchargement de l'agent est l'archive originale non modifiée.

Pour Windows Server, vous pouvez utiliser le MSI pour vérifier la signature.

Pour les ordinateurs macOS, la signature est incluse dans le package de téléchargement de l'agent.

Pour retrouver le fichier SIGNATURE correct, consultez le tableau suivant. Pour chaque architecture et système d'exploitation, il existe un lien général, ainsi que des liens pour chaque région. Par exemple, pour Amazon Linux 2023, Amazon Linux 2 et l'architecture x86-64, les trois liens valides sont les suivants :

- `https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm.sig`
- `https://amazoncloudwatch-agent-us-east-1.s3.us-east-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`
- `https://amazoncloudwatch-agent-eu-central-1.s3.eu-central-1.amazonaws.com/amazon_linux/amd64/latest/amazon-cloudwatch-agent.rpm`

> **Note**
>
> Pour télécharger l' CloudWatch agent, votre connexion doit utiliser le protocole TLS 1.2 ou version ultérieure.

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	Amazon Linux 2023 et Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Centos	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/centos/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/centos/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/centos/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/amd64*	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/amd64/*

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
		/latest/ .rpm amazon-cloudwatch-agent	*latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	Debian	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/debian/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/debian/amd64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/debian/amd64/latest/* .deb.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/amd64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/amd64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/amd64/latest/* .deb.sig amazon-cloudwatch-agent
x86-64	Oracle	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/oracle_linux/amd64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/oracle_linux/amd64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/oracle_linux/amd64/latest/* .rpm.sig amazon-cloudwatch-agent
x86-64	macOS	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/amd64/latest/* .pkg amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/darwin/amd64/latest/ amazon-cloudwatch-agent .pkg .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/darwin/amd64/latest/* .pkg.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
x86-64	Windows	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/windows/amd64/latest/* .msi amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/windows/amd64/latest/ amazon-cloudwatch-agent .msi .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/windows/amd64/latest/* .msi.sig amazon-cloudwatch-agent
ARM64	Amazon Linux 2023 et Amazon Linux 2	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/amazon_linux/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/amazon_linux/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Architecture	Plateforme	Lien de téléchargement	Lien de fichier SIGNATURE
ARM64	Redhat	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/redhat/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/redhat/arm64/latest/* .rpm.sig amazon-cloudwatch-agent
ARM64	Ubuntu	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/arm64/latest/* .deb amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/ubuntu/arm64/latest/ amazon-cloudwatch-agent .deb.sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/ubuntu/arm64/latest/* .deb.sig amazon-cloudwatch-agent
ARM64	SUSE	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/arm64/latest/* .rpm amazon-cloudwatch-agent	https://amazoncloudwatch-agent.s3.amazonaws.com/suse/arm64/latest/ amazon-cloudwatch-agent .rpm .sig https://amazoncloudwatch-agent - *région* s.3. *région .amazonaws.com/suse/arm64/latest/* .rpm.sig amazon-cloudwatch-agent

Pour vérifier le package de CloudWatch l'agent sur un serveur Linux

1. Téléchargez la clé publique.

```
shell$ wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-
cloudwatch-agent.gpg
```

2. Importez la clé publique dans votre porte-clés.

```
shell$  gpg --import amazon-cloudwatch-agent.gpg
gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
gpg: Total number processed: 1
gpg: imported: 1 (RSA: 1)
```

   Notez la valeur de la clé ; vous en aurez besoin lors de l'étape suivante. Dans l'exemple précédent, la valeur de la clé est 3B789C72.

3. Vérifiez l'empreinte en exécutant la commande suivante, en remplaçant *key-value* (valeur clé) par la valeur de l'étape précédente :

```
shell$  gpg --fingerprint key-value
pub   2048R/3B789C72 2017-11-14
      Key fingerprint = 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid                  Amazon CloudWatch Agent
```

   La chaîne d'empreinte digitale doit être égale à ce qui suit :

   ```
   9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
   ```

   Si la chaîne d'empreintes ne correspond pas, n'installez pas l'agent. Contactez Amazon Web Services.

   Après avoir vérifié l'empreinte digitale, vous pouvez l'utiliser pour vérifier la signature du package de l' CloudWatch agent.

4. Téléchargez le fichier SIGNATURE de package avec wget. Pour déterminer le fichier SIGNATURE adéquat, consultez le tableau précédent.

```
wget Signature File Link
```

5. Pour vérifier la signature, exécutez gpg --verify.

```
shell$ gpg --verify signature-filename agent-download-filename
gpg: Signature made Wed 29 Nov 2017 03:00:59 PM PST using RSA key ID 3B789C72
gpg: Good signature from "Amazon CloudWatch Agent"
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Si le résultat inclut l'expression `BAD signature`, vérifiez si vous avez effectué la procédure correctement. Si vous continuez à obtenir cette réponse, contactez Amazon Web Services et évitez d'utiliser le fichier téléchargé.

Notez l'avertissement sur la confiance. Une clé est uniquement approuvée si vous ou une personne de confiance l'a signée. Cela ne signifie pas que la signature n'est pas valide, mais seulement que vous n'avez pas vérifié la clé publique.

Pour vérifier le package de l' CloudWatch agent sur un serveur exécutant Windows Server

1. Téléchargez et installez GnuPG pour Windows à partir de https://gnupg.org/download/. Lors de l'installation, incluez l'option Shell Extension (GpgEx).

   Vous pouvez effectuer les étapes restantes sous Windows PowerShell.

2. Téléchargez la clé publique.

   ```
   PS> wget https://amazoncloudwatch-agent.s3.amazonaws.com/assets/amazon-cloudwatch-agent.gpg -OutFile amazon-cloudwatch-agent.gpg
   ```

3. Importez la clé publique dans votre porte-clés.

   ```
   PS>  gpg --import amazon-cloudwatch-agent.gpg
   gpg: key 3B789C72: public key "Amazon CloudWatch Agent" imported
   gpg: Total number processed: 1
   gpg: imported: 1 (RSA: 1)
   ```

   Notez la valeur de la clé, car vous en aurez besoin lors de l'étape suivante. Dans l'exemple précédent, la valeur de la clé est `3B789C72`.

4. Vérifiez l'empreinte en exécutant la commande suivante, en remplaçant *`key-value`* (valeur clé) par la valeur de l'étape précédente :

```
PS>  gpg --fingerprint key-value
pub   rsa2048 2017-11-14 [SC]
      9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
uid           [ unknown] Amazon CloudWatch Agent
```

La chaîne d'empreinte digitale doit être égale à ce qui suit :

```
9376 16F3 450B 7D80 6CBD 9725 D581 6730 3B78 9C72
```

Si la chaîne d'empreintes ne correspond pas, n'installez pas l'agent. Contactez Amazon Web Services.

Après avoir vérifié l'empreinte digitale, vous pouvez l'utiliser pour vérifier la signature du package de l' CloudWatch agent.

5. Téléchargez le fichier SIGNATURE de package avec wget. Pour déterminer le fichier de signature approprié, consultez la section Liens de téléchargement de CloudWatch l'agent.
6. Pour vérifier la signature, exécutez gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Si le résultat inclut l'expression `BAD signature`, vérifiez si vous avez effectué la procédure correctement. Si vous continuez à obtenir cette réponse, contactez Amazon Web Services et évitez d'utiliser le fichier téléchargé.

Notez l'avertissement sur la confiance. Une clé est uniquement approuvée si vous ou une personne de confiance l'a signée. Cela ne signifie pas que la signature n'est pas valide, mais seulement que vous n'avez pas vérifié la clé publique.

Pour vérifier le package de CloudWatch l'agent sur un ordinateur macOS

- Il existe deux méthodes de vérification de signature sur macOS.
  - Vérifiez l'empreinte digitale en exécutant la commande suivante.

```
pkgutil --check-signature amazon-cloudwatch-agent.pkg
```

Le résultat doit ressembler à ce qui suit.

```
Package "amazon-cloudwatch-agent.pkg":
       Status: signed by a developer certificate issued by Apple for
 distribution
       Signed with a trusted timestamp on: 2020-10-02 18:13:24 +0000
       Certificate Chain:
       1. Developer ID Installer: AMZN Mobile LLC (94KV3E626L)
       Expires: 2024-10-18 22:31:30 +0000
       SHA256 Fingerprint:
       81 B4 6F AF 1C CA E1 E8 3C 6F FB 9E 52 5E 84 02 6E 7F 17 21 8E FB
       0C 40 79 13 66 8D 9F 1F 10 1C

 ------------------------------------------------------------------------
       2. Developer ID Certification Authority
       Expires: 2027-02-01 22:12:15 +0000
       SHA256 Fingerprint:
       7A FC 9D 01 A6 2F 03 A2 DE 96 37 93 6D 4A FE 68 09 0D 2D E1 8D 03
       F2 9C 88 CF B0 B1 BA 63 58 7F

 ------------------------------------------------------------------------
       3. Apple Root CA
       Expires: 2035-02-09 21:40:36 +0000
       SHA256 Fingerprint:
       B0 B1 73 0E CB C7 FF 45 05 14 2C 49 F1 29 5E 6E DA 6B CA ED 7E 2C
       68 C5 BE 91 B5 A1 10 01 F0 24
```

- Ou, téléchargez et utilisez le fichier .sig. Pour utiliser cette méthode, procédez comme suit.
  - Installez l'application GPG sur votre hôte macOS en saisissant la commande suivante.

    ```
    brew install GnuPG
    ```

- Téléchargez le fichier signature de package avec curl. Pour déterminer le fichier de signature approprié, consultez la section Liens de téléchargement de CloudWatch l'agent.
- Pour vérifier la signature, exécutez gpg --verify.

```
PS> gpg --verify sig-filename agent-download-filename
gpg: Signature made 11/29/17 23:00:45 Coordinated Universal Time
```

```
gpg:                using RSA key D58167303B789C72
gpg: Good signature from "Amazon CloudWatch Agent" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: 9376 16F3 450B 7D80 6CBD  9725 D581 6730 3B78 9C72
```

Si le résultat inclut l'expression `BAD signature`, vérifiez si vous avez effectué la procédure correctement. Si vous continuez à obtenir cette réponse, contactez Amazon Web Services et évitez d'utiliser le fichier téléchargé.

Notez l'avertissement sur la confiance. Une clé est uniquement approuvée si vous ou une personne de confiance l'a signée. Cela ne signifie pas que la signature n'est pas valide, mais seulement que vous n'avez pas vérifié la clé publique.

Création du fichier de configuration de CloudWatch l'agent

Avant d'exécuter l' CloudWatch agent sur un serveur, vous devez créer un ou plusieurs fichiers de configuration de CloudWatch l'agent.

Le fichier de configuration d'agent est un fichier JSON qui spécifie les métriques, les journaux et les traces que l'agent doit collecter, notamment les métriques personnalisées. Vous pouvez le créer à l'aide de l'assistant ou en le créant vous-même à partir de zéro. Vous pouvez également utiliser l'assistant pour créer le fichier de configuration, puis le modifier manuellement. Si vous créez ou modifiez le fichier manuellement, le processus est plus complexe, mais vous avez plus de contrôle sur les métriques collectées, et vous pouvez spécifier des métriques non disponibles via l'assistant.

Chaque fois que vous modifiez le fichier de configuration d'agent, vous devez ensuite redémarrer l'agent pour que les modifications prennent effet. Pour redémarrer l'agent, suivez les instructions de la section Démarrez l' CloudWatch agent.

Une fois que vous avez créé un fichier de configuration, vous pouvez l'enregistrer manuellement en tant que fichier JSON, puis utiliser ce fichier lors de l'installation de l'agent sur vos serveurs. Vous pouvez également le stocker dans le Parameter Store du Systems Manager si vous allez utiliser Systems Manager lorsque vous installez l'agent sur les serveurs.

L' CloudWatch agent prend en charge l'utilisation de plusieurs fichiers de configuration. Pour plus d’informations, consultez Plusieurs fichiers CloudWatch de configuration d'agents.

Les métriques, les journaux et les traces collectés par l' CloudWatch agent entraînent des frais. Pour plus d'informations sur les tarifs, consultez Amazon CloudWatch Pricing.

Table des matières

Créez le fichier de configuration de CloudWatch l'agent à l'aide de l'assistant

L'assistant du fichier de configuration de l'agent pose une série de questions pour vous aider à configurer l' CloudWatch agent en fonction de vos besoins. `amazon-cloudwatch-agent-config-wizard`

Informations d'identification requises

L'assistant peut détecter automatiquement les informations d'identification et AWS la région à utiliser si vous disposez des AWS informations d'identification et des fichiers de configuration avant de démarrer l'assistant. Pour plus d'informations sur ces fichiers, consultez Configuration and Credential Files (Fichiers de configuration et d'informations d'identification) dans le Guide de l'utilisateur AWS Systems Manager .

Dans le fichier AWS d'informations d'identification, l'assistant vérifie les informations d'identification par défaut et recherche également une `AmazonCloudWatchAgent` section telle que la suivante :

```
[AmazonCloudWatchAgent]
aws_access_key_id = my_access_key
aws_secret_access_key = my_secret_key
```

L'Assistant affiche les informations d'identification par défaut, les informations d'identification à partir de la `AmazonCloudWatchAgent` et une option `Others`. Vous pouvez sélectionner les informations d'identification à utiliser. Si vous choisissez `Others`, vous pouvez saisir les informations d'identification.

Pour *`my_access_key`* et *`my_secret_key`*, utilisez les clés provenant de l'utilisateur IAM qui possède les autorisations pour écrire sur le Parameter Store du Systems Manager. Pour plus d'informations sur les utilisateurs IAM nécessaires à l' CloudWatch agent, consultezCréation d'utilisateurs IAM à utiliser avec l' CloudWatch agent sur des serveurs locaux.

Dans le fichier AWS de configuration, vous pouvez spécifier la région à laquelle l'agent envoie les métriques si elle est différente de la `[default]` section. Par défaut, les métriques sont publiées dans la région où se trouve l'instance Amazon EC2. Si les métriques doivent être publiées dans une autre région, indiquez-la ici. Dans l'exemple suivant, les métriques sont publiées dans la région `us-west-1`.

```
[AmazonCloudWatchAgent]
region = us-west-1
```

Exécutez l'assistant de configuration de l' CloudWatch agent

Pour créer le fichier de configuration de CloudWatch l'agent

1. Démarrez l'assistant de configuration de l' CloudWatch agent en saisissant les informations suivantes :

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
   ```

   Sur un serveur exécutant Windows Server, exécutez les commandes suivantes pour lancer l'assistant :

   ```
   cd "C:\Program Files\Amazon\AmazonCloudWatchAgent"
   ```

   ```
   .\amazon-cloudwatch-agent-config-wizard.exe
   ```

2. Répondez aux questions pour personnaliser le fichier de configuration pour votre serveur.
3. Si vous stockez le fichier de configuration localement, le fichier de configuration `config.json` est stocké dans `/opt/aws/amazon-cloudwatch-agent/bin/` sur les serveurs Linux et dans `C:\Program Files\Amazon\AmazonCloudWatchAgent` sous Windows Server. Vous pouvez ensuite copier ce fichier vers d'autres serveurs sur lesquels vous souhaitez installer l'agent.

   Si vous allez utiliser Systems Manager pour installer et configurer l'agent, assurez-vous de répondre Yes (Oui) lorsque vous êtes invité à choisir si le fichier doit être stocké dans le Parameter Store du Systems Manager. Vous pouvez également choisir de stocker le fichier dans Parameter Store même si vous n'utilisez pas l'agent SSM pour installer l' CloudWatch agent. Pour être en mesure de stocker le fichier dans le Parameter Store, vous devez utiliser un rôle

IAM avec les autorisations suffisantes. Pour plus d'informations, consultez Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

CloudWatch ensembles de mesures prédéfinis par l'agent

L'Assistant est configuré avec des ensembles de métriques prédéfinis, dotés de différents niveaux de détail. Ces ensembles de métriques sont affichés dans les tableaux suivants. Pour plus d'informations sur ces métriques, consultez Métriques collectées par l' CloudWatchagent.

> **Note**
>
> Le Parameter Store prend en charge les paramètres des niveaux Standard et Avancé. Ces niveaux de paramètres ne sont pas liés aux niveaux de base, standard et avancé des détails des métriques décrits dans ces tableaux.

Instances Amazon EC2 exécutant Linux

Niveau de détail	Métriques incluses
Base	Mem : mem_used_percent Disue : disk_used_percent Les métriques `disk` telles que `disk_used_percent` ont une dimension pour `Partition`, ce qui signifie que le nombre de métriques personnalisées générées dépend du nombre de partitions associées à votre instance. Le nombre de partitions de disque dont vous disposez dépend de l'AMI que vous utilisez et du nombre de volumes Amazon EBS que vous attachez au serveur.
Standard	UC : `cpu_usage_idle`, `cpu_usage_iowait`, `cpu_usage_user`, `cpu_usage_system` Disque : `disk_used_percent`, `disk_inodes_free` Diskio : `diskio_io_time` Mem : `mem_used_percent`

Niveau de détail	Métriques incluses
	Swap : `swap_used_percent`
Advanced (Avancé)	UC : `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_user` , `cpu_usage_system` Disque : `disk_used_percent` , `disk_inodes_free` Diskio : `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem : `mem_used_percent` Netstat : `netstat_tcp_established` , `netstat_tcp_time_wait` Swap : `swap_used_percent`

Serveurs sur site exécutant Linux

Niveau de détail	Métriques incluses
Base	Disque : `disk_used_percent` Diskio : `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem : `mem_used_percent` Net : `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap : `swap_used_percent`
Standard	UC : `cpu_usage_idle` , `cpu_usage_iowait` Disque : `disk_used_percent` , `disk_inodes_free` Diskio : `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads`

Niveau de détail	Métriques incluses
	Mem : `mem_used_percent` Net : `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Swap : `swap_used_percent`
Advanced (Avancé)	UC : `cpu_usage_guest` , `cpu_usage_idle` , `cpu_usage_iowait` , `cpu_usage_steal` , `cpu_usage_user` , `cpu_usage_system` Disque : `disk_used_percent` , `disk_inodes_free` Diskio : `diskio_io_time` , `diskio_write_bytes` , `diskio_read_bytes` , `diskio_writes` , `diskio_reads` Mem : `mem_used_percent` Net : `net_bytes_sent` , `net_bytes_recv` , `net_packets_sent` , `net_packets_recv` Netstat : `netstat_tcp_established` , `netstat_tcp_time_wait` Swap : `swap_used_percent`

Instances Amazon EC2 exécutant Windows Server

> **Note**
>
> Les noms des métriques répertoriés dans ce tableau indiquent comment les métriques apparaissent lorsqu'elles sont affichées dans la console. Le nom de la métrique peut ne pas inclure le premier mot. Par exemple, le nom de la métrique pour `LogicalDisk % Free Space` est simplement `% Free Space`.

Niveau de détail	Métriques incluses
Base	Mémoire : `Memory % Committed Bytes In Use`

Niveau de détail	Métriques incluses
	LogicalDisk: `LogicalDisk % Free Space`
Standard	Mémoire : `Memory % Committed Bytes In Use` Pagination : `Paging File % Usage` Processeur : `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` PhysicalDisk: `PhysicalDisk % Disk Time` LogicalDisk: `LogicalDisk % Free Space`
Advanced (Avancé)	Mémoire : `Memory % Committed Bytes In Use` Pagination : `Paging File % Usage` Processeur : `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` TCP : `TCPv4 Connections Established` , `TCPv6 Connections Established`

Serveur sur site exécutant Windows Server

> **Note**
>
> Les noms des métriques répertoriés dans ce tableau indiquent comment les métriques apparaissent lorsqu'elles sont affichées dans la console. Le nom de la métrique peut ne pas

inclure le premier mot. Par exemple, le nom de la métrique pour `LogicalDisk % Free Space` est simplement `% Free Space`.

Niveau de détail	Métriques incluses
Base	Pagination : `Paging File % Usage` Processeur : `Processor % Processor Time` LogicalDisk:`LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Mémoire : `Memory % Committed Bytes In Use` Interface réseau : `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`
Standard	Pagination : `Paging File % Usage` Processeur : `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Mémoire : `Memory % Committed Bytes In Use` Interface réseau : `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec`

Niveau de détail	Métriques incluses
Advanced (Avancé)	Pagination :`Paging File % Usage` Processeur : `Processor % Processor Time`, `Processor % Idle Time`, `Processor % Interrupt Time`, `Processor % User Time` LogicalDisk: `LogicalDisk % Free Space` PhysicalDisk: `PhysicalDisk % Disk Time` , `PhysicalDisk Disk Write Bytes/sec` , `PhysicalDisk Disk Read Bytes/sec` , `PhysicalDisk Disk Writes/sec` , `PhysicalDisk Disk Reads/sec` Mémoire : `Memory % Committed Bytes In Use` Interface réseau : `Network Interface Bytes Sent/sec`, `Network Interface Bytes Received/sec` , `Network Interface Packets Sent/sec`, `Network Interface Packets Received/sec` TCP : `TCPv4 Connections Established` , `TCPv6 Connections Established`

Création ou modification manuelle du fichier de configuration de CloudWatch l'agent

Le fichier de configuration de l' CloudWatch agent est un fichier JSON composé de quatre sections `agent` `metrics`,`logs`,`traces`, et décrites comme suit :

- La section `agent` inclut des champs pour la configuration générale de l'agent.
- La `metrics` section spécifie les métriques personnalisées pour la collecte et la publication sur CloudWatch. Si vous utilisez l'agent uniquement pour collecter les journaux, vous pouvez omettre la section `metrics` du fichier.
- La `logs` section indique quels fichiers journaux sont publiés dans CloudWatch Logs. Ceci peut inclure des événements du journal des événements Windows si le serveur exécute Windows Server.
- La `traces` section indique les sources des traces collectées et envoyées AWS X-Ray.

Les sections suivantes décrivent la structure et les champs de ce fichier JSON. Vous pouvez également consulter la définition de schéma de ce fichier de configuration. La définition de schéma se trouve au niveau de *installation-directory*/doc/amazon-cloudwatch-agent-schema.json sur les serveurs Linux et au niveau de *installation-directory*/amazon-cloudwatch-agent-schema.json sur les serveurs exécutant Windows Server.

Si vous créez ou modifiez manuellement la configuration d'agent , vous pouvez lui donner n'importe quel nom. Pour faciliter les éventuels dépannages, nous vous recommandons de le nommer /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json sur un serveur Linux et $Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json sur les serveurs exécutant Windows Server. Vous pouvez ensuite copier ce fichier vers d'autres serveurs sur lesquels vous souhaitez installer l'agent une fois que vous avez créé le fichier.

> **Note**
>
> Les métriques, les journaux et les traces collectés par l' CloudWatch agent entraînent des frais. Pour plus d'informations sur les tarifs, consultez Amazon CloudWatch Pricing.

CloudWatch fichier de configuration de l'agent : section Agent

La section agent peut inclure les champs suivants. L'Assistant ne crée pas de section agent. Au lieu de cela, l'Assistant l'omet et il utilise les valeurs par défaut pour tous les champs de cette section.

- metrics_collection_interval – Facultatif. Indique la fréquence à laquelle toutes les métriques spécifiées dans ce fichier de configuration doivent être collectées. Vous pouvez remplacer cette valeur pour des types spécifiques de métriques.

  Cette valeur est spécifiée en secondes. Par exemple, en spécifiant 10, les métriques sont collectées toutes les 10 secondes, et en le fixant à 300, les métriques sont collectées toutes les 5 minutes.

  Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.

  La valeur par défaut est 60.
- region— Spécifie la région à utiliser pour le CloudWatch point de terminaison lorsqu'une instance Amazon EC2 est surveillée. Les métriques collectées sont envoyées à cette région, par exemple

`us-west-1`. Si vous ne spécifiez pas ce champ, l'agent envoie les métriques à la région dans laquelle se trouve l'instance Amazon EC2.

Si vous surveillez un serveur sur site, ce champ n'est pas utilisé et l'agent lit la région à partir du profil `AmazonCloudWatchAgent` du fichier de configuration AWS .

- `credentials`— Spécifie un rôle IAM à utiliser lors de l'envoi de métriques, de journaux et de traces vers un autre AWS compte. S'il est spécifié, ce champ contient un paramètre, `role_arn`.
  - `role_arn` – Spécifie l'Amazon Resource Name (ARN) d'un rôle IAM à utiliser pour l'authentification lors de l'envoi des métriques, des journaux et des traces à un compte AWS différent. Pour plus d'informations, consultez Envoi de métriques, de journaux et de traces à un autre compte.
- `debug` – Facultatif. Spécifie l'exécution de l' CloudWatch agent avec les messages du journal de débogage. La valeur par défaut est `false`.
- `aws_sdk_log_level` – Facultatif. Pris en charge uniquement dans les versions 1.247350.0 et ultérieures de l'agent. CloudWatch

  Vous pouvez spécifier ce champ pour que l'agent enregistre les points de terminaison du AWS SDK. La valeur de ce champ peut inclure une ou plusieurs des options suivantes. Séparez les différentes options avec le caractère `|`.
  - `LogDebug`
  - `LogDebugWithSigning`
  - `LogDebugWithHTTPBody`
  - `LogDebugRequestRetries`
  - `LogDebugWithEventStreamBody`

  Pour plus d'informations sur ces options, consultez LogLevelType.
- `logfile`— Spécifie l'emplacement où l' CloudWatch agent écrit les messages de journal. Si vous spécifiez une chaîne vide, le journal est acheminé vers stderr. Si vous ne spécifiez pas cette option, les emplacements par défaut sont les suivants :
  - Linux : `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log`
  - Windows Server : `c:\\ProgramData\\Amazon\\CloudWatchAgent\\Logs\\amazon-cloudwatch-agent.log`

  L' CloudWatch agent fait automatiquement pivoter le fichier journal qu'il crée. Un fichier journal fait l'objet d'une rotation lorsqu'il atteint 100 Mo de taille. L'agent conserve pendant une durée

de sept jours les fichiers journaux ayant fait l'objet d'une rotation et il peut conserver jusqu'à cinq fichiers journaux de sauvegarde qui ont fait l'objet d'une rotation. Les fichiers journaux sauvegardés disposent d'un horodatage ajouté à leur nom de fichier. L'horodatage indique la date et l'heure auxquelles le fichier a fait l'objet d'une rotation : par exemple, `amazon-cloudwatch-agent-2018-06-08T21-01-50.247.log.gz`.

- `omit_hostname` – Facultatif. Par défaut, le nom d'hôte est publié sous la forme de la dimension des métriques qui sont collectées par l'agent, sauf si vous utilisez le champ `append_dimensions` dans la section `metrics`. Définissez `omit_hostname` sur `true` pour éviter que le nom d'hôte soit publié en tant que dimension même si vous n'utilisez pas `append_dimensions`. La valeur par défaut est `false`.
- `run_as_user` – Facultatif. Spécifie un utilisateur à utiliser pour exécuter l' CloudWatch agent. Si vous ne spécifiez pas ce paramètre, l'utilisateur racine est utilisé. Cette option est valide uniquement sur les serveurs Linux.

  Si vous spécifiez cette option, l'utilisateur doit exister avant de démarrer l' CloudWatch agent. Pour plus d'informations, consultez Exécution de l' CloudWatch agent en tant qu'utilisateur différent.
- `user_agent` – Facultatif. Spécifie la `user-agent` chaîne utilisée par l' CloudWatch agent lorsqu'il effectue des appels d'API au CloudWatch backend. La valeur par défaut est une chaîne composée de la version de l'agent, de la version du langage de programmation Go utilisé pour compiler l'agent, du système d'exploitation et de l'architecture d'exécution, du temps de génération et des plug-ins activés.
- `usage_data` : facultatif. Par défaut, l' CloudWatch agent envoie des données de santé et de performance le concernant à CloudWatch chaque fois qu'il publie des métriques ou se connecte à CloudWatch. Ces données ne vous coûtent rien. Vous pouvez empêcher l'agent d'envoyer ces données en spécifiant `false` pour `usage_data`. Si vous omettez ce paramètre, la valeur par défaut `true` est utilisée et l'agent envoie des données de santé et de performance.

  Si vous définissez cette valeur sur `false`, vous devez arrêter et redémarrer l'agent pour qu'elle prenne effet.

Voici un exemple de valeur pour une section `agent`.

```
"agent": {
   "metrics_collection_interval": 60,
   "region": "us-west-1",
   "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log",
   "debug": false,
```

```
   "run_as_user": "cwagent"
  }
```

CloudWatch fichier de configuration de l'agent : section Metrics

Champs communs à Linux et Windows

Sur les serveurs exécutant Linux ou Windows Server, la section `metrics` comprend les champs suivants :

- `namespace` – Facultatif. Espace de noms à utiliser pour les métriques collectées par l'agent. La valeur par défaut est `CWAgent`. La longueur maximale est de 255 caractères. Voici un exemple :

  ```
  {
    "metrics": {
      "namespace": "Development/Product1Metrics",
     ......
     },
  }
  ```

- `append_dimensions` – Facultatif. Ajoute des dimensions métriques Amazon EC2 à toutes les métriques collectées par l'agent. Cela provoque également l'agent de ne pas publier le nom d'hôte en tant que dimension.

  Les seules paires clé-valeur prises en charge pour `append_dimensions` sont affichées dans la liste suivante. Toutes les autres paires clé-valeur sont ignorées. L'agent prend en charge ces paires clé-valeur exactement comme indiqué dans la liste suivante. Vous ne pouvez pas modifier les valeurs clés pour publier des noms de dimension différents pour celles-ci.

  - `"ImageId":"${aws:ImageId}"` définit l'ID d'AMI de l'instance comme valeur de la dimension `ImageId`.
  - `"InstanceId":"${aws:InstanceId}"` définit l'ID d'instance de l'instance comme valeur de la dimension `InstanceId`.
  - `"InstanceType":"${aws:InstanceType}"` définit le type d'instance de l'instance comme valeur de la dimension `InstanceType`.
  - `"AutoScalingGroupName":"${aws:AutoScalingGroupName}"` définit le nom du groupe Auto Scaling de l'instance comme valeur de la dimension `AutoScalingGroupName`.

  Si vous souhaitez ajouter des dimensions aux métriques avec des paires clé-valeur arbitraires, utilisez le paramètre `append_dimensions` dans le champ pour ce type de métrique particulier.

Si vous spécifiez une valeur qui dépend des métadonnées Amazon EC2 et que vous utilisez des proxys, vous devez vous assurer que le serveur peut accéder au point de terminaison pour Amazon EC2. Pour plus d'informations sur ces points de terminaison, consultez la rubrique Amazon Elastic Compute Cloud (Amazon EC2) du document Référence générale d'Amazon Web Services.

- `aggregation_dimensions` – Facultatif. Spécifie les dimensions sur lesquelles les métriques collectées vont être regroupées. Par exemple, si vous regroupez les métriques sur la dimension `AutoScalingGroupName`, les métriques de toutes les instances dans chaque groupe Auto Scaling sont regroupées et peuvent être affichées dans leur ensemble.

  Vous pouvez regrouper des métriques sur une ou plusieurs dimensions. Par exemple, spécifier `[["InstanceId"], ["InstanceType"], ["InstanceId","InstanceType"]]` regroupe les métriques pour l'ID d'instance individuellement, le type d'instance individuellement et pour la combinaison des deux dimensions.

  Vous pouvez également spécifier `[]` pour regrouper toutes les métriques dans une collection, indépendamment de toutes les dimensions.

- `endpoint_override` – Spécifie un point de terminaison FIPS ou un lien privé à utiliser en tant que point de terminaison auquel l'agent envoie les métriques. Cette spécification et la définition d'un lien privé vous permettent d'envoyer les métriques à un point de terminaison Amazon VPC. Pour de plus amples informations, consultez Qu'est-ce qu'Amazon VPC ?.

  La valeur de `endpoint_override` doit être une chaîne qui est une URL.

  Par exemple, la partie suivante de la section Métriques du fichier de configuration définit l'agent pour qu'il utilise un point de terminaison d'un VPC lors de l'envoi de métriques.

```
{
  "metrics": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.monitoring.us-
east-1.vpce.amazonaws.com",
   ......
   },
}
```

- `metrics_collected` – Obligatoire Spécifie les métriques qui doivent être collectées, y compris les métriques personnalisées collectées via `StatsD` ou `collectd`. Cette section inclut plusieurs sous-sections.

Le contenu de la section `metrics_collected` varie selon que ce fichier de configuration est pour un serveur exécutant Linux ou Windows Server.

- `force_flush_interval` – Spécifie en secondes la durée maximale pendant laquelle les métriques demeurent dans la mémoire tampon avant d'être envoyées au serveur. Quelle que soit la configuration, si la taille des métriques dans la mémoire tampon atteint 1 Mo ou le nombre de 1000 différentes métriques, les métriques sont immédiatement envoyées au serveur.

  La valeur par défaut est 60.
- `credentials` – Spécifie un rôle IAM à utiliser lors de l'envoi de métriques à un compte différent. S'il est spécifié, ce champ contient un paramètre, `role_arn`.
  - `role_arn` – Spécifie l'ARN d'un rôle IAM à utiliser pour l'authentification lors de l'envoi de métriques à un compte différent. Pour de plus amples informations, consultez Envoi de métriques, de journaux et de traces à un autre compte. Si elle est spécifiée ici, cette valeur remplace le `role_arn` spécifié dans la section `agent` du fichier de configuration, le cas échéant.

Section Linux

Sur les serveurs exécutant Linux, la section `metrics_collected` du fichier de configuration peut également contenir les champs suivants.

La plupart de ces champs peuvent inclure des sections `measurement` qui répertorient les métriques que vous souhaitez collecter pour cette ressource. Ces sections `measurement` peuvent spécifier soit le nom complet de la métrique, par exemple `swap_used`, soit uniquement la partie du nom de la métrique qui sera ajoutée au type de ressource. Par exemple, la spécification de `reads` dans la section `measurement` de la section `diskio` entraîne la collecte de la métrique `diskio_reads`.

- `collectd` – Facultatif. Spécifie que vous souhaitez récupérer les métriques personnalisées à l'aide du protocole `collectd`. Vous utilisez `collectd` un logiciel pour envoyer les métriques à l' CloudWatch agent. Pour plus d'informations sur les options de configuration de collectd, consultez Récupération de métriques personnalisées avec collectd.
- `cpu` – Facultatif. Indique que des métriques d'UC doivent être collectées. Cette section est valable uniquement pour les instances Linux. Vous devez inclure au moins un des champs `totalcpu` et `resources` pour toutes les métriques d'UC à collecter. Cette section peut inclure les champs suivants :

- `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
- `resources` : facultatif. Spécifiez ce champ avec la valeur `*` pour déclencher la collecte des métriques par uc. La seule valeur autorisée est `*`.
- `totalcpu` – Facultatif. Indique s'il est nécessaire de signaler les métriques agrégées entre tous les cœurs de cpu. Par défaut, la valeur est true.
- `measurement` – Spécifie l'éventail de métriques d'UC à collecter. Les valeurs possibles sont `time_active`, `time_guest`, `time_guest_nice`, `time_idle`, `time_iowait`, `time_irq`, `time_nice`, `time_softirq`, `time_steal`, `time_system`, `time_user`, `usage_active`, `usage_guest`, `usage_guest_nice`, `usage_idle`, `usage_iowait`, `usage_irq`, `usage_nice`, `usage_softirq`, `usage_steal`, `usage_system` et `usage_user`. Ce champ est obligatoire si vous incluez `cpu`.

  Par défaut, l'unité pour les métriques `cpu_usage_*` est `Percent` et les métriques `cpu_time_*` n'ont pas d'unité.

  Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :

  - `rename` – Spécifie un nom différent pour cette métrique.
  - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
- `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques cpu, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

  Cette valeur est spécifiée en secondes. Par exemple, en spécifiant 10, les métriques sont collectées toutes les 10 secondes, et en le fixant à 300, les métriques sont collectées toutes les 5 minutes.

Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.

- `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques cpu. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` global qui est utilisé pour tous les types de métriques collectées par l'agent.

- `disk` – Facultatif. Indique les métriques de disque qui doivent être collectées. Cette section est valable uniquement pour les instances Linux. Cette section peut inclure les champs suivants :
  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `resources` – Facultatif. Spécifie un ensemble de points de montage de disque. Ce champ permet uniquement CloudWatch de collecter des métriques à partir des points de montage répertoriés. Vous pouvez spécifier * comme valeur pour la collecter des métriques de tous les points de montage. La valeur par défaut consiste à collecter les métriques de tous points de montage.
  - `measurement` – Spécifie l'éventail de métriques de disque à collecter. Les valeurs possibles sont : `free`, `total`, `used`, `used_percent`, `inodes_free`, `inodes_used` et `inodes_total`. Ce champ est obligatoire si vous incluez `disk`.

    > **Note**
    >
    > Les métriques `disk` ont une dimension pour `Partition`, ce qui signifie que le nombre de métriques personnalisées générées dépend du nombre de partitions associées à votre instance. Le nombre de partitions de disque dont vous disposez dépend de l'AMI que vous utilisez et du nombre de volumes Amazon EBS que vous attachez au serveur.

Pour voir les unités par défaut de chaque métrique `disk`, consultez Métriques collectées par l'CloudWatchagent sur les instances Linux et macOS.

Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :

- `rename` – Spécifie un nom différent pour cette métrique.
- `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.

- `ignore_file_system_types` – Spécifie les types de systèmes de fichiers à exclure lors de la collecte des métriques de disque. Les valeurs valides sont `sysfs`, `devtmpfs` et ainsi de suite.
- `drop_device` – Si vous définissez ce paramètre sur `true`, `Device` n'est pas inclus comme dimension pour les métriques de disque.

  Empêcher `Device` d'être utilisé comme dimension peut être utile sur les instances qui font appel au système Nitro, car sur ces instances, les noms d'appareil changent pour chaque montage de disque lors du redémarrage de l'instance. Cela peut entraîner des données incohérentes dans vos métriques et faire en sorte que les alertes basées sur ces métriques passent à l'état `INSUFFICIENT DATA`.

  La valeur par défaut est `false`.
- `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques disque, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

  Cette valeur est spécifiée en secondes.

  Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour de plus amples informations, consultez Métriques haute résolution.
- `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques de disque. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` qui est utilisé pour tous les types de métriques collectées par l'agent.

- `diskio` – Facultatif. Spécifie les métriques disque i/o qui doivent être collectées. Cette section est valable uniquement pour les instances Linux. Cette section peut inclure les champs suivants :
  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `resources` : facultatif. Si vous spécifiez un ensemble d'appareils, CloudWatch collecte les métriques uniquement à partir de ces appareils. Dans le cas contraire, les métriques de tous les appareils sont collectées. Vous pouvez aussi spécifier * comme valeur pour la collecter des métriques de tous les appareils.
  - `measurement` – Spécifie l'éventail de métriques diskio à collecter. Les valeurs possibles sont : `reads`, `writes`, `read_bytes`, `write_bytes`, `read_time`, `write_time`, `io_time` et `iops_in_progress`. Ce champ est obligatoire si vous incluez `diskio`.

    Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :
    - `rename` – Spécifie un nom différent pour cette métrique.
    - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
  - `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques diskio, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

    Cette valeur est spécifiée en secondes.

    Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.
  - `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques diskio. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` qui est utilisé pour tous les types de métriques collectées par l'agent.

- `swap` – Facultatif. Indique que les métriques de mémoire swap doivent être collectées. Cette section est valable uniquement pour les instances Linux. Cette section peut inclure les champs suivants :
  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `measurement` – Spécifie l'éventail de métriques de swap à collecter. Les valeurs possibles sont `free`, `used` et `used_percent`. Ce champ est obligatoire si vous incluez `swap`.

    Pour voir les unités par défaut de chaque métrique `swap`, consultez Métriques collectées par l' CloudWatchagent sur les instances Linux et macOS.

    Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :
    - `rename` – Spécifie un nom différent pour cette métrique.
    - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
  - `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques swap, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

    Cette valeur est spécifiée en secondes.

    Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.
  - `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques swap. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` global qui est utilisé pour tous les types de métriques collectées par l'agent. Elle est collectée sous la forme d'une métrique haute résolution.

- `mem` – Facultatif. Indique que les métriques de mémoire doivent être collectées. Cette section est valable uniquement pour les instances Linux. Cette section peut inclure les champs suivants :
  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `measurement` – Spécifie l'éventail de métriques de mémoire à collecter. Les valeurs possibles sont : `active`, `available`, `available_percent`, `buffered`, `cached`, `free`, `inactive`, `total`, `used` et `used_percent`. Ce champ est obligatoire si vous incluez `mem`.

    Pour voir les unités par défaut de chaque métrique `mem`, consultez Métriques collectées par l'CloudWatchagent sur les instances Linux et macOS.

    Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :
    - `rename` – Spécifie un nom différent pour cette métrique.
    - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
  - `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques mem, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

    Cette valeur est spécifiée en secondes.

    Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.
  - `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques mem. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` global qui est utilisé pour tous les types de métriques collectées par l'agent.

- `net` – Facultatif. Indique que les métriques networking doivent être collectées. Cette section est valable uniquement pour les instances Linux. Cette section peut inclure les champs suivants :
  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `resources` : facultatif. Si vous spécifiez un ensemble d'interfaces réseau, CloudWatch collecte les métriques uniquement à partir de ces interfaces. Dans le cas contraire, les métriques de tous les appareils sont collectées. Vous pouvez aussi spécifier `*` comme valeur pour collecter des métriques de toutes les interfaces.
  - `measurement` – Spécifie l'éventail de métriques de réseautage à collecter. Les valeurs possibles sont : `bytes_sent`, `bytes_recv`, `drop_in`, `drop_out`, `err_in`, `err_out`, `packets_sent` et `packets_recv`. Ce champ est obligatoire si vous incluez `net`.

    Pour voir les unités par défaut de chaque métrique `net`, consultez Métriques collectées par l' CloudWatchagent sur les instances Linux et macOS.

    Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :
    - `rename` – Spécifie un nom différent pour cette métrique.
    - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
  - `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques net, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

    Cette valeur est spécifiée en secondes. Par exemple, en spécifiant 10, les métriques sont collectées toutes les 10 secondes, et en le fixant à 300, les métriques sont collectées toutes les 5 minutes.

Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.

- `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques net. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` qui est utilisé pour tous les types de métriques collectées par l'agent.

- `netstat` – Facultatif. Indique que l'état de connexion TCP et les métriques de connexion UDP doivent être collectés. Cette section est valable uniquement pour les instances Linux. Cette section peut inclure les champs suivants :
  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `measurement` – Spécifie l'éventail de métriques netstat à collecter. Les valeurs possibles sont `tcp_close`, `tcp_close_wait`, `tcp_closing`, `tcp_established`, `tcp_fin_wait1`, `tcp_fin_wait2`, `tcp_last_ack`, `tcp_listen`, `tcp_none`, `tcp_syn_sent`, `tcp_syn_recv`, `tcp_time_wait` et `udp_socket`. Ce champ est obligatoire si vous incluez `netstat`.

    Pour voir les unités par défaut de chaque métrique `netstat`, consultez Métriques collectées par l' CloudWatchagent sur les instances Linux et macOS.

    Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :
    - `rename` – Spécifie un nom différent pour cette métrique.
    - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
  - `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques netstat, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

Cette valeur est spécifiée en secondes.

Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.

- `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques netstat. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` qui est utilisé pour tous les types de métriques collectées par l'agent.

- `processes` – Facultatif. Indique que les métriques processes doivent être collectées. Cette section est valable uniquement pour les instances Linux. Cette section peut inclure les champs suivants :
  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `measurement` – Spécifie l'éventail de métriques de processus à collecter. Les valeurs possibles sont : `blocked`, `dead`, `idle`, `paging`, `running`, `sleeping`, `stopped`, `total`, `total_threads`, `wait` et `zombies`. Ce champ est obligatoire si vous incluez `processes`.

    Pour toutes les métriques `processes`, l'unité par défaut est `None`.

    Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :
    - `rename` – Spécifie un nom différent pour cette métrique.
    - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
  - `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques processes, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

Cette valeur est spécifiée en secondes. Par exemple, en spécifiant 10, les métriques sont collectées toutes les 10 secondes, et en le fixant à 300, les métriques sont collectées toutes les 5 minutes.

Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour de plus amples informations, consultez Métriques haute résolution.

- `append_dimensions` – Facultatif. Autres dimensions à utiliser pour seulement les métriques process. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` qui est utilisé pour tous les types de métriques collectées par l'agent.

- `nvidia_gpu` – Facultatif. Indique les métriques GPU NVIDIA qui doivent être collectées. Cette section est valable uniquement pour les instances Linux sur des hôtes configurés avec un accélérateur GPU NVIDIA et sur lesquelles l'interface de gestion du système NVIDIA (nvidia-smi) est installée.

  Les métriques GPU NVIDIA collectées sont préfixées avec la chaîne `nvidia_smi_` pour les distinguer des métriques collectées pour d'autres types d'accélérateurs. Cette section peut inclure les champs suivants :

  - `drop_original_metrics` – Facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.
  - `measurement` – Spécifie l'éventail de métriques GPU NVIDIA à collecter. Pour obtenir une liste des valeurs possibles à utiliser ici, consultez la colonne Metric (Métrique) du tableau dans Collecter des métriques GPU NVIDIA .

    Dans l'entrée de chaque métrique, vous pouvez, si vous le souhaitez, spécifier l'une des valeurs suivantes :

    - `rename` – Spécifie un nom différent pour cette métrique.

    - `unit` – Spécifie l'unité à utiliser pour cette métrique, en remplaçant l'unité par défaut de `None` pour la métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.
  - `metrics_collection_interval` : facultatif. Indique la fréquence de collecte des métriques GPU NVIDIA, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.
- `procstat` – Facultatif. Spécifie que vous souhaitez récupérer les métriques auprès de processus individuels. Pour plus d'informations sur les options de configuration de procstat, consultez Collecter des métriques de processus avec le plugin procstat.
- `statsd` – Facultatif. Spécifie que vous souhaitez récupérer les métriques personnalisées à l'aide du protocole `StatsD`. L' CloudWatch agent agit comme un démon pour le protocole. Vous utilisez n'importe quel `StatsD` client standard pour envoyer les métriques à l' CloudWatch agent. Pour plus d'informations sur les options de configuration de StatsD, consultez Récupération de métriques personnalisées avec StatsD .
- `ethtool` – Facultatif. Spécifie que vous souhaitez récupérer les métriques réseau à l'aide du plugin `ethtool`. Ce plugin peut importer à la fois les métriques collectées par l'utilitaire ethtool standard, ainsi que les mesures de performances réseau des instances Amazon EC2. Pour plus d'informations sur les options de configuration de ethtool, consultez Récupérez des métriques des performances réseau.

L'exemple suivant illustre une section `metrics` pour un serveur Linux. Dans cet exemple, trois métriques d'UC, trois métriques netstat, trois métriques de processus et une métrique de disque sont collectées, et l'agent est configuré pour recevoir des métriques supplémentaires de la part d'un client `collectd`.

```
"metrics": {
    "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId",
 "InstanceType"],[]],
    "metrics_collected": {
      "collectd": {},
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit": "Percent"},
          {"name": "cpu_usage_nice", "unit": "Percent"},
```

```
          "cpu_usage_guest"
        ],
        "totalcpu": false,
        "drop_original_metrics": [ "cpu_usage_guest" ],
        "metrics_collection_interval": 10,
        "append_dimensions": {
          "test": "test1",
          "date": "2017-10-01"
        }
      },
      "netstat": {
        "measurement": [
          "tcp_established",
          "tcp_syn_sent",
          "tcp_close"
        ],
        "metrics_collection_interval": 60
      },
       "disk": {
        "measurement": [
          "used_percent"
        ],
        "resources": [
          "*"
        ],
        "drop_device": true
      },
      "processes": {
        "measurement": [
          "running",
          "sleeping",
          "dead"
        ]
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    }
  }
```

Windows Server

Dans la section `metrics_collected` pour Windows Server, vous pouvez avoir des sous-sections pour chaque objet de performance Windows, par exemple `Memory`, `Processor` et `LogicalDisk`. Pour plus d'informations sur les objets et les compteurs disponibles, consultez la rubrique Compteurs de performances de la documentation Microsoft Windows.

Dans la sous-section de chaque objet, vous spécifiez un tableau `measurement` des compteurs à collecter. Le tableau `measurement` est requis pour chaque objet que vous spécifiez dans le fichier de configuration. Vous pouvez également spécifier un champ `resources` pour nommer les instances à partir desquelles collecter les métriques. Vous pouvez également spécifier `*` pour `resources`, afin de collecter des métriques distinctes pour chaque instance. Si vous omettez `resources` pour les compteurs qui ont des instances, les données de toutes les instances sont regroupées dans un ensemble. Si vous omettez `resources` pour les compteurs qui n'ont pas d'instances, les compteurs ne sont pas collectés par l'agent. CloudWatch Pour déterminer si des compteurs disposent d'instances, vous pouvez utiliser l'une des commandes suivantes.

PowerShell :

```
Get-Counter -ListSet *
```

Ligne de commande (non Powershell) :

```
TypePerf.exe -q
```

Dans chaque section d'objet, vous pouvez également spécifier les champs facultatifs suivants :

- `metrics_collection_interval` – Facultatif. Indique la fréquence de collecte des métriques de cet objet, en remplaçant la valeur `metrics_collection_interval` globale spécifiée dans la section `agent` du fichier de configuration.

  Cette valeur est spécifiée en secondes. Par exemple, en spécifiant 10, les métriques sont collectées toutes les 10 secondes, et en le fixant à 300, les métriques sont collectées toutes les 5 minutes.

  Si vous définissez une valeur inférieure à 60 secondes, chaque métrique est collectée sous la forme d'une métrique haute résolution. Pour de plus amples informations, consultez Métriques haute résolution.

- `append_dimensions` – Facultatif. Spécifie d'autres dimensions à utiliser uniquement pour les métriques de cet objet. Si vous spécifiez ce champ, il est utilisé en plus des dimensions spécifiées dans le champ `append_dimensions` global qui est utilisé pour tous les types de métriques collectées par l'agent.
- `drop_original_metrics` : facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.

Dans chaque section de compteur, vous pouvez également spécifier les champs facultatifs suivants :

- `rename`— Spécifie un autre nom à utiliser CloudWatch pour cette métrique.
- `unit` – Spécifie l'unité à utiliser pour cette métrique. L'unité que vous spécifiez doit être une unité CloudWatch métrique valide, comme indiqué dans la `Unit` description dans MetricDatum.

Il existe deux autres sections facultatives que vous pouvez inclure dans `metrics_collected` :

- `statsd` – Vous permet de récupérer les métriques personnalisées à l'aide du protocole `StatsD`. L' CloudWatch agent agit comme un démon pour le protocole. Vous utilisez n'importe quel `StatsD` client standard pour envoyer les métriques à l' CloudWatchagent. Pour plus d'informations, consultez Récupération de métriques personnalisées avec StatsD .
- `procstat` – Vous permet de récupérer les métriques auprès de processus individuels. Pour de plus amples informations, consultez Collecter des métriques de processus avec le plugin procstat.

L'exemple suivant illustre une section `metrics` à utiliser dans Windows Server. Dans cet exemple, de nombreuses métriques Windows sont collectées et l'ordinateur est également défini pour recevoir des métriques supplémentaires à partir d'un client `StatsD`.

```
"metrics": {
    "metrics_collected": {
      "statsd": {},
      "Processor": {
        "measurement": [
          {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
```

```
          "% Interrupt Time",
          "% User Time",
          "% Processor Time"
        ],
        "resources": [
          "*"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      },
      "LogicalDisk": {
        "measurement": [
          {"name": "% Idle Time", "unit": "Percent"},
          {"name": "% Disk Read Time", "rename": "DISK_READ"},
          "% Disk Write Time"
        ],
        "resources": [
          "*"
        ]
      },
      "Memory": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Available Bytes",
          "Cache Faults/sec",
          "Page Faults/sec",
          "Pages/sec"
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "Network Interface": {
        "metrics_collection_interval": 5,
        "measurement": [
          "Bytes Received/sec",
          "Bytes Sent/sec",
          "Packets Received/sec",
          "Packets Sent/sec"
        ],
        "resources": [
          "*"
```

```
        ],
        "append_dimensions": {
          "d3": "win_bo"
        }
      },
      "System": {
        "measurement": [
          "Context Switches/sec",
          "System Calls/sec",
          "Processor Queue Length"
        ],
        "append_dimensions": {
          "d1": "win_foo",
          "d2": "win_bar"
        }
      }
    },
    "append_dimensions": {
      "ImageId": "${aws:ImageId}",
      "InstanceId": "${aws:InstanceId}",
      "InstanceType": "${aws:InstanceType}",
      "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
    },
    "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]]
    }
  }
```

CloudWatch fichier de configuration de l'agent : section Logs

La section `logs` peut inclure les champs suivants :

- `logs_collected` – Obligatoire si la section `logs` est incluse. Spécifie les fichiers journaux et les journaux d'événements Windows qui doivent être collectés à partir du serveur. Elle peut inclure deux champs, `files` et `windows_events`.
  - `files`— Spécifie les fichiers journaux réguliers que l' CloudWatch agent doit collecter. Il contient un champ, `collect_list`, qui définit plus précisément ces fichiers.
    - `collect_list` – Obligatoire si `files` est inclus. Contient un tableau d'entrées, chacune spécifiant un fichier journal à collecter. Chacune de ces entrées peut inclure les champs suivants :
      - `file_path`— Spécifie le chemin du fichier journal à télécharger dans CloudWatch Logs. Les règles de correspondance glob Unix standard sont acceptées, avec l'ajout de **

comme super astérisque. Par exemple, si vous indiquez `/var/log/**.log`, tous les fichiers `.log` dans l'arborescence de répertoires `/var/log` doivent être collectés. Pour obtenir des exemples supplémentaires, consultez Glob Library.

L'astérisque standard peut également être utilisé comme caractère générique standard. Par exemple, `/var/log/system.log*` correspond à des fichiers tels que `system.log_1111`, `system.log_2222` et ainsi de suite dans `/var/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.2018-06-01-01` et `access_log.2018-06-01-02`, mais pas de types différents, 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 de l'agent, afin que chaque type de fichier journal soit envoyé dans un flux de journal différent.

- `auto_removal` – Facultatif. Dans ce cas`true`, l' CloudWatch agent supprime automatiquement ce fichier journal après l'avoir lu et après avoir fait l'objet d'une rotation. Les fichiers CloudWatch journaux sont généralement supprimés une fois que l'intégralité de leur contenu a été chargée dans Logs, mais si l'agent atteint l'EOF (fin de fichier) et détecte également un autre fichier journal plus récent qui correspond au même`file_path`, l'agent supprime l'ANCIEN fichier. Vous devez donc vous assurer que vous avez terminé d'écrire dans l'ANCIEN fichier avant de créer le NOUVEAU fichier. La bibliothèque de suivi RUST présente une incompatibilité connue car elle risque de créer un NOUVEAU fichier journal puis de tenter d'écrire dans l'ANCIEN fichier journal.

  L'agent supprime uniquement les fichiers complets issus des journaux qui créent plusieurs fichiers, tels que les journaux qui créent des fichiers distincts pour chaque date. Si un journal écrit en continu dans un fichier unique, celui-ci n'est pas supprimé.

  Si vous avez déjà une méthode de rotation ou de suppression de fichiers journaux en place, nous vous recommandons d'ignorer ce champ ou de le définir sur `false`.

  Si vous omettez ce champ, la valeur par défaut de `false` est utilisée.

- `log_group_name` – Facultatif. Spécifie le nom du groupe de journaux à utiliser dans CloudWatch Logs.

  Nous vous recommandons d’utiliser ce champ pour spécifier un nom de groupe de journaux afin d’éviter toute confusion. Si vous omettez `log_group_name`, la valeur `file_path`

jusqu'au point final est utilisée comme nom de groupe de journaux. Par exemple, si le chemin de fichier est `/tmp/TestLogFile.log.2017-07-11-14`, le nom du groupe de journal est `/tmp/TestLogFile.log`.

Si vous spécifiez un nom de groupe de journaux, vous pouvez utiliser `{instance_id}`, `{hostname}`, `{local_hostname}` et `{ip_address}` en tant que variables au sein du nom. `{hostname}` récupère le nom d'hôte à partir des métadonnées EC2 et `{local_hostname}` utilise le nom d'hôte du fichier de configuration du réseau.

Si vous utilisez ces variables pour créer de nombreux groupes de journaux différents, gardez à l'esprit la limite de 1 000 000 groupes de journaux par compte et par région.

Les caractères autorisés sont : a–z, A–Z, 0–9, « _ » (trait de soulignement), « - » (tiret), « / » (barre oblique) et « . » (point).

- `log_group_class` : facultatif. Spécifie la classe de groupe de journaux à utiliser pour le nouveau groupe de journaux. Pour plus d'informations sur les classes de groupes de journaux, veuillez consulter la rubrique Log classes.

  Les valeurs valides sont `STANDARD` et `INFREQUENT_ACCESS`. Si vous omettez ce champ, la valeur par défaut de `STANDARD` est utilisée.

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

- `log_stream_name` : facultatif. Spécifie le nom du flux de journal à utiliser dans CloudWatch Logs. Dans le cadre du nom, vous pouvez utiliser `{instance_id}`, `{hostname}`, `{local_hostname}` et `{ip_address}` en tant que variables au sein du nom. `{hostname}` récupère le nom d'hôte à partir des métadonnées EC2 et `{local_hostname}` utilise le nom d'hôte du fichier de configuration du réseau.

  Si vous omettez ce champ, la valeur du paramètre `log_stream_name` dans la section `logs` globale est utilisée. Si ce champ est également omis, la valeur par défaut pour `{instance_id}` est utilisée.

  Un flux de journaux est créé automatiquement s'il n'en existe pas déjà.

- `retention_in_days` – Facultatif. Spécifie le nombre de jours de conservation des événements du journal dans le groupe de journaux spécifié.

- Si l'agent crée ce groupe de journaux maintenant et que vous omettez ce champ, la rétention de ce nouveau groupe de journaux est définie comme n'expirant jamais.
- Si ce groupe de journaux existe déjà et que vous spécifiez ce champ, la nouvelle rétention que vous spécifiez est utilisée. Si vous omettez ce champ pour un groupe de journaux existant, la rétention du groupe de journaux n'est pas modifiée.

  L'assistant de l' CloudWatch agent utilise `-1` comme valeur par défaut ce champ lorsqu'il est utilisé pour créer le fichier de configuration de l'agent et que vous ne spécifiez aucune valeur pour la conservation des journaux. Cette `-1` valeur définie par l'assistant indique que les événements du groupe de journaux n'expireront jamais. Cependant, la modification manuelle de cette valeur sur `-1` n'a aucun effet.

Les valeurs possibles sont : 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288, et 3653.

Si vous configurez l'agent pour qu'il écrive plusieurs flux de journaux dans le même groupe de journaux, le fait de spécifier le `retention_in_days` dans un seul endroit définira la rétention des journaux pour l'ensemble du groupe de journaux. Si vous spécifiez le `retention_in_days` pour le même groupe de journaux à plusieurs endroits, la rétention est définie si toutes ces valeurs sont égales. Toutefois, si différentes valeurs de `retention_in_days` sont spécifiées pour le même groupe de journaux à plusieurs endroits, la rétention des journaux ne sera pas définie et l'agent s'arrêtera, renvoyant une erreur.

> **Note**
>
> Le rôle IAM ou l'utilisateur IAM de l'agent doit disposer du `logs:PutRetentionPolicy` pour qu'il puisse définir des politiques de rétention. Pour plus d'informations, consultez Autoriser l' CloudWatch agent à définir une politique de conservation des journaux.

> **Warning**
>
> Si vous définissez le `retention_in_days` pour un groupe de journaux existant, tous les journaux de ce groupe publiés avant le nombre de jours que vous spécifiez

sont supprimés. Par exemple, une valeur de 3 entraînerait la suppression de tous les journaux datant de 3 jours et plus.

- `filters` : facultatif. Peut contenir un tableau d'entrées, chacune spécifiant une expression régulière et un type de filtre pour spécifier s'il faut publier ou supprimer des entrées de journal correspondant au filtre. Si vous omettez ce champ, tous les journaux du fichier journal sont publiés dans CloudWatch Logs. Si vous incluez ce champ, l'agent traite chaque message de journal avec tous les filtres que vous spécifiez, et seuls les événements du journal qui répondent à tous les filtres sont publiés dans CloudWatch Logs. Les entrées du journal qui ne répondent pas à tous les filtres resteront dans le fichier journal de l'hôte, mais ne seront pas envoyées à CloudWatch Logs.

  Chaque entrée du tableau de filtres peut comprendre les champs suivants :

  - `type`– Indique le type de filtre. Les valeurs valides sont `include` et `exclude`. Avec`include`, l'entrée du journal doit correspondre à l'expression à publier dans CloudWatch Logs. Avec`exclude`, chaque entrée de journal correspondant au filtre n'est pas envoyée à CloudWatch Logs.
  - `expression`– Chaîne d'expression régulière qui suit la Syntaxe RE2.

    > **Note**
    >
    > L' CloudWatch agent ne vérifie pas les performances des expressions régulières que vous fournissez et ne limite pas la durée d'exécution de l'évaluation des expressions régulières. Nous vous recommandons de faire attention à ne pas écrire une expression dont l'évaluation est coûteuse. Pour plus d'informations sur les problèmes possibles, consultez Regular Expression Denial of Service - ReDoS

  Par exemple, l'extrait suivant du fichier de configuration de l' CloudWatch agent publie des journaux qui sont des requêtes PUT et POST dans CloudWatch Logs, à l'exception des journaux provenant de Firefox.

```
"collect_list": [
  {
    "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
    "log_group_name": "test.log",
    "log_stream_name": "test.log",
```

```
    "filters": [
      {
        "type": "exclude",
        "expression": "Firefox"
      },
      {
        "type": "include",
        "expression": "P(UT|OST)"
      }
    ]
  },
  .....
]
```

> **Note**
>
> L'ordre des filtres dans le fichier de configuration est important pour les performances. Dans l'exemple précédent, l'agent élimine tous les journaux qui correspondent à `Firefox` avant de commencer à évaluer le deuxième filtre. Afin que moins d'entrées de journaux ne soient évaluées par plusieurs filtres, placez en premier le filtre qui devrait exclure le plus de journaux dans le fichier de configuration.

- `timezone` – Facultatif. Spécifie le fuseau horaire à utiliser lors de l'écriture des horodatages sur les événements du journal. Les valeurs valides sont `UTC` et `Local`. La valeur par défaut est `Local`.

  Ce paramètre est ignoré si vous ne spécifiez pas de valeur pour `timestamp_format`.

- `timestamp_format` – Facultatif. Spécifie le format d'horodatage à l'aide d'un texte brut et de symboles spéciaux commençant par %. Si vous omettez ce champ, l'heure actuelle est utilisée. Si vous utilisez ce champ, vous pouvez utiliser les symboles dans la liste suivante dans le cadre du format.

  Si une seule entrée de journal contient deux horodatages qui correspondent au format, le premier horodatage est utilisé.

  Cette liste de symboles est différente de celle utilisée par l'ancien agent CloudWatch Logs. Pour obtenir un résumé de ces différences, consultez Différences d'horodatage entre l'CloudWatch agent unifié et l'ancien CloudWatch agent Logs.

`%y`

Année sans siècle sous forme de nombre décimal auquel est ajouté un zéro. Par exemple, 19 pour représenter 2019.

`%Y`

Année sans siècle sous forme de nombre décimal. Par exemple, 2019.

`%b`

Mois sous forme du nom abrégé dans la langue locale

`%B`

Mois sous forme du nom complet dans la langue locale

`%m`

Mois sous forme de nombre décimal auquel est ajouté un zéro

`%-m`

Mois sous forme de nombre décimal (sans zéro supplémentaire)

`%d`

Jour du mois sous forme de nombre décimal auquel est ajouté un zéro

`%-d`

Jour du mois sous forme de nombre décimal (sans zéro supplémentaire)

`%A`

Nom complet du jour de la semaine, par exemple `Monday`

`%a`

Abréviation du jour de la semaine, par exemple `Mon`

`%H`

Heure (au format 24 heures) sous forme de nombre décimal auquel est ajouté un zéro

`%I`

Heure (au format 12 heures) sous forme de nombre décimal auquel est ajouté un zéro

`%-I`

Heure (au format 12 heures) sous forme de nombre décimal (sans zéro supplémentaire)

`%p`

AM ou PM

`%M`

Minutes sous forme de nombre décimal auquel est ajouté un zéro

`%-M`

Minutes sous forme de nombre décimal (sans zéro supplémentaire)

`%S`

Secondes sous forme de nombre décimal auquel est ajouté un zéro

`%-S`

Secondes sous forme de nombre décimal (sans zéro supplémentaire)

`%f`

Secondes fractionnelles sous forme de nombre décimal (1 à 9 chiffres), avec remplissage de zéros sur la gauche.

`%Z`

Fuseau horaire, par exemple `PST`

`%z`

Fuseau horaire, exprimé en tant que décalage entre le fuseau horaire local et l'heure universelle coordonnée (UTC). Par exemple, `-0700`. Seul ce format est pris en charge. Par exemple, `-07:00` n'est pas un format valide.

- `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.

  Si vous ne renseignez pas ce champ, le mode à plusieurs lignes est désactivé, et toute ligne commençant par un caractère autre qu'un espace ferme le message de journal précédent et commence un nouveau message de journal.

Si vous incluez ce champ, vous pouvez spécifier `{timestamp_format}` afin d'utiliser la même expression régulière que votre format d'horodatage. Sinon, vous pouvez spécifier une expression régulière différente que CloudWatch Logs utilisera pour déterminer les lignes de départ des entrées multilignes.

- `encoding` – Spécifie l'encodage du fichier journal pour que le fichier puisse être lu correctement. Si vous spécifiez un encodage incorrect, cela peut entraîner une perte de données car les caractères qui ne peuvent pas être décodés seront remplacés par d'autres caractères.

  La valeur par défaut est `utf-8`. Les valeurs possibles sont les suivantes :

  ```
  ascii, big5, euc-jp, euc-kr, gbk, gb18030, ibm866, iso2022-jp,
  iso8859-2, iso8859-3, iso8859-4, iso8859-5, iso8859-6, iso8859-7,
  iso8859-8, iso8859-8-i, iso8859-10, iso8859-13, iso8859-14,
  iso8859-15, iso8859-16, koi8-r, koi8-u, macintosh, shift_jis, utf-8,
  utf-16, utf-16le, UTF-16, UTF-16LE, windows-874, windows-1250,
  windows-1251, windows-1252, windows-1253, windows-1254,
  windows-1255, windows-1256, windows-1257, windows-1258, x-mac-
  cyrillic
  ```

- La section `windows_events` indique le type d'événements Windows à collecter depuis les serveurs exécutant Windows Server. Il inclut les champs suivants :
  - `collect_list` – Obligatoire si `windows_events` est inclus. Indique les types et niveaux des événements Windows à collecter. Chaque journal à collecter comporte une entrée dans cette section, laquelle peut inclure les champs suivants :
    - `event_name` – Spécifie le type des événements Windows à consigner. Ceci est équivalent au nom du canal du journal des événements de Windows : par exemple, `System`, `Security`, `Application` etc.. Ce champ est obligatoire pour chaque type d'événement Windows à consigner.

      > **Note**
      >
      > Lorsqu'il CloudWatch récupère des messages d'un canal de journal Windows, il recherche le canal de journal en fonction de ses `Full Name` propriétés. Pendant ce temps, le panneau de navigation de l'Observateur d'événements Windows affiche la propriété `Log Name` des canaux de journal. Les propriétés `Full Name` et `Log Name` ne correspondent pas toujours. Pour vérifier que la propriété `Full Name`

d'un canal, faites un clic droit dessus dans l'Observateur d'événements Windows et ouvrez Properties (Propriétés).

- `event_levels` – Spécifie les niveaux d'événement à consigner. Vous devez spécifier chaque niveau à consigner. Les valeurs possibles incluent `INFORMATION`, `WARNING`, `ERROR`, `CRITICAL` et `VERBOSE`. Ce champ est obligatoire pour chaque type d'événement Windows à consigner.
- `log_group_name` – Obligatoire Spécifie le nom du groupe de journaux à utiliser dans CloudWatch Logs.
- `log_stream_name` : facultatif. Spécifie le nom du flux de journal à utiliser dans CloudWatch Logs. Dans le cadre du nom, vous pouvez utiliser `{instance_id}`, `{hostname}`, `{local_hostname}` et `{ip_address}` en tant que variables au sein du nom. `{hostname}` récupère le nom d'hôte à partir des métadonnées EC2 et `{local_hostname}` utilise le nom d'hôte du fichier de configuration du réseau.

  Si vous omettez ce champ, la valeur du paramètre `log_stream_name` dans la section `logs` globale est utilisée. Si ce champ est également omis, la valeur par défaut pour `{instance_id}` est utilisée.

  Un flux de journaux est créé automatiquement s'il n'en existe pas déjà.
- `event_format` – Facultatif. Spécifie le format à utiliser lors du stockage des événements Windows dans CloudWatch les journaux. `xml`utilise le format XML comme dans l'Observateur d'événements Windows. `text`utilise l'ancien format de l'agent CloudWatch Logs.
- `retention_in_days` : facultatif. Spécifie le nombre de jours de conservation des événements Windows dans le groupe de journaux spécifié.
  - Si l'agent crée ce groupe de journaux maintenant et que vous omettez ce champ, la rétention de ce nouveau groupe de journaux est définie comme n'expirant jamais.
  - Si ce groupe de journaux existe déjà et que vous spécifiez ce champ, la nouvelle rétention que vous spécifiez est utilisée. Si vous omettez ce champ pour un groupe de journaux existant, la rétention du groupe de journaux n'est pas modifiée.

    L'assistant de l' CloudWatch agent utilise `-1` comme valeur par défaut ce champ lorsqu'il est utilisé pour créer le fichier de configuration de l'agent et que vous ne spécifiez aucune valeur pour la conservation des journaux. La valeur `-1` indique que la valeur définie par

l'assistant spécifie que les événements du groupe de journaux n'expirent pas. Cependant, la modification manuelle de cette valeur sur `-1` n'a aucun effet.

Les valeurs possibles sont : 1, 3, 5, 7, 14, 30, 60, 90, 120, 150, 180, 365, 400, 545, 731, 1827, 2192, 2557, 2922, 3288, et 3653.

Si vous configurez l'agent pour qu'il écrive plusieurs flux de journaux dans le même groupe de journaux, le fait de spécifier le `retention_in_days` dans un seul endroit définira la rétention des journaux pour l'ensemble du groupe de journaux. Si vous spécifiez le `retention_in_days` pour le même groupe de journaux à plusieurs endroits, la rétention est définie si toutes ces valeurs sont égales. Toutefois, si différentes valeurs de `retention_in_days` sont spécifiées pour le même groupe de journaux à plusieurs endroits, la rétention des journaux ne sera pas définie et l'agent s'arrêtera, renvoyant une erreur.

> **Note**
>
> Le rôle IAM ou l'utilisateur IAM de l'agent doit disposer du `logs:PutRetentionPolicy` pour qu'il puisse définir des politiques de rétention. Pour plus d'informations, consultez Autoriser l' CloudWatch agent à définir une politique de conservation des journaux.

> **Warning**
>
> Si vous définissez le `retention_in_days` pour un groupe de journaux existant, tous les journaux de ce groupe publiés avant le nombre de jours que vous spécifiez sont supprimés. Par exemple, une valeur de 3 entraînerait la suppression de tous les journaux datant de 3 jours et plus.

- `log_stream_name` – Obligatoire Indique le nom de flux de journal par défaut à utiliser pour les journaux ou les événements Windows qui n'ont pas de nom de flux de journal individuel défini dans le paramètre `log_stream_name` au sein de leur entrée dans `collect_list`.
- `endpoint_override` – Spécifie un point de terminaison FIPS ou un lien privé à utiliser en tant que point de terminaison auquel l'agent envoie les journaux. Cette spécification de champ et la définition d'un lien privé vous permettent d'envoyer les journaux à un point de terminaison Amazon VPC. Pour de plus amples informations, consultez Qu'est-ce qu'Amazon VPC ?.

La valeur de `endpoint_override` doit être une chaîne qui est une URL.

Par exemple, la partie suivante de la section Journaux (logs) du fichier de configuration définit l'agent pour qu'il utilise un point de terminaison d'un VPC lors de l'envoi de journaux.

```
{
  "logs": {
    "endpoint_override": "vpce-XXXXXXXXXXXXXXXXXXXXXXXX.logs.us-
east-1.vpce.amazonaws.com",
   ......
   },
}
```

- `force_flush_interval` – Spécifie en secondes la durée maximale pendant laquelle les journaux demeurent dans la mémoire tampon avant d'être envoyés au serveur. Quelle que soit la configuration de ce champ, si la taille des journaux dans la mémoire tampon atteint 1 Mo, les journaux sont immédiatement envoyés au serveur. La valeur par défaut est 5.

  Si vous utilisez l'agent pour signaler des métriques haute résolution au format de métrique intégrée et que vous définissez des alertes sur ces métriques, conservez la valeur par défaut de 5 pour ce paramètre. Dans le cas contraire, les métriques sont signalées avec un délai susceptible de déclencher des alertes si des données sont partielles ou incomplètes.
- `credentials`— Spécifie un rôle IAM à utiliser lors de l'envoi de journaux vers un autre AWS compte. S'il est spécifié, ce champ contient un paramètre, `role_arn`.
  - `role_arn`— Spécifie l'ARN d'un rôle IAM à utiliser pour l'authentification lors de l'envoi de journaux vers un autre AWS compte. Pour plus d'informations, consultez Envoi de métriques, de journaux et de traces à un autre compte. S'il est spécifié ici, il remplace le `role_arn` spécifié dans la section `agent` du fichier de configuration, le cas échéant.
- `metrics_collected`— Ce champ peut contenir des sections spécifiant que l'agent doit collecter des journaux afin de permettre des cas d'utilisation tels que CloudWatch Application Signals et Container Insights avec une meilleure observabilité pour Amazon EKS.
  - `app_signals`(Facultatif) Spécifie que vous souhaitez activer les signaux CloudWatch d'application. Pour plus d'informations sur cette configuration, consultezActiver les signaux CloudWatch d'application.
  - `kubernetes` : ce champ peut contenir un paramètre `enhanced_container_insights` que vous pouvez utiliser pour activer Container Insights avec une observabilité améliorée pour Amazon EKS.

- `enhanced_container_insights` : définissez cette option à `true` pour activer Container Insights avec une observabilité améliorée pour Amazon EKS. Pour plus d'informations, consultez Container Insights avec observabilité améliorée pour Amazon EKS.
- `accelerated_compute_metrics`— Définissez ce paramètre sur `false` pour désactiver la collecte des métriques du GPU Nvidia sur les clusters Amazon EKS. Pour plus d'informations, consultez Métriques du GPU NVIDIA.

- `emf` : pour collecter des métriques intégrées dans des journaux, il n'est plus nécessaire d'ajouter ce champ `emf`. Il s'agit d'un champ hérité spécifiant que l'agent doit collecter les journaux au format de métrique intégrée. Vous pouvez générer des données de métriques à partir de ces journaux. Pour de plus amples informations, consultez Intégration de métriques dans les journaux.

Voici un exemple de section `logs`.

```
"logs":
   {
       "logs_collected": {
           "files": {
               "collect_list": [
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\amazon-cloudwatch-agent.log",
                       "log_group_name": "amazon-cloudwatch-agent.log",
                       "log_stream_name": "my_log_stream_name_1",
                       "timestamp_format": "%H: %M: %S%y%b%-d"
                   },
                   {
                       "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\Logs\\test.log",
                       "log_group_name": "test.log",
                       "log_stream_name": "my_log_stream_name_2"
                   }
               ]
           },
           "windows_events": {
               "collect_list": [
                   {
                       "event_name": "System",
                       "event_levels": [
                           "INFORMATION",
```

```
                        "ERROR"
                    ],
                    "log_group_name": "System",
                    "log_stream_name": "System"
                },
                {
                    "event_name": "CustomizedName",
                    "event_levels": [
                        "INFORMATION",
                        "ERROR"
                    ],
                    "log_group_name": "CustomizedLogGroup",
                    "log_stream_name": "CustomizedLogStream"
                }
            ]
        }
    },
    "log_stream_name": "my_log_stream_name",
    "metrics_collected": {
        "kubernetes": {
            "enhanced_container_insights": true
    }
  }
}
```

CloudWatch fichier de configuration de l'agent : section Traces

En ajoutant une `traces` section au fichier de configuration de l' CloudWatch agent, vous pouvez activer CloudWatch Application Signals ou collecter des traces à partir de X-Ray et du SDK d' OpenTelemetry instrumentation et les envoyer à X-Ray.

> **Important**
>
> Le rôle IAM ou l'utilisateur IAM de l'agent doit avoir pour AWSXrayWriteOnlyAccesspolitique d'envoyer des données de suivi à X-Ray. Pour plus d’informations, consultez Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

Pour commencer rapidement à collecter des traces, vous pouvez simplement ajouter les éléments suivants au fichier de configuration de l' CloudWatch agent.

```
"traces_collected": {
```

```
        "xray": {
        },
        "otlp": {
        }
      }
```

Si vous ajoutez la section précédente au fichier de configuration de l' CloudWatch agent et que vous redémarrez l'agent, celui-ci commencera à collecter des traces en utilisant les options et valeurs par défaut suivantes. Pour plus d'informations sur ces paramètres, consultez les définitions des paramètres plus loin dans cette section.

```
"traces_collected": {
        "xray": {
            "bind_address": "127.0.0.1:2000",
            "tcp_proxy": {
              "bind_address": "127.0.0.1:2000"
            }
        },
        "otlp": {
            "grpc_endpoint": "127.0.0.1:4317",
            "http_endpoint": "127.0.0.1:4318"
        }
      }
```

La section `traces` peut inclure les champs suivants :

- `traces_collected` – Obligatoire si la section `traces` est incluse. Spécifie les kits SDK à partir desquels collecter des traces. La structure peut inclure les champs suivants :
  - `app_signals` : facultatif. Spécifie que vous souhaitez activer les signaux CloudWatch d'application. Pour plus d'informations sur cette configuration, consultezActiver les signaux CloudWatch d'application.
  - `xray` : facultatif. Spécifie que vous souhaitez collecter des traces à partir du kit SDK X-Ray. Cette section peut inclure les champs suivants :
    - `bind_address` – Facultatif. Spécifie l'adresse UDP que l' CloudWatch agent doit utiliser pour écouter les traces de X-Ray. Le format est `ip:port`. Cette adresse doit correspondre à l'adresse définie dans le kit SDK X-Ray.

      Si vous omettez ce champ, la valeur par défaut de `127.0.0.1:2000` est utilisée.

- `tcp_proxy` : facultatif. Configure l'adresse d'un proxy utilisé pour prendre en charge l'échantillonnage à distance X-Ray. Pour de plus amples informations, veuillez consulter Configuring sampling rules dans la documentation X-Ray.

  Cette section peut contenir les champs suivants.

  - `bind_address` : facultatif. Spécifie l'adresse TCP à laquelle l' CloudWatch agent doit configurer le proxy. Le format est `ip:port`. Cette adresse doit correspondre à l'adresse définie dans le kit SDK X-Ray.

    Si vous omettez ce champ, la valeur par défaut de `127.0.0.1:2000` est utilisée.

- `otlp` : facultatif. Spécifie que vous souhaitez collecter des traces à partir du OpenTelemetry SDK. Pour plus d'informations sur la AWS distribution pour OpenTelemetry, consultez la section AWS Distribution pour. OpenTelemetry Pour plus d'informations sur la AWS distribution pour les OpenTelemetry SDK, consultez la section Introduction.

  Cette section peut inclure les champs suivants :

  - `grpc_endpoint` – Facultatif. Spécifie l'adresse que l' CloudWatch agent doit utiliser pour écouter les OpenTelemetry traces envoyées à l'aide des appels de procédure à distance gRPC. Le format est `ip:port`. Cette adresse doit correspondre à l'adresse définie pour l'exportateur gRPC dans le OpenTelemetry SDK.

    Si vous omettez ce champ, la valeur par défaut de `127.0.0.1:4317` est utilisée.

  - `http_endpoint` : facultatif. Spécifie l'adresse que l' CloudWatch agent doit utiliser pour écouter les traces OTLP envoyées via HTTP. Le format est `ip:port`. Cette adresse doit correspondre à l'adresse définie pour l'exportateur HTTP dans le OpenTelemetry SDK.

    Si vous omettez ce champ, la valeur par défaut de `127.0.0.1:4318` est utilisée.

- `concurrency` : facultatif. Spécifie le nombre maximum d'appels simultanés à X-Ray qui peuvent être utilisés pour télécharger des traces. La valeur par défaut est 8.
- `local_mode` : facultatif. Si la valeur est `true`, l'agent ne collecte pas les métadonnées de l'instance Amazon EC2. La valeur par défaut est `false`.
- `endpoint_override` : facultatif. Spécifie un point de terminaison FIPS ou un lien privé à utiliser comme point de terminaison où l' CloudWatch agent envoie des traces. Cette spécification de champ et la définition d'un lien privé vous permettent d'envoyer les traces à un point de terminaison Amazon VPC. Pour plus d'informations, consultez Qu'est-ce qu'Amazon VPC ?.

  La valeur de `endpoint_override` doit être une chaîne qui est une URL.

- `region_override` : facultatif. Spécifie la région à utiliser pour le point de terminaison X-Ray. L' CloudWatch agent envoie les traces à X-Ray dans la région spécifiée. Si vous ne spécifiez pas ce champ, l'agent envoie les traces à la région dans laquelle se trouve l'instance Amazon EC2.

  Si vous spécifiez une région ici, elle est prioritaire sur le réglage du paramètre `region` dans la section `agent` du fichier de configuration.
- `proxy_override` : facultatif. Spécifie l'adresse du serveur proxy que l' CloudWatch agent doit utiliser lors de l'envoi de demandes à X-Ray. Le protocole du serveur proxy doit être spécifié dans le cadre de cette adresse.
- `credentials`— Spécifie un rôle IAM à utiliser lors de l'envoi de traces vers un autre AWS compte. S'il est spécifié, ce champ contient un paramètre, `role_arn`.
  - `role_arn`— Spécifie l'ARN d'un rôle IAM à utiliser pour l'authentification lors de l'envoi de traces vers un autre AWS compte. Pour plus d'informations, consultez Envoi de métriques, de journaux et de traces à un autre compte. S'il est spécifié ici, il remplace le `role_arn` spécifié dans la section `agent` du fichier de configuration, le cas échéant.

CloudWatch fichier de configuration de l'agent : exemples complets

Voici un exemple de fichier de configuration d' CloudWatch agent complet pour un serveur Linux.

Les éléments répertoriés dans les sections `measurement` qui concernent les métriques que vous souhaitez collecter peuvent spécifier soit l'intégralité du nom de la métrique, par exemple, ou simplement la partie du nom de la métrique qui sera ajoutée au type de ressource. Par exemple, la spécification de `reads` ou `diskio_reads` dans la section `measurement` de la section `diskio` entraîne la collecte de la métrique `diskio_reads`.

Cet exemple illustre les deux méthodes possibles pour spécifier des métriques dans la section `measurement`.

```
{
  "agent": {
    "metrics_collection_interval": 10,
    "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
  },
  "metrics": {
    "namespace": "MyCustomNamespace",
    "metrics_collected": {
      "cpu": {
        "resources": [
```

```
            "*"
          ],
          "measurement": [
            {"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit":
"Percent"},
            {"name": "cpu_usage_nice", "unit": "Percent"},
            "cpu_usage_guest"
          ],
          "totalcpu": false,
          "metrics_collection_interval": 10,
          "append_dimensions": {
            "customized_dimension_key_1": "customized_dimension_value_1",
            "customized_dimension_key_2": "customized_dimension_value_2"
          }
        },
        "disk": {
          "resources": [
            "/",
            "/tmp"
          ],
          "measurement": [
            {"name": "free", "rename": "DISK_FREE", "unit": "Gigabytes"},
            "total",
            "used"
          ],
           "ignore_file_system_types": [
            "sysfs", "devtmpfs"
          ],
          "metrics_collection_interval": 60,
          "append_dimensions": {
            "customized_dimension_key_3": "customized_dimension_value_3",
            "customized_dimension_key_4": "customized_dimension_value_4"
          }
        },
        "diskio": {
          "resources": [
            "*"
          ],
          "measurement": [
            "reads",
            "writes",
            "read_time",
            "write_time",
            "io_time"
```

```
            ],
            "metrics_collection_interval": 60
          },
          "swap": {
            "measurement": [
              "swap_used",
              "swap_free",
              "swap_used_percent"
            ]
          },
          "mem": {
            "measurement": [
              "mem_used",
              "mem_cached",
              "mem_total"
            ],
            "metrics_collection_interval": 1
          },
          "net": {
            "resources": [
              "eth0"
            ],
            "measurement": [
              "bytes_sent",
              "bytes_recv",
              "drop_in",
              "drop_out"
            ]
          },
          "netstat": {
            "measurement": [
              "tcp_established",
              "tcp_syn_sent",
              "tcp_close"
            ],
            "metrics_collection_interval": 60
          },
          "processes": {
            "measurement": [
              "running",
              "sleeping",
              "dead"
            ]
          }
```

```
        },
        "append_dimensions": {
          "ImageId": "${aws:ImageId}",
          "InstanceId": "${aws:InstanceId}",
          "InstanceType": "${aws:InstanceType}",
          "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
        },
        "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
 ["d1"],[]],
        "force_flush_interval" : 30
      },
      "logs": {
        "logs_collected": {
          "files": {
            "collect_list": [
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "log_stream_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.log",
                "log_group_name": "test.log",
                "log_stream_name": "test.log",
                "timezone": "Local"
              }
            ]
          }
        },
        "log_stream_name": "my_log_stream_name",
        "force_flush_interval" : 15,
        "metrics_collected": {
           "kubernetes": {
                "enhanced_container_insights": true
      }
    }
  }
}
```

Voici un exemple de fichier de configuration d' CloudWatch agent complet pour un serveur exécutant Windows Server.

```
{
      "agent": {
        "metrics_collection_interval": 60,
        "logfile": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\\amazon-
cloudwatch-agent.log"
      },
      "metrics": {
        "namespace": "MyCustomNamespace",
        "metrics_collected": {
          "Processor": {
            "measurement": [
              {"name": "% Idle Time", "rename": "CPU_IDLE", "unit": "Percent"},
              "% Interrupt Time",
              "% User Time",
              "% Processor Time"
            ],
            "resources": [
              "*"
            ],
            "append_dimensions": {
              "customized_dimension_key_1": "customized_dimension_value_1",
              "customized_dimension_key_2": "customized_dimension_value_2"
            }
          },
          "LogicalDisk": {
            "measurement": [
              {"name": "% Idle Time", "unit": "Percent"},
              {"name": "% Disk Read Time", "rename": "DISK_READ"},
              "% Disk Write Time"
            ],
            "resources": [
              "*"
            ]
          },
          "customizedObjectName": {
            "metrics_collection_interval": 60,
            "customizedCounterName": [
              "metric1",
              "metric2"
            ],
            "resources": [
              "customizedInstances"
            ]
```

```
        },
        "Memory": {
          "metrics_collection_interval": 5,
          "measurement": [
            "Available Bytes",
            "Cache Faults/sec",
            "Page Faults/sec",
            "Pages/sec"
          ]
        },
        "Network Interface": {
          "metrics_collection_interval": 5,
          "measurement": [
            "Bytes Received/sec",
            "Bytes Sent/sec",
            "Packets Received/sec",
            "Packets Sent/sec"
          ],
          "resources": [
            "*"
          ],
          "append_dimensions": {
            "customized_dimension_key_3": "customized_dimension_value_3"
          }
        },
        "System": {
          "measurement": [
            "Context Switches/sec",
            "System Calls/sec",
            "Processor Queue Length"
          ]
        }
      },
      "append_dimensions": {
        "ImageId": "${aws:ImageId}",
        "InstanceId": "${aws:InstanceId}",
        "InstanceType": "${aws:InstanceType}",
        "AutoScalingGroupName": "${aws:AutoScalingGroupName}"
      },
      "aggregation_dimensions" : [["ImageId"], ["InstanceId", "InstanceType"],
["d1"],[]]
    },
    "logs": {
      "logs_collected": {
```

```
          "files": {
            "collect_list": [
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\amazon-cloudwatch-agent.log",
                "log_group_name": "amazon-cloudwatch-agent.log",
                "timezone": "UTC"
              },
              {
                "file_path": "c:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\Logs\
\test.log",
                "log_group_name": "test.log",
                "timezone": "Local"
              }
            ]
          },
          "windows_events": {
            "collect_list": [
              {
                "event_name": "System",
                "event_levels": [
                  "INFORMATION",
                  "ERROR"
                ],
                "log_group_name": "System",
                "log_stream_name": "System",
                "event_format": "xml"
              },
              {
                "event_name": "CustomizedName",
                "event_levels": [
                  "WARNING",
                  "ERROR"
                ],
                "log_group_name": "CustomizedLogGroup",
                "log_stream_name": "CustomizedLogStream",
                "event_format": "xml"
              }
            ]
          }
        },
        "log_stream_name": "example_log_stream_name"
      }
```

```
    }
```

Enregistrez le fichier de configuration de l' CloudWatch agent manuellement

Si vous créez ou modifiez le fichier de configuration de l' CloudWatch agent manuellement, vous pouvez lui donner n'importe quel nom. Pour faciliter les éventuels dépannages, nous vous recommandons de le nommer `/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` sur un serveur Linux et `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json` sur les serveurs exécutant Windows Server. Vous pouvez ensuite copier ce fichier vers d'autres serveurs sur lesquels vous souhaitez exécuter l'agent une fois que vous avez créé le fichier.

Téléchargement du fichier de configuration de l' CloudWatch agent dans le magasin de paramètres de Systems Manager

Si vous envisagez d'utiliser l'agent SSM pour installer l' CloudWatch agent sur des serveurs, après avoir modifié manuellement le fichier de configuration de l' CloudWatch agent, vous pouvez le télécharger dans le magasin de paramètres de Systems Manager. Pour cela, utilisez la commande Systems Manager `put-parameter`.

Pour être en mesure de stocker le fichier dans le Parameter Store, vous devez utiliser un rôle IAM avec les autorisations suffisantes. Pour de plus amples informations, consultez Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch .

Utilisez la commande suivante, où *`nom de paramètre`* est le nom à utiliser pour ce fichier dans le Parameter Store et *`configuration_file_pathname`* est le chemin et le nom du fichier de configuration que vous avez modifié.

```
aws ssm put-parameter --name "parameter name" --type "String" --value
 file://configuration_file_pathname
```

Activer les signaux CloudWatch d'application

Utilisez les signaux d' CloudWatch application pour piloter automatiquement vos applications AWS afin de suivre les performances des applications par rapport à vos objectifs commerciaux. Application Signals vous fournit une vue unifiée et centrée sur les applications de vos applications Java, de leurs dépendances et de leurs limites. Pour plus d’informations, consultez Application Signals.

CloudWatch Application Signals utilise l' CloudWatch agent pour recevoir des métriques et des traces de vos applications instrumentées automatiquement, appliquer éventuellement des règles pour

réduire la cardinalité élevée, puis publier la télémétrie traitée sur. CloudWatch Vous pouvez fournir une configuration personnalisée à l' CloudWatch agent spécifiquement pour les signaux d'application à l'aide du fichier de configuration de l'agent. Pour commencer, la présence d'une `app_signals` section sous la `metrics_collected` section au sein de la `logs` section du fichier de configuration de l'agent indique que l' CloudWatch agent recevra des métriques de vos applications instrumentées automatiquement. De même, la présence d'une `app_signals` section sous la `traces_collected` section au sein de la `traces` section du fichier de configuration de l'agent indique que l' CloudWatch agent est activé pour recevoir des traces de vos applications instrumentées automatiquement. En outre, vous pouvez éventuellement transmettre des règles de configuration personnalisées afin de réduire la publication de télémétrie à cardinalité élevée, comme indiqué dans cette section.

- Pour les clusters Amazon EKS, lorsque vous installez le module complémentaire Amazon CloudWatch Observability EKS, l' CloudWatch agent est activé par défaut pour recevoir à la fois des métriques et des traces de vos applications instrumentées automatiquement. Si vous souhaitez éventuellement transmettre des règles de configuration personnalisées, vous pouvez le faire en transmettant une configuration d'agent personnalisée au module complémentaire Amazon EKS lorsque vous le créez ou le mettez à jour en utilisant une configuration supplémentaire, comme indiqué dans (Facultatif) Configuration supplémentaire.
- Pour les autres plateformes prises en charge, notamment Amazon EC2, vous devez démarrer l' CloudWatch agent avec une configuration d'agent qui active les signaux d'application en spécifiant les `app_signals` sections et éventuellement les règles de configuration personnalisées, comme indiqué plus loin dans cette section.

Vous trouverez ci-dessous un aperçu des champs du fichier de configuration de l' CloudWatch agent liés aux signaux CloudWatch d'application.

- `logs`
  - `metrics_collected`— Ce champ peut contenir des sections spécifiant que l'agent doit collecter des journaux afin de permettre des cas d'utilisation tels que CloudWatch Application Signals et Container Insights avec une meilleure observabilité pour Amazon EKS.

    > **Note**
    >
    > Auparavant, cette section était également utilisée pour spécifier que l'agent doit collecter les journaux au format de métrique intégrée. Ces paramètres ne sont plus nécessaires.

- `app_signals`(Facultatif) Spécifie que vous souhaitez permettre à CloudWatch Application Signals de recevoir des métriques de vos applications instrumentées automatiquement afin de faciliter les signaux d' CloudWatch application.
  - `rules` : (facultatif) un ensemble de règles permettant de sélectionner de manière conditionnelle des métriques et des suivis et d'appliquer des actions pour gérer les scénarios de cardinalité élevée. Chaque règle peut contenir les champs suivants :
    - `rule_name` : (facultatif) le nom de la règle.
    - `selectors` : (facultatif) un ensemble de métriques et d'analyseur de dimension de suivis. Chaque sélecteur doit fournir les champs suivants :
      - `dimension` : obligatoire si `selectors` n'est pas vide. Cela indique la dimension des métriques et des suivis à utiliser comme filtre.
      - `match` : obligatoire si `selectors` n'est pas vide. Un motif de caractères génériques utilisé pour faire correspondre les valeurs de la dimension spécifiée.
    - `action` : (facultatif) action à appliquer aux métriques et aux suivis correspondant aux sélecteurs spécifiés. La valeur de `action` doit être l'un des mots clés suivants :
      - `keep`Spécifie de n'envoyer que les métriques et les traces CloudWatch si elles correspondent au`selectors`.
      - `drop` : spécifie de supprimer la métrique et les suivis qui correspondent aux `selectors`.
      - `replace` : spécifie de remplacer les dimensions des métriques et des suivis correspondantes aux `selectors`. Ils sont remplacés conformément à la section `replacements`.
    - `replacements` Obligatoire si `action` a pour valeur `replace`. Un tableau de paires de dimensions / valeurs qui seront appliquées aux métriques et aux suivis qui correspondent aux `selectors` spécifiés lorsque `action` est `replace`. Chaque remplacement doit fournir les champs suivants :
      - `target_dimension` : obligatoire si `replacements` n'est pas vide. Spécifie la dimension à remplacer.
      - `value` : obligatoire si `replacements` n'est pas vide. La valeur par laquelle remplacer la valeur `target_dimension` d'origine.
  - `limiter`(Facultatif) Utilisez cette section pour limiter le nombre de mesures et de dimensions auxquelles Application Signals envoie CloudWatch, afin d'optimiser vos coûts.

- `disabled`(Facultatif) Si`true`, la fonction de limitation des métriques est désactivée. La valeur par défaut est `false`.
- `drop_threshold`(Facultatif) Le nombre maximum de métriques distinctes par service dans un intervalle de rotation qui peuvent être exportées par un CloudWatch agent. La valeur par défaut est 500.
- `rotation_interval`(Facultatif) Intervalle auquel le limiteur réinitialise les enregistrements métriques pour le comptage des distinctions. Ceci est exprimé sous la forme d'une chaîne avec une séquence de nombres et un suffixe d'unité. Les fractions sont prises en charge. Les suffixes d'unités pris en charge sont `sm`,, `hms`, et `us ns`

  La valeur par défaut est `1h` d'une heure.
- `log_dropped_metrics`(Facultatif) Spécifie si l'agent doit écrire des journaux dans les journaux de l' CloudWatch agent lorsque les métriques Application Signals sont supprimées. L'argument par défaut est `false`.

  > **Note**
  >
  > Pour activer cette journalisation, le `debug` paramètre de la `agent` section doit également être défini sur`true`.

- `traces`
  - `traces_collected`
    - `app_signals` : facultatif. Spécifiez-le pour permettre à l' CloudWatch agent de recevoir des traces de vos applications instrumentées automatiquement afin de faciliter les signaux CloudWatch d'application.

> **Note**
>
> Même si les règles personnalisées `app_signals` sont spécifiées dans la section `metrics_collected` contenue dans la section `logs`, elles s'appliquent également implicitement à la section `traces_collected`. Le même ensemble de règles s'appliquera à la fois aux métriques et aux suivis.

Lorsqu'il existe plusieurs règles comportant des actions différentes, elles s'appliquent dans l'ordre suivant : `keep`, puis `drop`, puis `replace`.

Voici un exemple de fichier de configuration d' CloudWatch agent complet qui applique des règles personnalisées.

```
{
  "logs": {
    "metrics_collected": {
      "app_signals": {
        "rules": [
          {
            "rule_name": "keep01",
            "selectors": [
              {
                "dimension": "Service",
                "match": "pet-clinic-frontend"
              },
              {
                "dimension": "RemoteService",
                "match": "customers-service"
              }
            ],
            "action": "keep"
          },
          {
            "rule_name": "drop01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "GET /api/customer/owners/*"
              }
            ],
            "action": "drop"
          },
          {
            "rule_name": "replace01",
            "selectors": [
              {
                "dimension": "Operation",
                "match": "PUT /api/customer/owners/*/pets/*"
              },
              {
                "dimension": "RemoteOperation",
                "match": "PUT /owners"
              }
```

```
            ],
            "replacements": [
              {
                "target_dimension": "Operation",
                "value": "PUT /api/customer/owners/{ownerId}/pets{petId}"
              }
            ],
            "action": "replace"
          }
        ]
      }
    }
  },
  "traces": {
    "traces_collected": {
      "app_signals": {}
    }
  }
}
```

Dans l'exemple de fichier de configuration précédent, les `rules` sont traitées comme suit :

1. La règle `keep01` garantit que toute métrique et tout suivi dont la dimension `Service` est `pet-clinic-frontend` et la dimension `RemoteService` est `customers-service` sont conservées.
2. Pour les métriques et les suivis traités après application de `keep01`, la règle `drop01` garantit que les métriques et les suivis dont la dimension `Operation` est `GET /api/customer/owners/*` sont supprimées.
3. Pour les métriques et les suivis traités après application de `drop01`, la règle `replace01` met à jour les métriques et les suivis dont la dimension `Operation` est `PUT /api/customer/owners/*/pets/*` et la dimension `RemoteOperation` est `PUT /owners` telles que leur dimension `Operation` est désormais remplacée par la valeur `PUT /api/customer/owners/{ownerId}/pets{petId}`.

Voici un exemple complet de fichier de CloudWatch configuration qui gère la cardinalité dans Application Signals en modifiant la limite des métriques à 100, en activant l'enregistrement des métriques supprimées et en fixant l'intervalle de rotation à deux heures.

```
{
    "logs": {
```

```
        "metrics_collected": {
            "app_signals": {
                "limiter": {
                    "disabled": false,
                    "drop_threshold": 100,
                    "rotation_interval": "2h",
                    "log_dropped_metrics": true
                }
            }
        },
        "traces": {
            "traces_collected": {
                "app_signals": {}
            }
        }
    }
}
```

Récupérez des métriques des performances réseau

Les instances EC2 exécutées sur Linux qui utilisent l'adaptateur Elastic Network Adapter (ENA) publient des métriques de performances réseau. Les versions 1.246396.0 et ultérieures de l' CloudWatch agent vous permettent d'importer ces mesures de performance réseau dans. CloudWatch Lorsque vous importez ces mesures de performance réseau dans CloudWatch, elles sont facturées en tant que mesures CloudWatch personnalisées.

Pour plus d'informations sur le pilote ENA, consultez Enabling enhanced networking with the Elastic Network Adapter (ENA) on Linux instances (Activation de la mise en réseau améliorée avec Elastic Network Adapter [ENA] sur les instances Linux) et Enabling enhanced networking with the Elastic Network Adapter (ENA) on Windows instances (Activation de la mise en réseau améliorée avec Elastic Network Adapter [ENA] sur les instances Windows).

La façon dont vous configurez l'ensemble des métriques de performances réseau diffère sur les serveurs Linux et Windows.

Le tableau suivant répertorie ces métriques de performances réseau activées par l'adaptateur ENA. Lorsque l' CloudWatch agent importe ces métriques CloudWatch depuis des instances Linux, il les ajoute `ethtool_` au début de chaque nom de métrique.

Métrique	Description
Nom sur les serveurs Linux : **bw_in_allowance_exceeded** Nom sur les serveurs Windows : **Aggregate inbound BW allowance exceeded**	Nombre de paquets mis en file d'attente et/ou ignorés parce que la bande passante agrégée entrante a dépassé le maximum de l'instance. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune
Nom sur les serveurs Linux : **bw_out_allowance_exceeded** Nom sur les serveurs Windows : **Aggregate outbound BW allowance exceeded**	Nombre de paquets mis en file d'attente et/ou ignorés parce que la bande passante agrégée sortante a dépassé le maximum de l'instance. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune
Nom sur les serveurs Linux :**conntrack_allowance_available** Nom sur les serveurs Windows : **Available connection tracking allowance**	Signale le nombre de connexions suivies pouvant être établies par l'instance avant d'atteindre l'allocation Connexions suivies de ce type d'instance. Cette métrique est disponible uniquement sur les instances EC2 basées sur Nitro utilisant le pilote Linux pour Elastic Network Adapter (ENA) à partir de la version 2.8.1, et sur les ordinateurs utilisant le pilote Windows pour Elastic Network Adapter (ENA) à partir de la version 2.6.0.

Métrique	Description
	Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune
Nom sur les serveurs Linux :**`ena_srd_mode`** Nom sur les serveurs Windows : **`ena srd mode`**	Décrit les fonctionnalités d'ENA Express activées. Pour plus d'informations sur ENA Express, voir Améliorer les performances du réseau avec ENA Express sur les instances Linux. Les valeurs sont les suivantes : • `0` = ENA Express désactivé, UDP désactivé • `1` = ENA Express activé, UDP désactivé • `2` = ENA Express désactivé, UDP activé **Note** Cela se produit uniquement lorsque ENA Express a été initialement activé et que UDP a été configuré pour l'utiliser. La valeur précédente est conservée pour le trafic UDP. • `3` = ENA Express activé, UDP activé

Métrique	Description
Nom sur les serveurs Linux :**ena_srd_eligible_tx_pkts** Nom sur les serveurs Windows : **ena srd eligible tx pkts**	Le nombre de paquets réseau envoyés au cours d'une période donnée qui répondent aux exigences d'éligibilité du datagramme fiable AWS évolutif (SRD), comme suit : • Les types d’instance d’envoi et de réception sont pris en charge. • ENA Express doit être configuré pour les instances d’envoi et de réception. • Les instances d'envoi et de réception doivent se trouver sur le même sous-réseau. • Le chemin réseau entre les instances ne doit pas inclure de boîtiers intergiciels. ENA Express ne prend actuellement pas en charge les boîtiers intergiciels.
Nom sur les serveurs Linux :**ena_srd_tx_pkts** Nom sur les serveurs Windows : **ena srd tx pkts**	Le nombre de paquets SRD transmis au cours d'une période donnée.
Nom sur les serveurs Linux :**ena_srd_rx_pkts** Nom sur les serveurs Windows : **ena srd rx pkts**	Le nombre de paquets SRD reçus au cours d'une période donnée.
Nom sur les serveurs Linux :**ena_srd_resource_utilization** Nom sur les serveurs Windows : **ena srd resource utilization**	Pourcentage de l'utilisation maximale de la mémoire autorisée pour les connexions SRD simultanées consommées par l'instance.

Métrique	Description
Nom sur les serveurs Linux : **`linklocal_allowance_exceeded`** Nom sur les serveurs Windows : **`Link local packet rate allowance exceeded`**	Nombre de paquets ignorés abandonné que le PPS du trafic vers les services proxy locaux a dépassé le maximum de l'interface réseau. Cela affecte le trafic vers le service DNS, le service des métadonnées d'instance et le service Amazon Time Sync. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune
Nom sur les serveurs Linux : **`linklocal_allowance_exceeded`** Nom sur les serveurs Windows : **`Link local packet rate allowance exceeded`**	Nombre de paquets ignorés abandonné que le PPS du trafic vers les services proxy locaux a dépassé le maximum de l'interface réseau. Cela affecte le trafic vers le service DNS, le service des métadonnées d'instance et le service Amazon Time Sync. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune

Métrique	Description
Nom sur les serveurs Linux : **pps_allowance_exceeded** Nom sur les serveurs Windows : **PPS allowance exceeded**	Nombre de paquets mis en file d'attente et/ou ignorés parce que le PPS bidirectionnel a dépassé le maximum de l'instance. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune

Configuration de Linux

Sur les serveurs Linux, le plugin ethtool vous permet d'importer les mesures de performance du réseau dans CloudWatch.

ethtool est un utilitaire Linux standard qui peut collecter des statistiques sur les périphériques Ethernet sur les serveurs Linux. Les statistiques qu'il recueille dépendent du périphérique réseau et du pilote. Parmi les exemples de ces statistiques figurent `tx_cnt`, `rx_bytes`, `tx_errors`, et `align_errors`. Lorsque vous utilisez le plug-in ethtool avec l' CloudWatch agent, vous pouvez également importer ces statistiques CloudWatch, ainsi que les mesures de performance du réseau EC2 répertoriées plus haut dans cette section.

> **Tip**
>
> Pour obtenir les statistiques disponibles sur notre système d'exploitation et notre périphérique réseau, utilisez la commande `ethtool -S`.

Lorsque l' CloudWatch agent importe des métriques dans CloudWatch, il ajoute un `ethtool_` préfixe aux noms de toutes les métriques importées. Ainsi, la statistique standard d'ethtool `rx_bytes` est appelée `ethtool_rx_bytes` CloudWatch, et la métrique de performance du réseau EC2 `bw_in_allowance_exceeded` est appelée. `ethtool_bw_in_allowance_exceeded` CloudWatch

Sur les serveurs Linux, pour importer les métriques ethtool, ajoutez une `ethtool` section à la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. La section `ethtool` peut inclure les sous-sections suivantes :

- interface_include – L'inclusion de cette section entraîne l'agent à collecter des métriques à partir des interfaces dont les noms sont répertoriés dans cette section. Si vous omettez cette section, les métriques sont collectées à partir de toutes les interfaces Ethernet qui ne sont pas répertoriées dans `interface_exclude`.

  L'interface Ethernet par défaut est `eth0`.
- interface_exclude – Si vous incluez cette section, listez les interfaces Ethernet à partir desquelles vous ne souhaitez pas collecter les métriques.

  Le plugin ethtool ignore toujours les interfaces de bouclage.
- metrics_include — Cette section répertorie les métriques dans lesquelles importer. CloudWatch Il peut inclure à la fois des statistiques standard collectées par ethtool et des métriques réseau haute résolution Amazon EC2.

L'exemple suivant montre une partie du fichier de configuration de l' CloudWatch agent. Cette configuration collecte les métriques ethtool standard `rx_packets` et `tx_packets`, et les métriques de performances réseau Amazon EC2 à partir de l'interface `eth1`.

Pour plus d'informations sur le fichier de configuration de l' CloudWatch agent, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

```
"metrics": {
    "append_dimensions": {
      "InstanceId": "${aws:InstanceId}"
    },
    "metrics_collected": {
      "ethtool": {
        "interface_include": [
          "eth1"
        ],
        "metrics_include": [
          "rx_packets",
          "tx_packets",
          "bw_in_allowance_exceeded",
          "bw_out_allowance_exceeded",
          "conntrack_allowance_exceeded",
```

```
          "linklocal_allowance_exceeded",
          "pps_allowance_exceeded"
         ]
      }
   }
}
```

Configuration Windows

Sur les serveurs Windows, les mesures de performance réseau sont disponibles via les compteurs de performance Windows, à partir desquels l' CloudWatch agent collecte déjà les mesures. Vous n'avez donc pas besoin d'un plugin pour collecter ces métriques à partir des serveurs Windows.

Voici un exemple de fichier de configuration pour collecter les métriques de performance du réseau à partir de Windows. Pour plus d'informations sur la modification du fichier de configuration de l' CloudWatch agent, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

```
{
    "metrics": {
        "append_dimensions": {
            "InstanceId": "${aws:InstanceId}"
        },
        "metrics_collected": {
            "ENA Packets Shaping": {
                "measurement": [
                    "Aggregate inbound BW allowance exceeded",
                    "Aggregate outbound BW allowance exceeded",
                    "Connection tracking allowance exceeded",
                    "Link local packet rate allowance exceeded",
                    "PPS allowance exceeded"
                ],
                "metrics_collection_interval": 60,
                "resources": [
                    "*"
                ]
            }
        }
    }
}
```

Afficher les métriques des performances réseau

Après avoir importé les indicateurs de performance du réseau CloudWatch, vous pouvez les visualiser sous forme de graphiques chronologiques et créer des alarmes qui peuvent surveiller ces indicateurs et vous avertir s'ils dépassent un seuil que vous spécifiez. La procédure suivante montre comment afficher les métriques d'ethtool sous la forme d'un graphique de séries chronologiques. Pour plus d'informations sur la configuration des alertes, consultez Utilisation des CloudWatch alarmes Amazon.

Toutes ces mesures étant des compteurs agrégés, vous pouvez utiliser des fonctions mathématiques CloudWatch métriques, par exemple `RATE(METRICS())` pour calculer le taux de ces mesures dans des graphiques ou les utiliser pour définir des alarmes. Pour de plus amples informations sur les fonctions mathématiques de métrique, consultez Utilisation des mathématiques appliquées aux métriques.

Pour consulter les indicateurs de performance du réseau dans la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sélectionnez l'espace de noms pour les métriques collectées par l'agent. Par défaut, il s'agit de CWagent, mais vous avez peut-être spécifié un espace de noms différent dans le fichier de configuration de l' CloudWatch agent.
4. Sélectionnez une dimension de métrique (Per-Instance Metrics (Métriques par instance) par exemple).
5. L'onglet All metrics (Toutes les métriques) affiche toutes les métriques pour cette dimension dans l'espace de nom. Vous pouvez effectuer les actions suivantes :
   a. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d’en-tête du tableau.
   b. Pour trier le tableau, utilisez l’en-tête de colonne.
   c. Pour filtrer par ressource, sélectionnez l'ID de ressource, puis Ajouter à la recherche.
   d. Pour filtrer par métrique, choisissez le nom de la métrique, puis Ajouter à la recherche.
6. (Facultatif) Pour ajouter ce graphique à un CloudWatch tableau de bord, choisissez Actions, puis sélectionnez Ajouter au tableau de bord.

Collecter des métriques GPU NVIDIA

Vous pouvez utiliser l' CloudWatch agent pour collecter les métriques du GPU NVIDIA à partir de serveurs Linux. Pour configurer cela, ajoutez une `nvidia_gpu` section à l'intérieur de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour plus d'informations, consultez Section Linux.

En outre, un pilote NVIDIA doit être installé sur l'instance. Pilotes NVIDIA préinstallés sur certaines Amazon Machine Images (AMI). Sinon, vous pouvez installer le pilote manuellement. Pour plus d'informations, consultez Installer les pilotes NVIDIA sur des instances Linux.

Les métriques suivantes peuvent être collectées. Toutes ces métriques sont collectées sans aucun CloudWatch `Unit`, mais vous pouvez spécifier une unité pour chaque métrique en ajoutant un paramètre au fichier de configuration de l' CloudWatch agent. Pour plus d'informations, consultez Section Linux.

Métrique	Nom de la métrique dans CloudWatch	Description
`utilization_gpu`	`nvidia_smi_utilization_gpu`	Pourcentage de temps sur la dernière période d'échantillonnage au cours de laquelle un ou plusieurs noyaux du GPU étaient en cours d'exécution.
`temperature_gpu`	`nvidia_smi_temperature_gpu`	Température centrale du GPU en degrés Celsius.
`power_draw`	`nvidia_smi_power_draw`	Dernière consommation d'énergie mesurée pour l'ensemble de la carte, en watts.
`utilization_memory`	`nvidia_smi_utilization_memory`	Pourcentage de temps sur la dernière période d'échantillonnage au cours de laquelle la mémoire globale (périphérique) était en cours de lecture ou d'écriture.
`fan_speed`	`nvidia_smi_fan_speed`	Pourcentage de la vitesse maximale du ventilateur auquel le ventilateur de l'appareil est censé fonctionner.

Métrique	Nom de la métrique dans CloudWatch	Description
`memory_total`	`nvidia_smi_memory_total`	Mémoire totale déclarée, en Mo.
`memory_used`	`nvidia_smi_memory_used`	Mémoire utilisée, en Mo.
`memory_free`	`nvidia_smi_memory_free`	Mémoire libre, en Mo.
`pcie_link_gen_current`	`nvidia_smi_pcie_link_gen_current`	Génération de liens actuelle.
`pcie_link_width_current`	`nvidia_smi_pcie_link_width_current`	Largeur de liens actuelle.
`encoder_stats_session_count`	`nvidia_smi_encoder_stats_session_count`	Nombre actuel de sessions de l'encodeur.
`encoder_stats_average_fps`	`nvidia_smi_encoder_stats_average_fps`	Moyenne mobile des images d'encodage par seconde.
`encoder_stats_average_latency`	`nvidia_smi_encoder_stats_average_latency`	Moyenne mobile de la latence d'encodage en microsecondes.
`clocks_current_graphics`	`nvidia_smi_clocks_current_graphics`	Fréquence actuelle de l'horloge de graphiques (ombrage).

Métrique	Nom de la métrique dans CloudWatch	Description
`clocks_current_sm`	`nvidia_smi_clocks_current_sm`	Fréquence actuelle de l'horloge multiprocesseur de streaming (SM).
`clocks_current_memory`	`nvidia_smi_clocks_current_memory`	Fréquence actuelle de l'horloge mémoire.
`clocks_current_video`	`nvidia_smi_clocks_current_video`	Fréquence actuelle des horloges vidéo (encodeur et décodeur).

Toutes ces mesures sont collectées avec les dimensions suivantes :

Dimension	Description
`index`	Identifiant unique du GPU sur ce serveur. Représente l'index de NVIDIA Management Library (NVML) (la bibliothèque de gestion NVIDIA) de l'appareil.
`name`	Type de GPU. Par exemple, `NVIDIA Tesla A100`

Dimension	Description
`host`	Nom d'hôte du serveur.

Collecter des métriques de processus avec le plugin procstat

Le plugin procstat vous permet de collecter des métriques auprès de processus individuels. Il est pris en charge sur les serveurs Linux et sur les serveurs exécutant une version compatible de Windows Server.

Rubriques

- Configuration de l' CloudWatch agent pour procstat
- Métriques collectées par procstat
- Affichage des métriques de processus importées par l' CloudWatch agent

Configuration de l' CloudWatch agent pour procstat

Pour utiliser le plugin procstat, ajoutez une `procstat` section dans la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Il existe trois façons de spécifier le processus à surveiller. Vous ne pouvez utiliser que l'une de ces méthodes, mais vous pouvez utiliser cette méthode pour spécifier un ou plusieurs processus à surveiller.

- `pid_file` : sélectionne les processus selon les noms des fichiers de numéros d'identification de processus (PID) qu'ils créent.
- `exe` : sélectionne les processus dont les noms correspondent à la chaîne que vous spécifiez, à l'aide de règles de correspondance d'expression régulière. La correspondance est une correspondance de type « contient », ce qui signifie que si vous spécifiez `agent` comme terme de correspondance, les processus avec des noms tels que `cloudwatchagent` correspondent à ce terme. Pour plus d'informations, consultez Syntaxe.
- `pattern` : sélectionne les processus selon les lignes de commande utilisées pour démarrer les processus. Toutes les processus sélectionnés ont des lignes de commande correspondant à la chaîne spécifiée à l'aide de règles de correspondance d'expression régulière. L'ensemble de la ligne de commande est vérifiée, y compris les paramètres et options utilisés avec la commande.

La correspondance est une correspondance de type « contient », ce qui signifie que si vous spécifiez `-c` comme terme de correspondance, les processus avec des paramètres tels que `-config` correspondent à ce terme.

- `drop_original_metrics` : facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.

L' CloudWatch agent n'utilise qu'une seule de ces méthodes, même si vous incluez plusieurs des sections ci-dessus. Si vous spécifiez plusieurs sections, l' CloudWatch agent utilise la `pid_file` section si elle est présente. Si ce n'est pas le cas, il utilise la section `exe`.

Sur les serveurs Linux, les chaînes que vous spécifiez dans une section `pattern` ou `exe` sont évaluées comme des expressions régulières. Sur les serveurs exécutant Windows Server, ces chaînes sont évaluées comme des requêtes WMI. Un exemple serait `pattern: "%apache%"`. Pour plus d'informations, consultez Opérateur LIKE.

Quelle que soit la méthode utilisée, vous pouvez inclure un paramètre `metrics_collection_interval` facultatif, qui spécifie la fréquence en secondes pour collecter ces métriques. Si vous omettez ce paramètre, la valeur par défaut de 60 secondes est utilisée.

Dans les exemples des sections suivantes, la section `procstat` est la seule section incluse dans la section `metrics_collected` du fichier de configuration de l'agent. Les fichiers de configuration réelle peuvent également inclure d'autres sections dans `metrics_collected`. Pour de plus amples informations, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

Configuration avec pid_file

L'exemple de section `procstat` suivant surveille les processus qui créent les fichiers PID `example1.pid` et `example2.pid`. Diverses métriques sont collectées auprès de chaque processus. Les métriques collectées auprès du processus qui crée `example2.pid` sont collectées toutes les 10 secondes, tandis que les métriques collectées auprès du processus `example1.pid` sont collectées toutes les 60 secondes, par défaut.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pid_file": "/var/run/example1.pid",
                    "measurement": [
                        "cpu_usage",
                        "memory_rss"
                    ]
                },
                {
                    "pid_file": "/var/run/example2.pid",
                    "measurement": [
                        "read_bytes",
                        "read_count",
                        "write_bytes"
                    ],
                    "metrics_collection_interval": 10
                }
            ]
        }
    }
}
```

Configuration avec exe

L'exemple de section `procstat` suivant surveille tous les processus avec des noms correspondant aux chaînes `agent` ou `plugin`. Les mêmes métriques sont collectées auprès de chaque processus.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "exe": "agent",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                },
                {
```

```
                    "exe": "plugin",
                    "measurement": [
                        "cpu_time",
                        "cpu_time_system",
                        "cpu_time_user"
                    ]
                }
            ]
        }
    }
}
```

Configuration avec un modèle

L'exemple de section `procstat` suivant surveille tous les processus avec des lignes de commande correspondant aux chaînes `config` ou `-c`. Les mêmes métriques sont collectées auprès de chaque processus.

```
{
    "metrics": {
        "metrics_collected": {
            "procstat": [
                {
                    "pattern": "config",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                },
                {
                    "pattern": "-c",
                    "measurement": [
                        "rlimit_memory_data_hard",
                        "rlimit_memory_data_soft",
                        "rlimit_memory_stack_hard",
                        "rlimit_memory_stack_soft"
                    ]
                }
            ]
        }
    }
```

```
}
```

Métriques collectées par procstat

Le tableau suivant répertorie les métriques que vous pouvez collecter avec le plugin `procstat`.

L' CloudWatch agent ajoute `procstat` au début des noms de métriques suivants. La syntaxe varie selon qu'elles ont été collectées à partir d'un serveur Linux ou d'un serveur exécutant Windows Server. Par exemple, la métrique `cpu_time` s'affiche en tant que `procstat_cpu_time` lorsqu'elle est collectée auprès de Linux, et en tant que `procstat cpu_time` lorsqu'elle est collectée auprès de Windows Server.

Nom de la métrique	Disponible sur	Description
`cpu_time`	Linux	Durée pendant laquelle le processus utilise l'UC. Cette métrique est présentée en centièmes de seconde. Unité : nombre
`cpu_time_guest`	Linux	Durée pendant laquelle le processus est en mode invité. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune

Nom de la métrique	Disponible sur	Description
`cpu_time_guest_nice`	Linux	Durée pendant laquelle le processus s'exécute en tant qu'invité nice. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune
`cpu_time_idle`	Linux	Durée pendant laquelle le processus est en mode inactif. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune

Nom de la métrique	Disponible sur	Description
`cpu_time_iowait`	Linux	Durée pendant laquelle le processus est en attente d'exécution d'opérations d'I/O. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune
`cpu_time_irq`	Linux	Durée pendant laquelle le processus prend en charge des interruptions. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune

Nom de la métrique	Disponible sur	Description
`cpu_time_nice`	Linux	Durée pendant laquelle le processus est en mode nice. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune
`cpu_time_soft_irq`	Linux	Durée pendant laquelle le processus prend en charge des interruptions logicielles. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune

Nom de la métrique	Disponible sur	Description
`cpu_time_steal`	Linux	Durée d'exécution sous d'autres systèmes d'exploit ation dans un environnement virtualisé. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune

Nom de la métrique	Disponible sur	Description
`cpu_time_stolen`	Linux, Windows Server	Durée pendant laquelle le processus subit un vol de temps, c'est-à-dire qu'il se consacre à d'autres systèmes d'exploit ation dans un environnement virtualisé. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : aucune
`cpu_time_system`	Linux, Windows Server, macOS	Durée pendant laquelle le processus est en mode système. Cette métrique est présentée en centièmes de seconde. Type : flottant Unité : nombre

Nom de la métrique	Disponible sur	Description
`cpu_time_user`	Linux, Windows Server, macOS	Durée pendant laquelle le processus est en mode utilisateur. Cette métrique est présentée en centièmes de seconde. Unité : nombre
`cpu_usage`	Linux, Windows Server, macOS	Durée, en pourcentage, pendant laquelle le processus est actif dans n'importe quelle capacité. Unité : pourcentage
`memory_data`	Linux, macOS	Quantité de mémoire utilisée par le processus pour les données. Unité : octets

Nom de la métrique	Disponible sur	Description
`memory_locked`	Linux, macOS	Quantité de mémoire que le processus a verrouillé. Unité : octets
`memory_rss`	Linux, Windows Server, macOS	Quantité de mémoire réelle (résident défini) que le processus utilise. Unité : octets
`memory_stack`	Linux, macOS	Quantité de mémoire de pile utilisée par le processus. Unité : octets
`memory_swap`	Linux, macOS	Quantité de mémoire d'échange utilisée par le processus. Unité : octets

Nom de la métrique	Disponible sur	Description
`memory_vms`	Linux, Windows Server, macOS	Quantité de mémoire virtuelle utilisée par le processus. Unité : octets
`num_fds`	Linux	Nombre de descripteurs de fichiers ouverts par ce processus. Unité : aucune
`num_threads`	Linux, Windows, macOS	Nombre de threads dans ce processus. Unité : aucune
`pid`	Linux, Windows Server, macOS	Identifiant de processus (ID). Unité : aucune

Nom de la métrique	Disponible sur	Description
`pid_count`	Linux, Windows Server, macOS	Nombre d'ID de processus associés au processus. Sur les serveurs Linux et les ordinateurs macOS, le nom complet de cette métrique est `procstat_lookup_pid_count` ; sur Windows Server, ce nom est `procstat_lookup pid_count` . Unité : aucune
`read_bytes`	Linux, Windows Server	Nombre d'octets lus par le processus sur les disques. Unité : octets

Nom de la métrique	Disponible sur	Description
`write_bytes`	Linux, Windows Server	Nombre d'octets écrits par le processus sur les disques. Unité : octets
`read_count`	Linux, Windows Server	Nombre d'opérations de lecture sur disque exécutées par le processus. Unité : aucune
`rlimit_realtime_priority_hard`	Linux	Limite stricte de la priorité en temps réel qui peut être définie pour ce processus. Unité : aucune
`rlimit_realtime_priority_soft`	Linux	Limite flexible de la priorité en temps réel qui peut être définie pour ce processus. Unité : aucune

Nom de la métrique	Disponible sur	Description
`rlimit_signals_pending_hard`	Linux	Limite stricte du nombre maximum de signaux pouvant être mis en file d'attente par ce processus. Unité : aucune
`rlimit_signals_pending_soft`	Linux	Limite flexible du nombre maximum de signaux pouvant être mis en file d'attente par ce processus. Unité : aucune
`rlimit_nice_priority_hard`	Linux	Limite stricte de la priorité nice maximale qui peut être définie par ce processus. Unité : aucune

Nom de la métrique	Disponible sur	Description
`rlimit_nice_priority_soft`	Linux	Limite flexible de la priorité nice maximale qui peut être définie par ce processus. Unité : aucune
`rlimit_num_fds_hard`	Linux	Limite stricte du nombre maximum de descripteurs de fichiers qu'un processus peut ouvrir. Unité : aucune
`rlimit_num_fds_soft`	Linux	Limite flexible du nombre maximum de descripteurs de fichiers qu'un processus peut ouvrir. Unité : aucune

Nom de la métrique	Disponible sur	Description
write_count	Linux, Windows Server	Nombre d'opérations d'écriture sur disque exécutées par le processus. Unité : aucune
involuntary_context_switches	Linux	Nombre de fois que le contexte du processus a été involontairement commuté. Unité : aucune
voluntary_context_switches	Linux	Nombre de fois que le contexte du processus a été volontairement commuté. Unité : aucune
realtime_priority	Linux	Utilisation actuelle de la priorité en temps réel du processus. Unité : aucune

Nom de la métrique	Disponible sur	Description
`nice_priority`	Linux	Utilisation actuelle de la priorité nice du processus. Unité : aucune
`signals_pending`	Linux	Nombre de signaux en attente de traitement par le processus. Unité : aucune
`rlimit_cpu_time_hard`	Linux	Limite de ressource s de temps UC irréversi ble pour le processus. Unité : aucune
`rlimit_cpu_time_soft`	Linux	Limite de ressource s de temps UC temporair e pour le processus. Unité : aucune

Nom de la métrique	Disponible sur	Description
`rlimit_file_locks_hard`	Linux	Limite de ressources de verrouillages de fichier irréversible pour le processus. Unité : aucune
`rlimit_file_locks_soft`	Linux	Limite de ressources de verrouillages de fichier temporaire pour le processus. Unité : aucune
`rlimit_memory_data_hard`	Linux	Limite de ressource irréversible sur le processus pour la mémoire utilisée pour les données. Unité : octets

Nom de la métrique	Disponible sur	Description
`rlimit_memory_data_soft`	Linux	Limite de ressource temporaire sur le processus pour la mémoire utilisée pour les données. Unité : octets
`rlimit_memory_locked_hard`	Linux	Limite de ressource irréversible sur le processus pour la mémoire verrouillée. Unité : octets
`rlimit_memory_locked_soft`	Linux	Limite de ressource temporaire sur le processus pour la mémoire verrouillée. Unité : octets

Nom de la métrique	Disponible sur	Description
`rlimit_memory_rss_hard`	Linux	Limite de ressource irréversible sur le processus pour la mémoire physique. Unité : octets
`rlimit_memory_rss_soft`	Linux	Limite de ressource temporaire sur le processus pour la mémoire physique. Unité : octets
`rlimit_memory_stack_hard`	Linux	Limite de ressources irréversible pour la pile de processus. Unité : octets
`rlimit_memory_stack_soft`	Linux	Limite de ressources temporaire pour la pile de processus. Unité : octets

Nom de la métrique	Disponible sur	Description
`rlimit_memory_vms_hard`	Linux	Limite de ressource irréversible sur le processus pour la mémoire virtuelle. Unité : octets
`rlimit_memory_vms_soft`	Linux	Limite de ressource temporaire sur le processus pour la mémoire virtuelle. Unité : octets

Affichage des métriques de processus importées par l' CloudWatch agent

Après avoir importé des métriques de processus dans CloudWatch, vous pouvez les visualiser sous forme de graphiques chronologiques et créer des alarmes qui peuvent surveiller ces métriques et vous avertir si elles dépassent un seuil que vous spécifiez. La procédure suivante montre comment afficher les métriques de processus sous la forme d'un graphique de séries chronologiques. Pour plus d'informations sur la configuration des alertes, consultez Utilisation des CloudWatch alarmes Amazon.

Pour afficher les métriques du processus dans la CloudWatch console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.

3. Sélectionnez l'espace de noms pour les métriques collectées par l'agent. Par défaut, il s'agit de CWagent, mais vous avez peut-être spécifié un espace de noms différent dans le fichier de configuration de l' CloudWatch agent.
4. Sélectionnez une dimension de métrique (Per-Instance Metrics (Métriques par instance) par exemple).
5. L'onglet All metrics (Toutes les métriques) affiche toutes les métriques pour cette dimension dans l'espace de nom. Vous pouvez effectuer les actions suivantes :

   a. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d'en-tête du tableau.

   b. Pour trier le tableau, utilisez l'en-tête de colonne.

   c. Pour filtrer par ressource, sélectionnez l'ID de ressource, puis Add to search Ajouter à la recherche).

   d. Pour filtrer par métrique, choisissez le nom de la métrique, puis Add to search (Ajouter à la recherche).
6. (Facultatif) Pour ajouter ce graphique à un CloudWatch tableau de bord, choisissez Actions, puis Ajouter au tableau de bord.

Récupération de métriques personnalisées avec StatsD

Vous pouvez récupérer des métriques personnalisées supplémentaires à partir de vos applications ou services à l'aide de l' CloudWatchagent associé au `StatsD` protocole. StatsD est une solution open source populaire qui peut collecter des métriques à partir d'une grande variété d'applications. StatsD est particulièrement utile pour instrumenter vos propres métriques. Pour un exemple d'utilisation conjointe de l' CloudWatch agent et de StatsD, consultez Comment mieux surveiller les métriques personnalisées de votre application à l'aide d'Amazon CloudWatch Agent.

`StatsD`est pris en charge à la fois sur les serveurs Linux et sur les serveurs exécutant Windows Server. CloudWatch prend en charge le `StatsD` format suivant :

```
MetricName:value|type|@sample_rate|#tag1:
  value,tag1...
```

- `MetricName` – Une chaîne sans signe deux-points, sans barre, sans caractère # ou @.
- `value` – Il peut s'agir d'un nombre entier ou à virgule flottante.

- `type` – Spécifie `c` pour compteur, `g` pour jauge, `ms` pour minuteur, `h` pour histogramme ou `s` pour définir.
- `sample_rate` – (Facultatif) Une valeur à virgule flottante comprise entre 0 et 1, inclus. À réserver aux métriques de compteur, d'histogramme et de minuteur. La valeur par défaut est 1 (échantillonnage 100 % du temps).
- `tags`— (Facultatif) Liste de balises séparées par des virgules. `StatsD`les balises sont similaires aux dimensions de CloudWatch. Utilisez le signe deux-points pour les balises clé/valeur, telles que `env:prod`.

Vous pouvez utiliser n'importe quel `StatsD` client utilisant ce format pour envoyer les métriques à l' CloudWatch agent. Pour plus d'informations sur certains des `StatsD` clients disponibles, consultez la page du client StatsD sur. GitHub

Pour collecter ces métriques personnalisées, ajoutez une ligne `"statsd": {}` à la section `metrics_collected` du fichier de configuration de l'agent. Vous pouvez ajouter cette ligne manuellement. Si vous utilisez l'assistant pour créer le fichier de configuration, cette opération est faite à votre place. Pour de plus amples informations, consultez Création du fichier de configuration de CloudWatch l'agent.

La configuration par défaut du protocole `StatsD` fonctionne pour la plupart des utilisateurs. Il existe des champs facultatifs que vous pouvez ajouter à la section statsd du fichier de configuration de l'agent, le cas échéant :

- `service_address`— L'adresse de service que l' CloudWatch agent doit écouter. Le format est le suivant *`ip`*`:`*`port`*. Si vous omettez l'adresse IP, l'agent écoute sur toutes les interfaces disponibles. Seul le format UDP est pris en charge, vous n'avez donc pas besoin de spécifier un préfixe UDP.

  La valeur par défaut est `:8125`.
- `metrics_collection_interval` – Indique la fréquence, en secondes, d'exécution et de collecte des métriques par le plugin `StatsD`. La valeur par défaut est de 10 secondes. La plage est comprise entre 1 et 172 000.
- `metrics_aggregation_interval`— À quelle fréquence, en secondes, CloudWatch agrège les métriques en points de données uniques. La valeur par défaut est de 60 secondes.

  Par exemple, si la valeur `metrics_collection_interval` est égale à 10 et `metrics_aggregation_interval` à 60, CloudWatch collecte des données toutes les 10

secondes. Après chaque minute, les six relevés de données de cette minute sont agrégés en un seul point de données, qui est envoyé à CloudWatch.

La plage est comprise entre 0 et 172 000. Le fait de définir `metrics_aggregation_interval` sur zéro désactive le regroupement de métriques `StatsD`.

- `allowed_pending_messages` – Nombre de messages UDP autorisés à être placés en file d'attente. Lorsque la file d'attente est pleine, le serveur StatsD commence à supprimer des paquets. La valeur par défaut est 10 000.
- `drop_original_metrics` : facultatif. Si vous utilisez le champ `aggregation_dimensions` de la section `metrics` pour regrouper les métriques dans des résultats agrégés, l'agent envoie par défaut les métriques agrégées et les métriques d'origine qui sont séparées pour chaque valeur de la dimension. Si vous ne souhaitez pas que les mesures d'origine soient envoyées à CloudWatch, vous pouvez spécifier ce paramètre avec une liste de mesures. Les mesures spécifiées avec ce paramètre ne sont pas signalées à CloudWatch. Au lieu de cela, seules les métriques agrégées sont signalées. Cela réduit le nombre de métriques collectées par l'agent, ce qui réduit vos coûts.

Voici un exemple de la section statsd du fichier de configuration de l'agent, qui utilise le port par défaut et des intervalles de collecte et de regroupement personnalisés.

```
{
   "metrics":{
      "metrics_collected":{
         "statsd":{
            "service_address":":8125",
            "metrics_collection_interval":60,
            "metrics_aggregation_interval":300
         }
      }
   }
}
```

Afficher les métriques StatsD importées par l'agent CloudWatch

Après avoir importé les métriques StatsD dans CloudWatch, vous pouvez les visualiser sous forme de graphiques chronologiques et créer des alarmes qui peuvent surveiller ces métriques et vous avertir si elles dépassent un seuil que vous spécifiez. La procédure suivante montre comment afficher les métriques StatsD sous la forme d'un graphique de séries chronologiques. Pour plus d'informations sur la configuration des alertes, consultez Utilisation des CloudWatch alarmes Amazon.

Pour afficher les métriques StatsD dans la console CloudWatch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sélectionnez l'espace de noms pour les métriques collectées par l'agent. Par défaut, il s'agit de CWagent, mais vous avez peut-être spécifié un espace de noms différent dans le fichier de configuration de l' CloudWatch agent.
4. Sélectionnez une dimension de métrique (Per-Instance Metrics (Métriques par instance) par exemple).
5. L'onglet All metrics (Toutes les métriques) affiche toutes les métriques pour cette dimension dans l'espace de nom. Vous pouvez effectuer les actions suivantes :

   a. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d’en-tête du tableau.

   b. Pour trier le tableau, utilisez l’en-tête de colonne.

   c. Pour filtrer par ressource, sélectionnez l'ID de ressource, puis Add to search Ajouter à la recherche).

   d. Pour filtrer par métrique, choisissez le nom de la métrique, puis Add to search (Ajouter à la recherche).
6. (Facultatif) Pour ajouter ce graphique à un CloudWatch tableau de bord, choisissez Actions, puis Ajouter au tableau de bord.

Récupération de métriques personnalisées avec collectd

Vous pouvez récupérer des métriques supplémentaires à partir de vos applications ou services à l'aide de l' CloudWatchagent avec le protocole collectd, qui n'est pris en charge que sur les serveurs Linux. collectd est une solution open source populaire dotée de plugins permettant de recueillir des statistiques système pour une grande variété d'applications. En combinant les métriques système que l' CloudWatch agent peut déjà collecter avec les métriques supplémentaires collectées, vous pouvez mieux surveiller, analyser et dépanner vos systèmes et applications. Pour de plus amples informations sur collectd, consultez collectd - Démon de collecte des statistiques système.

Vous utilisez le logiciel collectd pour envoyer les métriques à l' CloudWatch agent. Pour les métriques collectées, l' CloudWatch agent agit en tant que serveur tandis que le plugin collectd agit en tant que client.

Le logiciel collectd n'est pas installé automatiquement sur chaque serveur. Sur un serveur exécutant Amazon Linux 2, procédez comme suit pour installer collectd

```
sudo amazon-linux-extras install collectd
```

Pour obtenir des informations sur l'installation de collectd sur d'autres systèmes, consultez la page de téléchargement de collectd.

Pour collecter ces métriques personnalisées, ajoutez une ligne "collectd": {} à la section metrics_collected du fichier de configuration de l'agent. Vous pouvez ajouter cette ligne manuellement. Si vous utilisez l'Assistant pour créer le fichier de configuration, cette opération est faite pour vous. Pour de plus amples informations, consultez Création du fichier de configuration de CloudWatch l'agent.

Des paramètres facultatifs sont également disponibles. Si vous utilisez collectd et que vous n'utilisez pas `/etc/collectd/auth_file` comme votre collectd_auth_file, vous devez définir certaines de ces options.

- service_address : adresse du service que l' CloudWatch agent doit écouter. Le format est le suivant "`udp://`*`ip`*`:`*`port`*. L'argument par défaut est `udp://127.0.0.1:25826`.
- name_prefix : Un préfixe à attacher au début du nom de chaque métrique collectd. L'argument par défaut est `collectd_`. La longueur maximale est de 255 caractères.
- collectd_security_level : définit le niveau de sécurité pour la communication réseau. La valeur par défaut est encrypt.

  encrypt spécifie que seules les données chiffrées sont acceptées. sign spécifie que seules les données signées et chiffrées sont acceptées. none spécifie que toutes les données sont acceptées. Si vous spécifiez une valeur pour collectd_auth_file, les données chiffrées sont déchiffrées si cela est possible.

  Pour plus d'informations, consultez Client setup (Configuration de client) et Possible interactions (Interactions possibles) dans le Wiki collectd.
- collectd_auth_file Définit un fichier dans lequel les noms d'utilisateur sont mappés aux mots de passe. Ces mots de passe sont utilisés pour vérifier les signatures et pour déchiffrer des paquets réseau chiffrés. Le cas échéant, les données signées sont vérifiées et les paquets chiffrés sont déchiffrés. Dans le cas contraire, les données signées sont acceptées sans vérification de la signature et les données chiffrées ne peuvent pas être déchiffrées.

L’argument par défaut est `/etc/collectd/auth_file`.

Si collectd_security_level est défini sur none, l'opération est facultative. Si vous définissez collectd_security_level sur `encrypt` ou sign, vous devez spécifier collectd_auth_file.

Pour le format du fichier d'authentification, chaque ligne est un nom d'utilisateur suivi de deux points et d'un nombre quelconque d'espaces, puis du mot de passe. Par exemple :

```
user1: user1_password

user2: user2_password
```

- collectd_typesdb : liste d'un ou de plusieurs fichiers qui contiennent les descriptions des ensembles de données. La liste doit être entourée d'accolades, même si la liste ne comprend qu'une seule entrée. Chaque entrée de la liste doit figurer entre guillemets doubles. Si la liste contient plusieurs entrées, séparez-les par des virgules. La valeur par défaut sur les serveurs Linux est `["/usr/share/collectd/types.db"]`. La valeur par défaut sur les ordinateurs macOs dépend de la version de collectd. Par exemple, `["/usr/local/Cellar/collectd/5.12.0/share/collectd/types.db"]`.

  Pour plus d’informations, consultez https://www.collectd.org/documentation/manpages/types.db.html.
- metrics_aggregation_interval : fréquence à laquelle, en secondes, CloudWatch agrège les métriques en points de données uniques. Le durée par défaut est 60 secondes. La plage est comprise entre 0 et 172 000. La définir à zéro désactive le regroupement de métriques collectd.

L'exemple suivant illustre la section collectd du fichier de configuration d'agent.

```
{
   "metrics":{
      "metrics_collected":{
         "collectd":{
            "name_prefix":"My_collectd_metrics_",
            "metrics_aggregation_interval":120
         }
      }
   }
}
```

Afficher les métriques collectées importées par l'agent CloudWatch

Après avoir importé les métriques collectées dans CloudWatch, vous pouvez les visualiser sous forme de graphiques chronologiques et créer des alarmes qui peuvent surveiller ces métriques et vous avertir si elles dépassent un seuil que vous spécifiez. La procédure suivante montre comment afficher les métriques collectd sous la forme d'un graphique de séries chronologiques. Pour plus d'informations sur la configuration des alertes, consultez Utilisation des CloudWatch alarmes Amazon.

Pour afficher les métriques collectées dans la console CloudWatch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sélectionnez l'espace de noms pour les métriques collectées par l'agent. Par défaut, il s'agit de CWagent, mais vous avez peut-être spécifié un espace de noms différent dans le fichier de configuration de l' CloudWatch agent.
4. Sélectionnez une dimension de métrique (Per-Instance Metrics (Métriques par instance) par exemple).
5. L'onglet All metrics (Toutes les métriques) affiche toutes les métriques pour cette dimension dans l'espace de nom. Vous pouvez effectuer les actions suivantes :

    a. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d’en-tête du tableau.

    b. Pour trier le tableau, utilisez l’en-tête de colonne.

    c. Pour filtrer par ressource, sélectionnez l'ID de ressource, puis Add to search Ajouter à la recherche).

    d. Pour filtrer par métrique, choisissez le nom de la métrique, puis Add to search (Ajouter à la recherche).
6. (Facultatif) Pour ajouter ce graphique à un CloudWatch tableau de bord, choisissez Actions, puis Ajouter au tableau de bord.

Configuration et configuration de la collecte de mesures Prometheus sur des instances Amazon EC2

Les sections suivantes expliquent comment installer l' CloudWatch agent avec le système de surveillance Prometheus sur les instances EC2 et comment configurer l'agent pour détecter des

cibles supplémentaires. Elles fournissent également des didacticiels pour configurer des exemples d'applications à utiliser pour les tests avec la surveillance Prometheus.

Pour plus d'informations sur les systèmes d'exploitation pris en charge par l' CloudWatch agent, voir Collectez des métriques, des journaux et des traces avec l' CloudWatch agent

Exigences de groupe de sécurité VPC

Si vous utilisez un VPC, les conditions suivantes s'appliquent.

- Les règles d'entrée des groupes de sécurité pour les charges de travail Prometheus doivent ouvrir les ports Prometheus à l'agent pour récupérer les métriques Prometheus par CloudWatch l'adresse IP privée.
- Les règles de sortie du groupe de sécurité pour l' CloudWatch agent doivent permettre à l'agent de se connecter au CloudWatch port des charges de travail Prometheus via une adresse IP privée.

Rubriques

- Étape 1 : Installation de l' CloudWatch agent
- Étape 2 : Récupérer les sources Prometheus et importer des métriques
- Exemple : Paramétrez des exemples de charges de travail Java/JMX pour les tests de métrique Prometheus

Étape 1 : Installation de l' CloudWatch agent

La première étape consiste à installer l' CloudWatch agent sur l'instance EC2. Pour obtenir des instructions, veuillez consulter Installation de l' CloudWatch agent.

Étape 2 : Récupérer les sources Prometheus et importer des métriques

L' CloudWatch agent chargé de surveiller Prometheus a besoin de deux configurations pour récupérer les métriques Prometheus. L'une concerne les configurations standard Prometheus, comme décrit dans <scrape_config> dans la documentation Prometheus. L'autre concerne la configuration de l' CloudWatch agent.

Configuration de récupération Prometheus

L' CloudWatch agent prend en charge les configurations standard de Prometheus scrape, comme indiquéhttps://prometheus.io/docs/prometheus/latest/configuration/configuration/#scrape_config

<scrape_config>dans la documentation de Prometheus. Vous pouvez modifier cette section pour mettre à jour les configurations déjà présentes dans ce fichier et ajouter des cibles de récupération Prometheus supplémentaires. Un exemple de fichier de configuration contient les lignes de configuration globale suivantes :

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
- job_name: MY_JOB
  sample_limit: 10000
  file_sd_configs:
    - files: ["C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_1.yaml",
 "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\\prometheus_sd_2.yaml"]
```

La section `global` spécifie les paramètres valides dans tous les contextes de configuration. Ils servent également de valeurs par défaut pour d'autres sections de configuration. Elle contient les paramètres suivants :

- `scrape_interval` – Définit la fréquence à laquelle récupérer les cibles.
- `scrape_timeout` – Définit le temps d'attente avant l'expiration d'une requête de récupération.

La section `scrape_configs` spécifie un ensemble de cibles et de paramètres qui définissent comment les récupérer. Elle contient les paramètres suivants :

- `job_name` – Nom de tâche attribué par défaut pour récupérer les métriques.
- `sample_limit` – Limite par récupération du nombre d'échantillons récupérés qui seront acceptés.
- `file_sd_configs` – Liste des configurations de découverte de service de fichiers. Il lit un ensemble de fichiers contenant une liste de zéro ou plusieurs configurations statiques. La section `file_sd_configs` contient un paramètre `files` qui définit des modèles pour les fichiers à partir desquels les groupes cibles sont extraits.

L' CloudWatch agent prend en charge les types de configuration de découverte de services suivants.

`static_config` Permet de spécifier une liste de cibles et un ensemble d'étiquettes commun pour elles. C'est la façon canonique de spécifier des cibles statiques dans une configuration de récupération.

Voici un exemple de configuration statique pour récupérer les métriques Prometheus à partir d'un hôte local. Les métriques peuvent également être récupérées à partir d'autres serveurs si le port Prometheus est ouvert sur le serveur sur lequel l'agent s'exécute.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_sd_1.yaml
- targets:
    - 127.0.0.1:9404
  labels:
    key1: value1
    key2: value2
```

Cet exemple contient les paramètres suivants :

- `targets` – Cibles récupérées par la configuration statique.
- `labels` – Étiquettes affectées à toutes les mesures récupérées des cibles.

`ec2_sd_config` Permet de récupérer des cibles de raclage à partir d'instances Amazon EC2. Voici un exemple de `ec2_sd_config` pour récupérer les métriques Prometheus à partir d'une liste d'instances EC2. Les ports Prometheus de ces instances doivent s'ouvrir sur le serveur sur lequel CloudWatch l'agent s'exécute. Le rôle IAM pour l'instance EC2 sur laquelle l' CloudWatch agent s'exécute doit inclure l'`ec2:DescribeInstance`autorisation. Par exemple, vous pouvez associer la politique gérée AmazonEC2 ReadOnlyAccess à l'instance qui exécute l' CloudWatch agent.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: MY_JOB
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - i-98765432109876543
              - i-12345678901234567
```

Cet exemple contient les paramètres suivants :

- `region`— AWS Région dans laquelle se trouve l'instance EC2 cible. Si vous laissez ce champ vide, la région à partir des métadonnées d'instance est utilisée.
- `port` – Port à partir duquel récupérer les métriques.
- `filters` – Filtres facultatifs à utiliser pour filtrer la liste des instances. Cet exemple de filtre basé sur les ID d'instance EC2. Pour plus de critères sur lesquels vous pouvez filtrer, consultez DescribeInstances.

CloudWatch configuration de l'agent pour Prometheus

Le fichier de configuration de l' CloudWatch agent comprend des `prometheus` sections à la fois sous `logs` et`metrics_collected`. Il inclut les paramètres suivants.

- cluster_name – Spécifie le nom du cluster à ajouter en tant qu'étiquette dans l'évènement du journal. Ce champ est facultatif.
- log_group_name – Spécifie le nom du groupe de journaux pour les métriques Prometheus récupérées.
- prometheus_config_path – Spécifie le chemin d'accès du fichier de configuration de récupération Prometheus.
- emf_processor – Spécifie la configuration du processeur de format métrique incorporé. Pour de plus amples informations sur le format de métriques intégré, consultez Intégration de métriques dans les journaux.

  La section `emf_processor` peut contenir les paramètres suivants :

  - metric_declaration_dedup – Il est défini sur « true », la fonction de déduplication pour les métriques de format de métrique incorporée est activée.
  - metric_namespace — Spécifie l'espace de noms des métriques pour les métriques émises. CloudWatch
  - metric_unit – Spécifie le nom de la métrique : mappage d'unité métrique. Pour plus d'informations sur les unités de mesure prises en charge, consultez MetricDatum.
  - metric_declaration – Ce sont des sections qui spécifient le tableau de journaux avec le format de métrique intégré à générer. Il existe des `metric_declaration` sections pour chaque source Prometheus à partir de laquelle CloudWatch l'agent importe par défaut. Chacune de ces sections comprend les champs suivants :

    - `source_labels` spécifie la valeur des étiquettes qui sont vérifiées par la ligne `label_matcher`.

- `label_matcher` est une expression régulière qui vérifie la valeur des étiquettes répertoriées dans `source_labels`. Les métriques correspondantes sont activées pour être incluses dans le format de métrique intégré envoyé à CloudWatch.
- `metric_selectors`est une expression régulière qui spécifie les métriques à collecter et à envoyer CloudWatch.
- `dimensions`est la liste des étiquettes à utiliser comme CloudWatch dimensions pour chaque métrique sélectionnée.

Voici un exemple de configuration d' CloudWatch agent pour Prometheus.

```
{
   "logs":{
      "metrics_collected":{
         "prometheus":{
            "cluster_name":"prometheus-cluster",
            "log_group_name":"Prometheus",
            "prometheus_config_path":"C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus.yaml",
            "emf_processor":{
               "metric_declaration_dedup":true,
               "metric_namespace":"CWAgent-Prometheus",
               "metric_unit":{
                  "jvm_threads_current": "Count",
                  "jvm_gc_collection_seconds_sum": "Milliseconds"
               },
               "metric_declaration":[
                  {
                     "source_labels":[
                        "job", "key2"
                     ],
                     "label_matcher":"MY_JOB;^value2",
                     "dimensions":[
                        [
                           "key1", "key2"
                        ],
                        [
                           "key2"
                        ]
                     ],
                     "metric_selectors":[
                        "^jvm_threads_current$",
```

```
                        "^jvm_gc_collection_seconds_sum$"
                    ]
                }
            ]
        }
    }
  }
}
```

L'exemple précédent montre comment configurer une section de format de métrique intégrée à envoyer en tant qu'événement de journaux si les conditions suivantes sont remplies :

- La valeur de l'étiquette `job` est `MY_JOB`
- La valeur de l'étiquette `key2` est `value2`
- Les métriques Prometheus `jvm_threads_current` et `jvm_gc_collection_seconds_sum` contiennent les étiquettes `job` et `key2`.

L'événement de journal envoyé inclut la section en surbrillance suivante.

```
{
    "CloudWatchMetrics": [
        {
            "Metrics": [
                {
                    "Unit": "Count",
                    "Name": "jvm_threads_current"
                },
                {
                    "Unit": "Milliseconds",
                    "Name": "jvm_gc_collection_seconds_sum"
                }
            ],
            "Dimensions": [
                [
                    "key1",
                    "key2"
                ],
                [
                    "key2"
                ]
```

```
            ],
            "Namespace": "CWAgent-Prometheus"
        }
    ],
    "ClusterName": "prometheus-cluster",
    "InstanceId": "i-0e45bd06f196096c8",
    "Timestamp": "1607966368109",
    "Version": "0",
    "host": "EC2AMAZ-PDDOIUM",
    "instance": "127.0.0.1:9404",
    "jvm_threads_current": 2,
    "jvm_gc_collection_seconds_sum": 0.006000000000000002,
    "prom_metric_type": "gauge",
    ...
}
```

Exemple : Paramétrez des exemples de charges de travail Java/JMX pour les tests de métrique Prometheus

JMX Exporter est un exportateur Prometheus officiel qui peut récupérer et exposer des mBeans JMX en tant que métriques Prometheus. Pour plus d'informations, consultez prometheus/jmx_exporter.

L' CloudWatch agent peut collecter des métriques Prometheus prédéfinies à partir de Java Virtual Machine (JVM), Hjava et Tomcat (Catalina), à partir d'un exportateur JMX sur des instances EC2.

Étape 1 : Installation de l' CloudWatch agent

La première étape consiste à installer l' CloudWatch agent sur l'instance EC2. Pour obtenir des instructions, veuillez consulter Installation de l' CloudWatch agent.

Étape 2 : Démarrage de la charge de travail Java/JMX

L'étape suivante consiste à démarrer la charge de travail Java/JMX.

Tout d'abord, téléchargez le dernier fichier jar du JMX Exporter à partir de l'emplacement suivant : prometheus/jmx_exporter.

Utilisez le jar pour votre exemple d'application

Les exemples de commandes dans les sections suivantes utilisent `SampleJavaApplication-1.0-SNAPSHOT.jar` comme fichier jar. Remplacez les parties des commandes par le jar de votre application.

Préparez la configuration de JMX Exporter

Le fichier `config.yaml` est le fichier de configuration de JMX Exporter. Pour plus d'informations, consultez Configuration dans la documentation JMX Exporter.

Voici un exemple de configuration pour Java et Tomcat.

```
---
lowercaseOutputName: true
lowercaseOutputLabelNames: true

rules:
- pattern: 'java.lang<type=OperatingSystem><>(FreePhysicalMemorySize|
TotalPhysicalMemorySize|FreeSwapSpaceSize|TotalSwapSpaceSize|SystemCpuLoad|
ProcessCpuLoad|OpenFileDescriptorCount|AvailableProcessors)'
  name: java_lang_OperatingSystem_$1
  type: GAUGE

- pattern: 'java.lang<type=Threading><>(TotalStartedThreadCount|ThreadCount)'
  name: java_lang_threading_$1
  type: GAUGE

- pattern: 'Catalina<type=GlobalRequestProcessor, name=\"(\w+-\w+)-(\d+)\"><>(\w+)'
  name: catalina_globalrequestprocessor_$3_total
  labels:
    port: "$2"
    protocol: "$1"
  help: Catalina global $3
  type: COUNTER

- pattern: 'Catalina<j2eeType=Servlet, WebModule=//([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-
a-zA-Z0-9+&@#/%=~_|]), name=([-a-zA-Z0-9+/$%~_-|!.]*), J2EEApplication=none,
 J2EEServer=none><>(requestCount|maxTime|processingTime|errorCount)'
  name: catalina_servlet_$3_total
  labels:
    module: "$1"
    servlet: "$2"
  help: Catalina servlet $3 total
  type: COUNTER

- pattern: 'Catalina<type=ThreadPool, name="(\w+-\w+)-(\d+)"><>(currentThreadCount|
currentThreadsBusy|keepAliveCount|pollerThreadCount|connectionCount)'
  name: catalina_threadpool_$3
  labels:
```

```
    port: "$2"
    protocol: "$1"
  help: Catalina threadpool $3
  type: GAUGE

- pattern: 'Catalina<type=Manager, host=([-a-zA-Z0-9+&@#/%?=~_|!:.,;]*[-a-zA-
Z0-9+&@#/%=~_|]), context=([-a-zA-Z0-9+/$%~_-|!.]*)><>(processingTime|sessionCounter|
rejectedSessions|expiredSessions)'
  name: catalina_session_$3_total
  labels:
    context: "$2"
    host: "$1"
  help: Catalina session $3 total
  type: COUNTER

- pattern: ".*"
```

Démarrez l'application Java avec l'exportateur Prometheus

Démarrez l'exemple d'application Cela émettra des métriques Prometheus sur le port 9404. Veillez à remplacer le point d'entrée `com.gubupt.sample.app.App` avec la bonne information pour votre exemple d'application java.

Sur Linux, entrez la commande suivante.

```
$ nohup java -javaagent:./jmx_prometheus_javaagent-0.14.0.jar=9404:./config.yaml -cp
  ./SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App &
```

Sur Windows, entrez la commande suivante.

```
PS C:\> java -javaagent:.\jmx_prometheus_javaagent-0.14.0.jar=9404:.\config.yaml -cp  .
\SampleJavaApplication-1.0-SNAPSHOT.jar com.gubupt.sample.app.App
```

Vérifiez l'émission des métriques Prometheus

Vérifiez que les métriques Prometheus sont émises.

Sur Linux, entrez la commande suivante.

```
$ curl localhost:9404
```

Sur Windows, entrez la commande suivante.

```
PS C:\> curl  http://localhost:9404
```

Exemple de sortie sous Linux :

```
StatusCode        : 200
StatusDescription : OK
Content           : # HELP jvm_classes_loaded The number of classes that are currently
 loaded in the JVM
                    # TYPE jvm_classes_loaded gauge
                    jvm_classes_loaded 2526.0
                    # HELP jvm_classes_loaded_total The total number of class...
RawContent        : HTTP/1.1 200 OK
                    Content-Length: 71908
                    Content-Type: text/plain; version=0.0.4; charset=utf-8
                    Date: Fri, 18 Dec 2020 16:38:10 GMT

                    # HELP jvm_classes_loaded The number of classes that are
 currentl...
Forms             : {}
Headers           : {[Content-Length, 71908], [Content-Type, text/plain; version=0.0.4;
 charset=utf-8], [Date, Fri, 18
                    Dec 2020 16:38:10 GMT]}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : System.__ComObject
RawContentLength  : 71908
```

Étape 3 : configurer l' CloudWatch agent pour récupérer les métriques de Prometheus

Configurez ensuite la configuration Prometheus Scrape dans CloudWatch le fichier de configuration de l'agent.

Pour paramétrer la configuration de récupération Prometheus pour l'exemple Java/JMX

1. Configurez la configuration pour `file_sd_config` et `static_config`.

   Sur Linux, entrez la commande suivante.

   ```
   $ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus.yaml
   global:
     scrape_interval: 1m
   ```

```
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: [ "/opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml" ]
```

Sur Windows, entrez la commande suivante.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    file_sd_configs:
      - files: [ "C:\\ProgramData\\Amazon\\AmazonCloudWatchAgent\
\prometheus_file_sd.yaml" ]
```

2. Paramétrez la configuration des cibles de récupération.

   Sur Linux, entrez la commande suivante.

```
$ cat /opt/aws/amazon-cloudwatch-agent/var/prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: linux
```

   Sur Windows, entrez la commande suivante.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: windows
```

3. Paramétrez la configuration de récupération Prometheus avec `ec2_sc_config`. Remplacez *`votre-id-instance-ec2`* avec le bon ID d'instance EC2.

Sur Linux, entrez la commande suivante.

```
$ cat .\prometheus.yaml
global:
  scrape_interval: 1m
  scrape_timeout: 10s
scrape_configs:
  - job_name: jmx
    sample_limit: 10000
    ec2_sd_configs:
      - region: us-east-1
        port: 9404
        filters:
          - name: instance-id
            values:
              - your-ec2-instance-id
```

Sur Windows, entrez la commande suivante.

```
PS C:\ProgramData\Amazon\AmazonCloudWatchAgent> cat prometheus_file_sd.yaml
- targets:
  - 127.0.0.1:9404
  labels:
    application: sample_java_app
    os: windows
```

4. Configurez la configuration de CloudWatch l'agent. Commencez par accéder au bon répertoire. Sous Linux, il s'agit de `/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json`. Sous Windows, il s'agit de `C:\ProgramData\Amazon\AmazonCloudWatchAgent\cwagent-config.json`.

   Voici un exemple de configuration avec les métriques Prometheus Java/JHX définies. Assurez-vous de remplacer *`path-to-Prometheus-Scrape-Configuration-file`* par le bon chemin.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
```

```
    "prometheus": {
      "cluster_name": "my-cluster",
      "log_group_name": "prometheus-test",
      "prometheus_config_path": "path-to-Prometheus-Scrape-Configuration-file",
      "emf_processor": {
        "metric_declaration_dedup": true,
        "metric_namespace": "PrometheusTest",
        "metric_unit":{
          "jvm_threads_current": "Count",
          "jvm_classes_loaded": "Count",
          "java_lang_operatingsystem_freephysicalmemorysize": "Bytes",
          "catalina_manager_activesessions": "Count",
          "jvm_gc_collection_seconds_sum": "Seconds",
          "catalina_globalrequestprocessor_bytesreceived": "Bytes",
          "jvm_memory_bytes_used": "Bytes",
          "jvm_memory_pool_bytes_used": "Bytes"
        },
        "metric_declaration": [
          {
            "source_labels": ["job"],
            "label_matcher": "^jmx$",
            "dimensions": [["instance"]],
            "metric_selectors": [
              "^jvm_threads_current$",
              "^jvm_classes_loaded$",
              "^java_lang_operatingsystem_freephysicalmemorysize$",
              "^catalina_manager_activesessions$",
              "^jvm_gc_collection_seconds_sum$",
              "^catalina_globalrequestprocessor_bytesreceived$"
            ]
          },
          {
            "source_labels": ["job"],
            "label_matcher": "^jmx$",
            "dimensions": [["area"]],
            "metric_selectors": [
              "^jvm_memory_bytes_used$"
            ]
          },
          {
            "source_labels": ["job"],
            "label_matcher": "^jmx$",
            "dimensions": [["pool"]],
            "metric_selectors": [
```

```
                "^jvm_memory_pool_bytes_used$"
              ]
            }
          ]
        }
      }
    },
    "force_flush_interval": 5
  }
}
```

5. Redémarrez l' CloudWatch agent en saisissant l'une des commandes suivantes.

   Sur Linux, entrez la commande suivante.

   ```
   sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
   config -m ec2 -s -c file:/opt/aws/amazon-cloudwatch-agent/var/cwagent-config.json
   ```

   Sur Windows, entrez la commande suivante.

   ```
   & "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
    -a fetch-config -m ec2 -s -c file:C:\ProgramData\Amazon\AmazonCloudWatchAgent
   \cwagent-config.json
   ```

Affichage des journaux et métriques Prometheus

Vous pouvez maintenant afficher les métriques Java/JMX collectées.

Pour examiner les métriques de votre application Java/JMX

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans la région où votre cluster s'exécute, choisissez Metrics (Métriques) dans le volet de navigation de gauche. Trouvez l'PrometheusTestespace de noms pour voir les métriques.
3. Pour voir les événements CloudWatch Logs, choisissez Log groups dans le volet de navigation. Les événements sont dans le groupe de journaux prometheus-test.

Installez l' CloudWatch agent à l'aide du module complémentaire Amazon CloudWatch Observability EKS

Le module complémentaire Amazon CloudWatch Observability EKS installe l' CloudWatch agent et l'agent Fluent-bit sur un cluster Amazon EKS, l'observabilité améliorée de Container Insights pour Amazon EKS et les signaux d'CloudWatch application étant activée par défaut. Grâce au module complémentaire, vous pouvez collecter des métriques d'infrastructure, la télémétrie de performance des applications et des journaux de conteneurs à partir du cluster Amazon EKS.

Grâce à Container Insights avec observabilité améliorée pour Amazon EKS, les métriques de Container Insights sont facturées par observation au lieu d'être facturées par métrique stockée ou par journal ingéré. Pour Application Signals, la facturation est basée sur les demandes entrantes adressées à vos applications, les demandes sortantes provenant de vos applications et sur chaque objectif de niveau de service (SLO) configuré. Chaque requête entrante reçue génère un signal d'application, et chaque requête sortante effectuée génère un signal d'application. Chaque SLO crée deux signaux d'application par période de mesure. Pour plus d'informations sur CloudWatch les tarifs, consultez Amazon CloudWatch Pricing.

Le module complémentaire Amazon EKS active Container Insights sur les nœuds de travail Linux et Windows du cluster Amazon EKS. Pour activer Container Insights sous Windows, vous devez utiliser la version 1.5.0 ou ultérieure du module complémentaire Amazon EKS. Actuellement, Application Signals n'est pas pris en charge sous Windows dans les clusters Amazon EKS.

Le module complémentaire Amazon CloudWatch Observability EKS est pris en charge sur les clusters Amazon EKS exécutés avec Kubernetes version 1.23 ou ultérieure.

Lorsque vous installez le module complémentaire, vous devez également accorder des autorisations IAM pour permettre à l' CloudWatch agent d'envoyer des métriques, des journaux et des traces à CloudWatch. Il existe deux façons de procéder :

- Attachez une politique au rôle IAM de vos composants master. Cette option accorde des autorisations aux nœuds de travail auxquels envoyer des données télémétriques. CloudWatch
- Utilisez un rôle IAM pour les comptes de service des pods d'agent et attachez la politique à ce rôle. Cela ne fonctionne que pour les clusters Amazon EKS. Cette option donne CloudWatch accès uniquement aux modules d'agent appropriés.

Option 1 : Installation avec des autorisations IAM sur les composants master

Pour utiliser cette méthode, associez d'abord la politique CloudWatchAgentServerPolicyIAM à vos nœuds de travail en saisissant la commande suivante. Dans cette commande, remplacez-le *my-worker-node-role*par le rôle IAM utilisé par vos nœuds de travail Kubernetes.

```
aws iam attach-role-policy \
--role-name my-worker-node-role \
--policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy
```

Installez ensuite le module complémentaire Amazon CloudWatch Observability EKS. Pour installer le module complémentaire, vous pouvez utiliser la AWS CLI console ou Terraform. AWS CloudFormation

AWS CLI

Pour utiliser le module complémentaire Amazon CloudWatch Observability EKS AWS CLI pour installer le module complémentaire Amazon Observability

Entrez la commande suivante. Remplacez *my-cluster-name* par le nom de votre cluster.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-name my-cluster-name
```

Amazon EKS console

Pour utiliser la console Amazon EKS afin d'ajouter le module complémentaire Amazon CloudWatch Observability EKS

1. Ouvrez la console Amazon EKS à l'adresse https://console.aws.amazon.com/eks/home#/clusters.
2. Dans le panneau de navigation de gauche, choisissez Clusters.
3. Choisissez le nom du cluster pour lequel vous souhaitez configurer le module complémentaire Amazon CloudWatch Observability EKS.
4. Choisissez l'onglet Modules complémentaires.
5. Choisissez Obtenez plus de modules complémentaires.

6. Sur la page Sélectionner des modules complémentaires, procédez comme suit :

   a. Dans la section Amazon EKS-Addons, cochez la case Amazon CloudWatch Observability.

   b. Choisissez Suivant.

7. Sur la page Configurer les paramètres des modules complémentaires sélectionnés, procédez comme suit :

   a. Sélectionnez la version que vous souhaitez utiliser.

   b. Pour Sélectionner un rôle IAM, sélectionnez Hériter du nœud.

   c. (Facultatif) Vous pouvez développer les paramètres de configuration facultatifs. Si vous sélectionnez Remplacer pour la méthode de résolution des conflits, un ou plusieurs des paramètres du module complémentaire existant peuvent être remplacés par les paramètres du module complémentaire Amazon EKS. Si vous n'activez pas cette option et qu'il y a un conflit avec vos paramètres existants, l'opération échoue. Vous pouvez utiliser le message d'erreur qui en résulte pour résoudre le conflit. Avant de sélectionner cette option, assurez-vous que le module complémentaire Amazon EKS ne gère pas les paramètres que vous devez gérer vous-même.

   d. Choisissez Suivant.

8. Sur la page Vérifier et ajouter, choisissez Créer. Une fois l'installation du module complémentaire terminée, vous pouvez voir le module complémentaire installé.

AWS CloudFormation

À utiliser AWS CloudFormation pour installer le module complémentaire Amazon CloudWatch Observability EKS

Remplacez *my-cluster-name* par le nom de votre cluster. Pour plus d'informations, consultez AWS::EKS::Addon.

```
{
    "Resources": {
        "EKSAddOn": {
            "Type": "AWS::EKS::Addon",
            "Properties": {
                "AddonName": "amazon-cloudwatch-observability",
                "ClusterName": "my-cluster-name"
            }
```

```
        }
    }
}
```

Terraform

Pour utiliser Terraform pour installer le module complémentaire Amazon CloudWatch Observability EKS

Remplacez *my-cluster-name* par le nom de votre cluster. Pour plus d'informations, veuillez consulter Ressource : aws_eks_addon (langue française non garantie).

```
resource "aws_eks_addon" "example" {
  addon_name   = "amazon-cloudwatch-observability"
  cluster_name = "my-cluster-name"
}
```

Option 2 : installation à l'aide du rôle de compte de service IAM

Avant d'utiliser cette méthode, vérifiez que les conditions préalables suivantes sont respectées :

- Vous disposez d'un cluster Amazon EKS fonctionnel avec des nœuds rattachés à l'une des Régions AWS prenant en charge Container Insights. Pour obtenir la liste des régions prises en charge, consultez Container Insights.
- `kubectl` est installé et configuré pour le cluster. Pour plus d'informations, consultez Installation de `kubectl` dans le Guide de l'utilisateur Amazon EKS.
- `eksctl` est installé. Pour plus d'informations, veuillez consulter Installation ou mise à jour de `eksctl` dans le Guide de l'utilisateur Amazon EKS.

Pour installer le module complémentaire Amazon CloudWatch Observability EKS à l'aide du rôle de compte de service IAM

1. Saisissez la commande suivante pour créer un fournisseur OpenID Connect (OIDC), si le cluster n'en possède pas déjà un. Pour plus d'informations, veuillez consulter Configuration d'un compte de service Kubernetes pour assumer un rôle IAM dans le Guide de l'utilisateur Amazon EKS.

   ```
   eksctl utils associate-iam-oidc-provider --cluster my-cluster-name --approve
   ```

2. Entrez la commande suivante pour créer le rôle IAM avec la CloudWatchAgentServerPolicypolitique attachée, et configurez le compte de service de l'agent pour qu'il assume ce rôle à l'aide d'OIDC. Remplacez *my-cluster-name*par le nom de votre cluster, *my-service-account-role*puis par le nom du rôle auquel vous souhaitez associer le compte de service. Si le rôle n'existe pas encore, `eksctl` le crée pour vous.

```
eksctl create iamserviceaccount \
  --name cloudwatch-agent \
  --namespace amazon-cloudwatch --cluster my-cluster-name \
  --role-name my-service-account-role \
  --attach-policy-arn arn:aws:iam::aws:policy/CloudWatchAgentServerPolicy \
  --role-only \
  --approve
```

3. Installez l'add-on en saisissant la commande suivante. Remplacez *my-cluster-name*par le nom de votre cluster, remplacez *111122223333* par votre ID de compte et remplacez par le rôle IAM créé *my-service-account-role*à l'étape précédente.

```
aws eks create-addon --addon-name amazon-cloudwatch-observability --cluster-
name my-cluster-name --service-account-role-arn arn:aws:iam::111122223333:role/my-
service-account-role
```

(Facultatif) Configuration supplémentaire

Ne pas collecter les journaux de conteneurs

Par défaut, le module complémentaire utilise Fluent Bit pour collecter les journaux des conteneurs à partir de tous les pods, puis envoie les CloudWatch journaux à Logs. Pour plus d'informations sur les journaux collectés, veuillez consulter Configuration de Fluent Bit.

Pour refuser la collecte des journaux de conteneurs, passez l'option suivante lorsque vous créez ou mettez à jour le module complémentaire :

```
--configuration-values '{ "containerLogs": { "enabled": false } }'
```

Se désinscrire de la collecte de métriques sur les GPU NVIDIA

À partir de la version 1.300034.0 de l' CloudWatch agent, Container Insights collecte par défaut les métriques du GPU NVIDIA à partir des charges de travail EKS. Ces mesures sont répertoriées dans le tableau deMétriques du GPU NVIDIA.

Vous pouvez choisir de ne pas collecter les métriques du GPU NVIDIA en définissant l'`accelerated_compute_metrics`option dans le fichier de configuration de l' CloudWatch agent sur`false`. Cette option se trouve dans la `kubernetes` section de la `metrics_collected` section du fichier CloudWatch de configuration. Voici un exemple de configuration de désinscription.

```
{
  "agent": {
    "region": "us-east-1"
  },
  "logs": {
    "metrics_collected": {
      "emf": {
      },
      "kubernetes": {
        "enhanced_container_insights": true,
        "accelerated_compute_metrics": false
      }
    },
    "force_flush_interval": 5,
  }
}
```

Utiliser une configuration d' CloudWatch agent personnalisée

Pour collecter d'autres métriques, journaux ou traces à l'aide de l' CloudWatch agent, vous pouvez définir une configuration personnalisée tout en gardant Container Insights et CloudWatch Application Signals activés. Pour ce faire, intégrez le fichier de configuration de l' CloudWatch agent dans la clé de configuration située sous la clé d'agent de la configuration avancée que vous pouvez utiliser lors de la création ou de la mise à jour du module complémentaire EKS. Ce qui suit représente la configuration par défaut de l’agent lorsque vous ne fournissez aucune configuration supplémentaire.

> **Important**
>
> Toute configuration personnalisée que vous fournissez à l’aide de paramètres de configuration supplémentaires remplace la configuration par défaut utilisée par l’agent. Veillez à ne pas désactiver involontairement les fonctionnalités activées par défaut, telles que

Container Insights avec une observabilité améliorée et les signaux d' CloudWatch application. Dans le cas où vous devez fournir une configuration d’agent personnalisée, nous vous recommandons d’utiliser la configuration par défaut suivante comme référence, puis de la modifier en conséquence.

```
--configuration-values '{
  "agent": {
    "config": {
      "logs": {
        "metrics_collected": {
          "app_signals": {},
          "kubernetes": {
            "enhanced_container_insights": true
          }
        }
      },
      "traces": {
        "traces_collected": {
          "app_signals": {}
        }
      }
    }

}'
```

L'exemple suivant montre la configuration par défaut de l' CloudWatch agent sous Windows. L' CloudWatch agent sous Windows ne prend pas en charge la configuration personnalisée.

```
{
  "logs": {
    "metrics_collected": {
      "kubernetes": {
        "enhanced_container_insights": true
      },
    }
  }
}
```

Gérer les certificats TLS du webhook d’admission

Le module complémentaire Amazon CloudWatch Observability EKS utilise les webhooks d'admission Kubernetes pour valider et muter les demandes de ressources `Instrumentation` personnalisées (CR), `AmazonCloudWatchAgent` et éventuellement les demandes de pod Kubernetes sur le cluster si Application Signals est activé. CloudWatch Dans Kubernetes, les webhooks nécessitent que le serveur d'API soit configuré pour faire confiance à un certificat TLS afin de garantir une communication sécurisée.

Par défaut, le module complémentaire Amazon CloudWatch Observability EKS génère automatiquement une autorité de certification auto-signée et un certificat TLS signé par cette autorité de certification pour sécuriser la communication entre le serveur d'API et le serveur Webhook. Ce certificat généré automatiquement a une durée de validité par défaut de 10 ans et n'est pas renouvelé automatiquement à l'expiration. En outre, le bundle CA et le certificat sont régénérés chaque fois que le module complémentaire est mis à niveau ou réinstallé, ce qui réinitialise la durée de validité. Si vous souhaitez modifier la durée de validité par défaut du certificat généré automatiquement, vous pouvez utiliser les configurations supplémentaires suivantes lors de la création ou de la mise à jour du module complémentaire. *`expiry-in-days`*Remplacez-le par la durée de péremption souhaitée en jours.

```
--configuration-values '{ "admissionWebhooks": { "autoGenerateCert":
 { "expiryDays": expiry-in-days } } }'
```

Pour une solution d'autorité de certification plus sécurisée et riche en fonctionnalités, le module complémentaire prend en charge cert-manager, une solution largement adoptée pour la gestion des certificats TLS dans Kubernetes qui simplifie le processus d'obtention, de renouvellement, de gestion et d'utilisation de ces certificats. Il garantit que les certificats sont valides et à jour, et tente de renouveler les certificats à une heure configurée avant leur expiration. cert-manager facilite également l'émission de certificats provenant de diverses sources prises en charge, y compris l'autorité de AWS Certificate Manager Private Certificate Authority.

Nous vous recommandons de consulter les bonnes pratiques en matière de gestion des certificats TLS sur vos clusters et d'opter pour cert-manager pour les environnements de production. Notez que si vous acceptez d'activer cert-manager pour gérer les certificats TLS du webhook d'admission, vous devez préinstaller cert-manager sur votre cluster Amazon EKS avant d'installer le module complémentaire Amazon Observability EKS. CloudWatch Reportez-vous à la documentation de cert-manager pour en savoir plus sur les options d'installation disponibles. Après l'avoir installé, vous pouvez choisir d'utiliser cert-manager pour gérer les certificats TLS du webhook d'admission en utilisant la configuration supplémentaire suivante lors de la création ou de la mise à jour du module complémentaire.

```
--configuration-values '{ "admissionWebhooks": { "certManager": { "enabled":
 true } } }'
```

La configuration avancée décrite dans cette section utilisera par défaut un SelfSignedémetteur.

Collecte des identifiants de volumes Amazon EBS

Si vous souhaitez collecter les ID de volume Amazon EBS dans les journaux de performance, vous devez ajouter une autre politique au rôle IAM attaché aux composants master ou au compte de service. Ajoutez les éléments suivants en tant que politique en ligne. Pour de plus amples informations, veuillez consulter Ajout et suppression d'autorisations basées sur l'identité IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ec2:DescribeVolumes"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

Résolution des problèmes liés au module CloudWatch complémentaire Amazon Observability EKS

Utilisez les informations suivantes pour résoudre les problèmes liés au module complémentaire Amazon CloudWatch Observability EKS.

Mise à jour et suppression du module complémentaire Amazon CloudWatch Observability EKS

Pour obtenir des instructions sur la mise à jour ou la suppression du module complémentaire Amazon CloudWatch Observability EKS, consultez la section Gestion des modules complémentaires Amazon EKS. Utilisez `amazon-cloudwatch-observability` comme nom du module complémentaire.

Vérifiez la version de l' CloudWatch agent utilisée par le module complémentaire Amazon CloudWatch Observability EKS

Le module complémentaire Amazon CloudWatch Observability EKS installe une ressource personnalisée `AmazonCloudWatchAgent` qui contrôle le comportement du daemonset de l' CloudWatchagent sur le cluster, y compris la version de l' CloudWatch agent utilisée. Vous pouvez obtenir la liste de toutes les ressources `AmazonCloudWatchAgent` personnalisées installées sur votre cluster en saisissant la commande suivante :

```
kubectl get amazoncloudwatchagent -A
```

Dans le résultat de cette commande, vous devriez être en mesure de vérifier la version de l' CloudWatch agent. Vous pouvez également décrire la ressource `amazoncloudwatchagent` ou l'un des pods `cloudwatch-agent-*` exécutés sur votre cluster pour inspecter l'image utilisée.

Manipulation d'un ConfigurationConflict lors de la gestion du module complémentaire

Lorsque vous installez ou mettez à jour le module complémentaire Amazon CloudWatch Observability EKS, si vous remarquez une défaillance causée par un type `Health Issue ConfigurationConflict` de fichier dont la description commence par`Conflicts found when trying to apply. Will not continue due to resolve conflicts mode`, c'est probablement parce que vous avez déjà ClusterRoleBinding installé l' CloudWatch agent et ses composants associés tels que le ServiceAccount, le ClusterRole et le sur le cluster. Lorsque le module complémentaire tente d'installer l' CloudWatch agent et ses composants associés, s'il détecte une modification du contenu, il échoue par défaut à l'installation ou à la mise à jour pour éviter de modifier l'état des ressources du cluster.

Si vous essayez d'intégrer le module complémentaire Amazon CloudWatch Observability EKS et que vous constatez cet échec, nous vous recommandons de supprimer une configuration d' CloudWatch agent existante que vous aviez précédemment installée sur le cluster, puis d'installer le module complémentaire EKS. Veillez à sauvegarder toutes les personnalisations que vous avez éventuellement apportées à la configuration d'origine de l' CloudWatch agent, telle qu'une configuration d'agent personnalisée, et à les fournir au module complémentaire Amazon CloudWatch Ob servability EKS lors de sa prochaine installation ou mise à jour. Si vous avez déjà installé l' CloudWatch agent pour l'intégration à Container Insights, consultez Suppression de l' CloudWatch agent et de Fluent Bit for Container Insights pour plus d'informations.

Le module complémentaire prend également en charge une option de configuration de résolution des conflits capable de spécifier `OVERWRITE`. Vous pouvez utiliser cette option pour procéder à

l’installation ou à la mise à jour du module complémentaire en remplaçant les conflits sur le cluster. Si vous utilisez la console Amazon EKS, vous trouverez la Méthode de résolution des conflits lorsque vous choisissez les Paramètres de configuration facultatifs lorsque vous créez ou mettez à jour le module complémentaire. Si vous utilisez le AWS CLI, vous pouvez fournir le `--resolve-conflicts OVERWRITE` à votre commande pour créer ou mettre à jour le module complémentaire.

Métriques collectées par l' CloudWatchagent

Vous pouvez collecter des métriques auprès des serveurs en installant l' CloudWatch agent sur le serveur. Vous pouvez installer l'agent sur les deux instances Amazon EC2 et les serveurs locaux, ainsi que sur les ordinateurs exécutant Linux, Windows Server ou macOS. Si vous installez l'agent sur une instance Amazon EC2, les métriques collectées s'ajoutent aux métriques activées par défaut sur les instances Amazon EC2.

Pour plus d'informations sur l'installation de l' CloudWatch agent sur une instance, consultezCollectez des métriques, des journaux et des traces avec l' CloudWatch agent.

Toutes les métriques abordées dans cette section sont collectées directement par l' CloudWatch agent.

Mesures collectées par l' CloudWatchagent sur les instances Windows Server

Sur un serveur exécutant Windows Server, l'installation de l' CloudWatch agent vous permet de collecter les métriques associées aux compteurs dans Windows Performance Monitor. Les noms des CloudWatch métriques de ces compteurs sont créés en plaçant un espace entre le nom de l'objet et le nom du compteur. Par exemple, le nom de la métrique est attribué au `% Interrupt Time` compteur de l'`Processor`objet `Processor % Interrupt Time` dans CloudWatch. Pour plus d'informations sur les compteurs Windows Performance Monitor, consultez la documentation Microsoft Windows Server.

L'espace de noms par défaut pour les métriques collectées par l' CloudWatch agent est`CWAgent`, bien que vous puissiez spécifier un espace de noms différent lorsque vous configurez l'agent.

Métriques collectées par l' CloudWatchagent sur les instances Linux et macOS

Le tableau suivant répertorie les mesures que vous pouvez collecter avec l' CloudWatch agent sur les serveurs Linux et les ordinateurs macOS.

Métrique	Description
`cpu_time_active`	Durée pendant laquelle l'UC est active dans n'importe quelle capacité. Cette métrique est présentée en centièmes de seconde. Unité : aucune
`cpu_time_guest`	Durée pendant laquelle l'UC exécute une UC virtuelle pour un système d'exploitation invité. Cette métrique est présentée en centièmes de seconde. Unité : aucune
`cpu_time_guest_nice`	Durée pendant laquelle l'UC exécute une UC virtuelle pour un système d'exploitation invité ayant une priorité faible et qui peut être interrompu par d'autres processus. Cette métrique est présentée en centièmes de seconde. Unité : aucune
`cpu_time_idle`	Durée pendant laquelle l'UC est inactive. Cette métrique est présentée en centièmes de seconde. Unité : aucune
`cpu_time_iowait`	Durée pendant laquelle l'UC est en attente d'exécuti on d'opérations d'I/O. Cette métrique est présentée en centièmes de seconde. Unité : aucune
`cpu_time_irq`	Durée pendant laquelle l'UC prend en charge des interruptions. Cette métrique est présentée en centièmes de seconde. Unité : aucune

Métrique	Description
cpu_time_nice	Durée pendant laquelle l'UC est en mode utilisateur avec des processus à faible priorité pouvant facilement être interrompus par des processus à priorité élevée. Cette métrique est présentée en centièmes de seconde. Unité : aucune
cpu_time_softirq	Durée pendant laquelle l'UC prend en charge des interruptions logicielles. Cette métrique est présentée en centièmes de seconde. Unité : aucune
cpu_time_steal	Durée pendant laquelle l'UC subit un vol de temps, c'est-à-dire qu'elle se consacre à d'autres systèmes d'exploitation dans un environnement virtualisé. Cette métrique est présentée en centièmes de seconde. Unité : aucune
cpu_time_system	Durée pendant laquelle l'UC est en mode système. Cette métrique est présentée en centièmes de seconde. Unité : aucune
cpu_time_user	Durée pendant laquelle l'UC est en mode utilisateur. Cette métrique est présentée en centièmes de seconde. Unité : aucune
cpu_usage_active	Durée, en pourcentage, pendant laquelle l'UC est active dans n'importe quelle capacité. Unité : pourcentage

Métrique	Description
cpu_usage_guest	Pourcentage du temps pendant lequel l'UC exécute une UC virtuelle pour un système d'exploitation invité. Unité : pourcentage
cpu_usage_guest_nice	Pourcentage du temps pendant lequel l'UC exécute une UC virtuelle pour un système d'exploitation invité ayant une priorité faible et qui peut être interrompu par d'autres processus. Unité : pourcentage
cpu_usage_idle	Pourcentage de temps durant lequel l'UC est inactive. Unité : pourcentage
cpu_usage_iowait	Pourcentage du temps pendant lequel l'UC est en attente d'exécution d'opérations d'I/O. Unité : pourcentage
cpu_usage_irq	Pourcentage du temps pendant lequel l'UC prend en charge des interruptions. Unité : pourcentage
cpu_usage_nice	Pourcentage du temps pendant lequel l'UC est en mode utilisateur avec des processus à faible priorité pouvant facilement être interrompus par des processus à priorité élevée. Unité : pourcentage

Métrique	Description
cpu_usage_softirq	Pourcentage du temps pendant lequel l'UC prend en charge des interruptions logicielles. Unité : pourcentage
cpu_usage_steal	Pourcentage du temps pendant lequel l'UC subit un vol de temps, c'est-à-dire qu'elle se consacre à d'autres systèmes d'exploitation dans un environnement virtualisé. Unité : pourcentage
cpu_usage_system	Pourcentage du temps pendant lequel l'UC est en mode système. Unité : pourcentage
cpu_usage_user	Pourcentage du temps pendant lequel l'UC est en mode utilisateur. Unité : pourcentage
disk_free	Espace libre sur les disques. Unité : octets
disk_inodes_free	Nombre de nœuds d'index disponibles sur le disque. Unité : nombre
disk_inodes_total	Nombre total de nœuds d'index réservés sur le disque. Unité : nombre
disk_inodes_used	Nombre de nœuds d'index utilisés sur le disque. Unité : nombre

Métrique	Description
`disk_total`	Espace total sur les disques, y compris l'espace utilisé et l'espace libre. Unité : octets
`disk_used`	Espace utilisé sur les disques. Unité : octets
`disk_used_percent`	Pourcentage d'espace disque total utilisé. Unité : pourcentage
`diskio_iops_in_progress`	Nombre de demandes d'I/O émises pour le pilote du périphérique mais qui n'ont pas encore été exécutées . Unité : nombre
`diskio_io_time`	Durée pendant laquelle le disque a eu des demandes d'I/O placées en file d'attente. Unité : millisecondes La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.
`diskio_reads`	Nombre d'opérations de lecture sur disque. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.

Métrique	Description
`diskio_read_bytes`	Nombre d'octets lus sur les disques. Unité : octets La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.
`diskio_read_time`	Durée pendant laquelle les demandes de lecture ont attendu sur les disques. Plusieurs demandes de lecture simultanément en attente peuvent augmenter ce chiffre. Par exemple, si 5 demandes attendent toutes pendant 100 millisecondes en moyenne, 500 est indiqué. Unité : millisecondes La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.
`diskio_writes`	Nombre d'opérations d'écriture sur disque. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.
`diskio_write_bytes`	Nombre d'octets écrits sur les disques. Unité : octets La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.

Métrique	Description
`diskio_write_time`	Durée pendant laquelle les demandes d'écriture ont attendu sur les disques. Plusieurs demandes d'écritur e simultanément en attente peuvent augmenter ce chiffre. Par exemple, si 8 demandes attendent toutes pendant 1 000 millisecondes en moyenne, 8 000 est indiqué. Unité : millisecondes La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.
`ethtool_bw_in_allowance_exceeded`	Nombre de paquets mis en file d'attente et/ou ignorés parce que la bande passante agrégée entrante a dépassé le maximum de l'instance. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune
`ethtool_bw_out_allowance_exceeded`	Nombre de paquets mis en file d'attente ou ignorés parce que la bande passante agrégée sortante a dépassé le maximum de l'instance. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune

Métrique	Description
`ethtool_conntrack_allowance_exceeded`	Nombre de paquets ignorés flottee que le suivi des connexions a dépassé le maximum de l'instance et que de nouvelles connexions n'ont pas pu être établies. Cela peut entraîner une perte de paquets pour le trafic vers ou en provenance de l'instance. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune
`ethtool_linklocal_allowance_exceeded`	Nombre de paquets ignorés abandonné que le PPS du trafic vers les services proxy locaux a dépassé le maximum de l'interface réseau. Cela affecte le trafic vers le service DNS, le service des métadonnées d'instance et le service Amazon Time Sync. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour de plus amples informations, veuillez consulter Récupérez des métriques des performances réseau Unité : aucune

Métrique	Description
`ethtool_pps_allowance_exceeded`	Nombre de paquets mis en file d'attente et/ou ignorés parce que le PPS bidirectionnel a dépassé le maximum de l'instance. Cette métrique n'est collectée que si vous l'avez répertoriée dans la `ethtool` sous-section de la `metrics_collected` section du fichier de configuration de l' CloudWatch agent. Pour plus d'informations, consultez Récupérez des métriques des performances réseau. Unité : aucune
`mem_active`	Quantité de mémoire utilisée d'une manière ou d'une autre pendant la dernière période d'échantillonnage. Unité : octets
`mem_available`	Quantité de mémoire disponible et qui peut être attribuée instantanément aux processus. Unité : octets
`mem_available_percent`	Pourcentage de mémoire disponible et qui peut être attribuée instantanément aux processus. Unité : pourcentage
`mem_buffered`	Quantité de mémoire en cours d'utilisation pour les tampons. Unité : octets
`mem_cached`	Quantité de mémoire en cours d'utilisation pour les caches de fichier. Unité : octets

Métrique	Description
`mem_free`	Quantité de mémoire qui n'est pas en cours d'utilisation. Unité : octets
`mem_inactive`	Quantité de mémoire non utilisée d'une manière ou d'une autre pendant la dernière période d'échantillonnage Unité : octets
`mem_total`	Quantité totale de mémoire. Unité : octets
`mem_used`	Quantité de mémoire actuellement en cours d'utilisation. Unité : octets
`mem_used_percent`	Pourcentage de mémoire actuellement en cours d'utilisation. Unité : pourcentage
`net_bytes_recv`	Nombre d'octets reçus par l'interface réseau. Unité : octets La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.
`net_bytes_sent`	Nombre d'octets envoyés par l'interface réseau. Unité : octets La seule statistique qui doit être utilisée pour cette métrique est `Sum`. N'utilisez pas `Average`.

Métrique	Description
net_drop_in	Nombre de paquets reçus par cette interface réseau qui ont été abandonnés. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est Sum. N'utilisez pas Average.
net_drop_out	Nombre de paquets transmis par cette interface réseau qui ont été abandonnés. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est Sum. N'utilisez pas Average.
net_err_in	Nombre d'erreurs de réception détectées par cette interface réseau. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est Sum. N'utilisez pas Average.
net_err_out	Nombre d'erreurs de transmission détectées par cette interface réseau. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est Sum. N'utilisez pas Average.
net_packets_sent	Nombre de paquets envoyés par cette interface réseau. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est Sum. N'utilisez pas Average.

Métrique	Description
net_packets_recv	Nombre de paquets reçus par cette interface réseau. Unité : nombre La seule statistique qui doit être utilisée pour cette métrique est Sum. N'utilisez pas Average.
netstat_tcp_close	Nombre de connexions TCP sans état. Unité : nombre
netstat_tcp_close_wait	Nombre de connexions TCP en attente d'une demande de mise hors service de la part du client. Unité : nombre
netstat_tcp_closing	Nombre de connexions TCP en attente d'une demande de mise hors service avec accusé de réception du client. Unité : nombre
netstat_tcp_established	Nombre de connexions TCP établies. Unité : nombre
netstat_tcp_fin_wait1	Nombre de connexions TCP dans l'état FIN_WAIT1 pendant le processus de clôture d'une connexion. Unité : nombre
netstat_tcp_fin_wait2	Nombre de connexions TCP dans l'état FIN_WAIT2 pendant le processus de clôture d'une connexion. Unité : nombre

Métrique	Description
`netstat_tcp_last_ack`	Nombre de connexions TCP en attente de l'envoi par le client de son accusé de réception du message de mise hors service de la connexion. Il s'agit du dernier état juste avant la clôture de la connexion. Unité : nombre
`netstat_tcp_listen`	Nombre de ports TCP actuellement à l'écoute d'une demande de connexion. Unité : nombre
`netstat_tcp_none`	Nombre de connexions TCP comportant des clients inactifs. Unité : nombre
`netstat_tcp_syn_sent`	Nombre de connexions TCP en attente d'une demande de connexion correspondante après avoir envoyé une demande de connexion. Unité : nombre
`netstat_tcp_syn_recv`	Nombre de connexions TCP en attente d'un accusé de réception de demande de connexion après avoir envoyé et reçu une demande de connexion. Unité : nombre
`netstat_tcp_time_wait`	Nombre de connexions TCP actuellement en attente de la confirmation de la réception par le client de l'accusé de réception de sa demande de mise hors service de la connexion. Unité : nombre

Métrique	Description
`netstat_udp_socket`	Nombre de connexions UDP actuelles. Unité : nombre
`processes_blocked`	Nombre de processus bloqués. Unité : nombre
`processes_dead`	Nombre de processus morts, indiqué par le code d'état `X` sous Linux. Cette métrique n'est pas collectée sur les ordinateurs macOS. Unité : nombre
`processes_idle`	Nombre de processus inactifs (en veille depuis plus de 20 secondes). Disponible uniquement sur les instances FreeBSD. Unité : nombre
`processes_paging`	Nombre de processus en cours de pagination, indiqué par le code d'état `W` sous Linux. Cette métrique n'est pas collectée sur les ordinateurs macOS. Unité : nombre
`processes_running`	Nombre de processus en cours d'exécution, indiquée par le code d'état `R` sous Linux. Unité : nombre
`processes_sleeping`	Nombre de processus en veille, indiqué par le code d'état `S` sous Linux. Unité : nombre

Métrique	Description
processes_stopped	Nombre de processus arrêtés, indiqué par le code d'état T sous Linux. Unité : nombre
processes_total	Nombre total de processus sur l'instance. Unité : nombre
processes_total_threads	Nombre total de threads composant les processus. Cette métrique n'est disponible que sur les instances Linux. Cette métrique n'est pas collectée sur les ordinateurs macOS. Unité : nombre
processes_wait	Nombre de processus en cours de pagination, indiqué par le code d'état W sur les instances FreeBSD. Cette métrique est disponible uniquement sur les instances FreeBSD et n'est pas disponible sur les instances Linux, Windows Server ou macOS. Unité : nombre
processes_zombies	Nombre de processus zombies, indiqué par le code d'état Z sous Linux. Unité : nombre
swap_free	Quantité d'espace d'échange qui n'est pas en cours d'utilisation. Unité : octets

Métrique	Description
`swap_used`	Quantité d'espace d'échange actuellement en cours d'utilisation. Unité : octets
`swap_used_percent`	Pourcentage d'espace d'échange actuellement en cours d'utilisation. Unité : pourcentage

Définitions des métriques de mémoire collectées par l' CloudWatch agent

Lorsque l' CloudWatch agent collecte des métriques de mémoire, la source est le sous-système de gestion de la mémoire de l'hôte. Par exemple, le noyau Linux expose les données conservées par le système d'exploitation dans `/proc`. Pour ce qui est de la mémoire, les données sont dans `/proc/meminfo`.

Chaque système d'exploitation et chaque architecture utilisent des calculs différents des ressources utilisées par les processus. Pour plus d'informations, consultez les sections suivantes.

Au cours de chaque intervalle de collecte, l' CloudWatch agent de chaque instance collecte les ressources de l'instance et calcule les ressources utilisées par tous les processus exécutés dans cette instance. Ces informations sont reproduites dans les CloudWatch métriques. Vous pouvez configurer la durée de l'intervalle de collecte dans le fichier de configuration de l' CloudWatch agent. Pour plus d’informations, consultez CloudWatch fichier de configuration de l'agent : section Agent.

La liste suivante explique comment sont définies les métriques de mémoire collectées par l' CloudWatch agent.

- Mémoire active : mémoire utilisée par un processus. En d'autres termes, la mémoire utilisée par des applications en cours d'exécution.
- Mémoire disponible : mémoire qui peut être instantanément attribuée aux processus sans que le système soit remplacé (également appelée mémoire virtuelle).
- Mémoire tampon : zone de données partagée par des périphériques matériels ou des processus de programme qui fonctionnent à des vitesses et à des priorités différentes.

- Mémoire mise en cache : stocke les instructions et les données du programme qui sont utilisées à plusieurs reprises dans le cadre des programmes dont le processeur est susceptible d'avoir besoin par la suite.
- Mémoire libre : mémoire qui n'est pas du tout utilisée et qui est facilement disponible. L'utilisation du système est totalement gratuite en cas de besoin.
- Mémoire inactive : pages qui n'ont pas été consultées « récemment ».
- Mémoire totale : taille de la mémoire vive physique.
- Mémoire utilisée : mémoire actuellement utilisée par les programmes et les processus.

Rubriques

Linux : métriques collectées et calculs utilisés

Métriques collectées et unités :

- Active (octets)
- Disponible (octets)
- Pourcentage disponible (pourcentage)
- Mise en tampon (octets)
- Mise en cache (octets)
- Gratuite (octets)
- Inactive (octets)
- Totale (octets)
- Utilisée (octets)
- Pourcentage utilisé (pourcentage)

Mémoire utilisée = Mémoire totale - Mémoire libre - Mémoire en cache - Mémoire tampon

Mémoire totale = Mémoire utilisée + Mémoire libre + Mémoire en cache + Mémoire tampon

macOS : métriques collectées et calculs utilisés

Métriques collectées et unités :

- Active (octets)
- Disponible (octets)
- Pourcentage disponible (pourcentage)
- Gratuite (octets)
- Inactive (octets)
- Totale (octets)
- Utilisée (octets)
- Pourcentage utilisé (pourcentage)

Mémoire disponible = Mémoire libre + Mémoire inactive

Mémoire utilisée = Mémoire totale - Mémoire disponible

Mémoire totale = Mémoire disponible + Mémoire utilisée

Windows : métriques collectées

Les métriques collectées sur les hôtes Windows sont répertoriées ci-dessous. Toutes ces métriques ont `None` pour `Unit`.

- Disponible (octets)
- Défauts de cache/sec
- Défauts de page/sec
- Pages/sec

Aucun calcul n'est utilisé pour les métriques Windows car l' CloudWatch agent analyse les événements à partir des compteurs de performance.

Exemple : calcul des métriques de mémoire sous Linux

Par exemple, supposons que la saisie de la commande **`cat /proc/meminfo`** sur un hôte Linux donne les résultats suivants :

```
MemTotal:       3824388 kB
MemFree:         462704 kB
MemAvailable:   2157328 kB
Buffers:         126268 kB
Cached:         1560520 kB
SReclaimable:    289080 kB>
```

Dans cet exemple, l' CloudWatch agent collectera les valeurs suivantes. Toutes les valeurs que l' CloudWatch agent collecte et rapporte sont exprimées en octets.

- `mem_total` : 3 916 173 312 octets
- `mem_available`: 2209103872 octets (+ en cache) MemFree
- `mem_free` : 473 808 896 octets
- `mem_cached` : 1 893 990 400 octets (`cached` + `SReclaimable`)
- `mem_used` : 1 419 075 584 octets (`MemTotal` – (`MemFree` + `Buffers` + (`Cached` + `SReclaimable`)))
- `mem_buffered` : 129 667 072 octets
- `mem_available_percent` : 56,41 %
- `mem_used_percent` : 36,24 % (`mem_used` / `mem_total`) * 100

Scénarios courants avec l' CloudWatchagent

Les sections suivantes expliquent comment effectuer les tâches courantes de configuration et de personnalisation de l' CloudWatch agent.

Rubriques

- Exécution de l' CloudWatch agent en tant qu'utilisateur différent
- Comment l' CloudWatch agent gère les fichiers journaux épars
- Ajouter des dimensions personnalisées aux métriques collectées par l' CloudWatch agent
- Plusieurs fichiers CloudWatch de configuration d'agents
- Agrégation ou agrégation des métriques collectées par l'agent CloudWatch
- Collecte de métriques à haute résolution avec l'agent CloudWatch
- Envoi de métriques, de journaux et de traces à un autre compte
- Différences d'horodatage entre l' CloudWatch agent unifié et l'ancien CloudWatch agent Logs

Exécution de l' CloudWatch agent en tant qu'utilisateur différent

Sur les serveurs Linux, il CloudWatch s'exécute en tant qu'utilisateur root par défaut. Pour que l'agent soit exécuté sous un autre nom d'utilisateur, utilisez le `run_as_user` paramètre figurant dans la `agent` section du fichier de configuration de l' CloudWatch agent. Cette option est disponible uniquement sur les serveurs Linux.

Si vous exécutez déjà l'agent avec l'utilisateur racine et que vous souhaitez le modifier pour utiliser l'identité d'un autre utilisateur, suivez l'une des procédures ci-après.

Pour exécuter l' CloudWatch agent en tant qu'utilisateur différent sur une instance EC2 exécutant Linux

1. Téléchargez et installez un nouveau package CloudWatch d'agent. Pour plus d’informations, consultez Téléchargez le package de CloudWatch l'agent.
2. Créez un nouvel utilisateur Linux ou utilisez l'utilisateur par défaut nommé `cwagent` créé par le fichier DEB ou RPM.
3. Utilisez l'une des méthodes suivantes pour fournir les informations d'identification de cet utilisateur :
   - Si le fichier `.aws/credentials` existe dans le répertoire personnel de l'utilisateur root, vous devez créer un fichier d'informations d'identification pour l'utilisateur que vous allez utiliser pour exécuter l' CloudWatch agent. Ce fichier d'informations d'identification sera `/home/`*`username`*`/.aws/credentials`. Ensuite, affectez le chemin du fichier d'informations d'identification en tant que valeur du paramètre `shared_credential_file` dans `common-config.toml`. Pour de plus amples informations, consultez (Facultatif) Modifiez la Configuration commune pour les informations de proxy ou de région.
   - Si le fichier `.aws/credentials` n'existe pas dans le répertoire personnel de l'utilisateur racine, vous pouvez procéder de l'une des manières suivantes :
     - Créez un fichier d'informations d'identification pour l'utilisateur que vous allez utiliser pour exécuter l' CloudWatch agent. Ce fichier d'informations d'identification sera `/home/`*`username`*`/.aws/credentials`. Ensuite, affectez le chemin du fichier d'informations d'identification en tant que valeur du paramètre `shared_credential_file` dans `common-config.toml`. Pour de plus amples informations, consultez (Facultatif) Modifiez la Configuration commune pour les informations de proxy ou de région.

- Au lieu de créer un fichier d'informations d'identification, associez un rôle IAM à l'instance. L'agent utilise ce rôle en tant que fournisseur d'informations d'identification.

4. Dans le fichier de configuration de l' CloudWatch agent, ajoutez la ligne suivante dans la `agent` section :

```
"run_as_user": "username"
```

   Vous pouvez apporter d'autres modifications au fichier de configuration en fonction des besoins. Pour de plus amples informations, veuillez consulter Création du fichier de configuration de CloudWatch l'agent

5. Donnez à l'utilisateur les autorisations requises. L'utilisateur doit disposer des autorisations Read (r) pour que les fichiers journaux soient collectés et de l'autorisation Execute (x) sur chaque répertoire du chemin des fichiers journaux.
6. Démarrez l'agent avec le fichier de configuration que vous venez de modifier.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Pour exécuter l' CloudWatch agent en tant qu'utilisateur différent sur un serveur local exécutant Linux

1. Téléchargez et installez un nouveau package CloudWatch d'agent. Pour plus d'informations, consultez Téléchargez le package de CloudWatch l'agent.
2. Créez un nouvel utilisateur Linux ou utilisez l'utilisateur par défaut nommé `cwagent` créé par le fichier DEB ou RPM.
3. Stockez les informations d'identification de cet utilisateur dans un dossier auquel l'utilisateur peut accéder, tel que `/home/username/.aws/credentials`.
4. Affectez le chemin du fichier d'informations d'identification en tant que valeur du paramètre `shared_credential_file` dans `common-config.toml`. Pour plus d'informations, consultez (Facultatif) Modifiez la Configuration commune pour les informations de proxy ou de région.
5. Dans le fichier de configuration de l' CloudWatch agent, ajoutez la ligne suivante dans la `agent` section :

```
"run_as_user": "username"
```

Vous pouvez apporter d'autres modifications au fichier de configuration en fonction des besoins. Pour de plus amples informations, veuillez consulter Création du fichier de configuration de CloudWatch l'agent

6. Donnez à l'utilisateur les autorisations requises. L'utilisateur doit disposer des autorisations Read (r) pour que les fichiers journaux soient collectés et de l'autorisation Execute (x) sur chaque répertoire du chemin des fichiers journaux.
7. Démarrez l'agent avec le fichier de configuration que vous venez de modifier.

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-
config -m ec2 -s -c file:configuration-file-path
```

Comment l' CloudWatch agent gère les fichiers journaux épars

Les fichiers fragmentés sont des fichiers avec des blocs vides et du contenu réel. Un fichier épars utilise plus efficacement l'espace disque en écrivant de brèves informations représentant les blocs vides sur le disque au lieu des octets nuls réels qui composent le bloc. Généralement, la taille réelle d'un fichier fragmenté apparaît alors beaucoup plus petite que sa taille apparente.

Toutefois, l' CloudWatch agent ne traite pas les fichiers fragmentés différemment des fichiers normaux. Lorsque l'agent lit un fichier fragmenté, les blocs vides sont traités comme des blocs « réels » remplis d'octets nuls. De ce fait, l' CloudWatch agent publie autant d'octets que la taille apparente d'un fichier fragmenté sur CloudWatch.

La configuration de l' CloudWatch agent pour publier un fichier fragmenté peut entraîner des CloudWatch coûts plus élevés que prévu. Nous vous recommandons donc de ne pas le faire. Par exemple, `/var/logs/lastlog` sous Linux, il s'agit généralement d'un fichier très fragmenté, et nous vous recommandons de ne pas le publier CloudWatch dans.

Ajouter des dimensions personnalisées aux métriques collectées par l' CloudWatch agent

Pour ajouter des dimensions personnalisées telles que des étiquettes à des métriques collectées par l'agent, ajoutez le champ `append_dimensions` dans la section du fichier de configuration d'agent qui répertorie ces métriques.

Par exemple, l'exemple suivant de section du fichier de configuration ajoute une dimension personnalisée nommée `stackName` avec la valeur `Prod` pour les métriques `cpu` et `disk` collectées par l'agent.

```
"cpu":{
  "resources":[
    "*"
  ],
  "measurement":[
    "cpu_usage_guest",
    "cpu_usage_nice",
    "cpu_usage_idle"
  ],
  "totalcpu":false,
  "append_dimensions":{
    "stackName":"Prod"
  }
},
"disk":{
  "resources":[
    "/",
    "/tmp"
  ],
  "measurement":[
    "total",
    "used"
  ],
  "append_dimensions":{
    "stackName":"Prod"
  }
}
```

Souvenez-vous que chaque fois que vous modifiez le fichier de configuration d'agent, vous devez ensuite redémarrer l'agent pour que les modifications prennent effet.

Plusieurs fichiers CloudWatch de configuration d'agents

Sur les serveurs Linux comme sur les serveurs Windows, vous pouvez configurer l' CloudWatch agent pour qu'il utilise plusieurs fichiers de configuration. Par exemple, vous pouvez utiliser un fichier de configuration courant qui recueille un ensemble de métriques, de journaux et de traces que vous souhaitez toujours collecter à partir de tous les serveurs dans votre infrastructure. Vous pouvez

alors utiliser les fichiers de configuration supplémentaires qui collectent des métriques de certaines applications ou dans certaines situations.

Pour cette configuration, commencez par créer les fichiers de configuration que vous souhaitez utiliser. Tous les fichiers de configuration qui seront utilisés ensemble sur le même serveur doivent avoir différents noms de fichiers. Vous pouvez stocker les fichiers de configuration sur les serveurs ou dans le Parameter Store.

Démarrez l' CloudWatch agent à l'aide de l'`fetch-config`option et spécifiez le premier fichier de configuration. Pour ajouter le second fichier de configuration pour l'agent en cours d'exécution, utilisez la même commande, mais avec l'option `append-config`. L'ensemble des métriques, des journaux et des traces répertoriés dans l'un ou l'autre fichier de configuration sont collectés. Les exemples de commandes suivants illustrent ce scénario en utilisant des stockages de configurations sous forme de fichiers. La première ligne démarre l'agent à l'aide du fichier de configuration `infrastructure.json` et la seconde ligne ajoute le fichier de configuration `app.json`.

Les exemples de commandes suivants concernent Linux.

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2
 -s -c file:/tmp/infrastructure.json
```

```
/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a append-config -m
 ec2 -s -c file:/tmp/app.json
```

Les exemples de commandes suivants concernent Windows Server.

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a fetch-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\infrastructure.json"
```

```
& "C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1"
 -a append-config -m ec2 -s -c file:"C:\Program Files\Amazon\AmazonCloudWatchAgent
\app.json"
```

Les exemples de fichiers de configuration suivants illustrent une utilisation pour cette fonction. Le premier fichier de configuration est utilisé pour tous les serveurs de l'infrastructure. Le second collecte des journaux uniquement à partir d'une application et est ajouté aux serveurs exécutant cette application.

infrastructure.json

```
{
  "metrics": {
    "metrics_collected": {
      "cpu": {
        "resources": [
          "*"
        ],
        "measurement": [
          "usage_active"
        ],
        "totalcpu": true
      },
      "mem": {
         "measurement": [
           "used_percent"
        ]
      }
    }
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-
agent.log",
            "log_group_name": "amazon-cloudwatch-agent.log"
          },
          {
            "file_path": "/var/log/messages",
            "log_group_name": "/var/log/messages"
          }
        ]
      }
    }
  }
}
```

app.json

```
{
```

```
    "logs": {
        "logs_collected": {
            "files": {
                "collect_list": [
                    {
                        "file_path": "/app/app.log*",
                        "log_group_name": "/app/app.log"
                    }
                ]
            }
        }
    }
}
```

Tous les fichiers de configuration ajoutés à la configuration doivent avoir des noms de fichiers différents les uns des autres et du fichier de configuration initiale. Si vous utilisez `append-config` avec un fichier de configuration ayant le même nom de fichier qu'un fichier de configuration que l'agent utilise déjà, la commande Ajouter écrasera les informations provenant du premier fichier de configuration, au lieu d'effectuer un ajout. C'est le cas même si les deux fichiers de configuration avec le même nom de fichier sont sur différents chemins de fichier.

L'exemple précédent montre l'utilisation des deux fichiers de configuration, mais il n'y a pas de limite au nombre de fichiers de configuration que vous pouvez ajouter à la configuration de l'agent. Vous pouvez également combiner l'utilisation des fichiers de configuration situés sur les serveurs et les configurations situées dans le Parameter Store.

Agrégation ou agrégation des métriques collectées par l'agent CloudWatch

Pour regrouper ou propager les métriques recueillies par l'agent, ajoutez un champ `aggregation_dimensions` à la section correspondante dans le fichier de configuration d'agent.

Par exemple, le fichier de configuration suivant extrait les métriques sur la dimension `AutoScalingGroupName`. Les métriques de toutes les instances de chaque groupe Auto Scaling sont regroupées et peuvent être affichées comme un tout.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"]]
}
```

Pour déployer la combinaison de chacune des dimensions `InstanceId` et `InstanceType` en plus du déploiement sur le nom de groupe Auto Scaling, ajoutez les éléments suivants.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [["AutoScalingGroupName"], ["InstanceId", "InstanceType"]]
}
```

Pour propager des métriques dans une collection, utilisez `[]`.

```
"metrics": {
  "cpu":{...}
  "disk":{...}
  "aggregation_dimensions" : [[]]
}
```

Souvenez-vous que chaque fois que vous modifiez le fichier de configuration d'agent, vous devez ensuite redémarrer l'agent pour que les modifications prennent effet.

Collecte de métriques à haute résolution avec l'agent CloudWatch

Le champ `metrics_collection_interval` indique l'intervalle de temps pour les métriques collectées, en quelques secondes. En spécifiant une valeur inférieure à 60 pour ce champ, les métriques sont collectées en haute résolution.

Par exemple, si toutes vos métriques doivent être en haute résolution et collectées toutes les 10 secondes, spécifiez 10 comme valeur pour `metrics_collection_interval` sous la section `agent` comme intervalle de collecte des métriques globales.

```
"agent": {
  "metrics_collection_interval": 10
}
```

Sinon, l'exemple suivant définit les métriques `cpu` à collecter chaque seconde, et toutes les autres métriques sont collectées toutes les minutes.

```
"agent":{
  "metrics_collection_interval": 60
},
"metrics":{
```

```
  "metrics_collected":{
    "cpu":{
      "resources":[
        "*"
      ],
      "measurement":[
        "cpu_usage_guest"
      ],
      "totalcpu":false,
      "metrics_collection_interval": 1
    },
    "disk":{
      "resources":[
        "/",
        "/tmp"
      ],
      "measurement":[
        "total",
        "used"
      ]
    }
  }
}
```

Souvenez-vous que chaque fois que vous modifiez le fichier de configuration d'agent, vous devez ensuite redémarrer l'agent pour que les modifications prennent effet.

Envoi de métriques, de journaux et de traces à un autre compte

Pour que l' CloudWatch agent envoie les métriques, les journaux ou les traces à un autre compte, spécifiez un `role_arn` paramètre dans le fichier de configuration de l'agent sur le serveur d'envoi. La valeur `role_arn` spécifie un rôle IAM dans le compte cible que l'agent utilise pour envoyer des données vers le compte cible. Ce rôle permet au compte d'envoi d'assumer un rôle correspondant dans le compte de destination lorsque vous envoyez les métriques ou les journaux au compte de destination.

Vous pouvez également spécifier des chaînes `role_arn` distinctes dans le fichier de configuration de l'agent : une pour l'envoi de métriques, une autre pour l'envoi de journaux et une autre pour l'envoi de traces.

L'exemple suivant de l'extrait de la section `agent` du fichier de configuration paramètre l'agent afin qu'il utilise `CrossAccountAgentRole` pour envoyer des données à un compte différent.

```
{
  "agent": {
    "credentials": {
      "role_arn": "arn:aws:iam::123456789012:role/CrossAccountAgentRole"
    }
  },
  .....
}
```

Sinon, l'exemple suivant définit différents rôles que le compte d'envoi devra utiliser pour envoyer des métriques, des journaux et des traces :

```
"metrics": {
    "credentials": {
     "role_arn": "RoleToSendMetrics"
    },
    "metrics_collected": {....
```

```
"logs": {
    "credentials": {
    "role_arn": "RoleToSendLogs"
    },
    ....
```

Politiques requises

Lorsque vous spécifiez un `role_arn` dans le fichier de configuration de l'agent, vous devez également vous assurer que les rôles IAM des comptes d'envoi et de destination disposent de certaines stratégies. Les rôles des comptes d'envoi et de destination doivent avoir la politique `CloudWatchAgentServerPolicy`. Pour plus d'informations sur l'assignation de cette politique à un rôle, consultez Créez des rôles IAM à utiliser avec l' CloudWatch agent sur les instances Amazon EC2.

Le rôle dans le compte d'envoi doit également inclure la politique suivante. Vous ajoutez cette stratégie dans l'onglet Permissions (Autorisations) de la console IAM lorsque vous modifiez le rôle.

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sts:AssumeRole"
            ],
            "Resource": [
                "arn:aws:iam::target-account-ID:role/agent-role-in-target-account"
            ]
        }
    ]
}
```

Le rôle dans le compte de destination doit inclure la stratégie suivante, afin qu'il reconnaisse le rôle IAM utilisé par le compte d'envoi. Vous ajoutez cette stratégie dans l'onglet Trust relationships (Relations d'approbation) de la console IAM lorsque vous modifiez le rôle. Le rôle dans le compte de destination dans lequel vous ajoutez cette politique est le rôle que vous avez créé dans Création de rôles et d'utilisateurs IAM à utiliser avec l'agent CloudWatch . Ce rôle est celui spécifié dans *agent-role-in-target-account* dans la politique utilisée par le compte d'envoi.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::sending-account-ID:role/role-in-sender-account"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Différences d'horodatage entre l' CloudWatch agent unifié et l'ancien CloudWatch agent Logs

L' CloudWatch agent prend en charge un ensemble de symboles différent pour les formats d'horodatage par rapport à l'ancien agent CloudWatch Logs. Ces différences figurent dans le tableau suivant.

Symboles pris en charge par les deux agents	Symboles pris en charge uniquement par CloudWatch l'agent unifié	Symboles pris en charge uniquement par l'ancien agent CloudWatch Logs
%A, %a, %b, %B, %d, %f, %H, %I, %m, %M, %p, %S, %y, %Y, %Z, %z	%-d, %-I, %-m, %-M, %-S	%c, %j, %U, %W, %w

Pour plus d'informations sur la signification des symboles pris en charge par le nouvel CloudWatch agent, consultez la section Fichier de configuration de l' CloudWatch agent : journaux du guide de CloudWatch l'utilisateur Amazon. Pour plus d'informations sur les symboles pris en charge par l'agent CloudWatch Logs, consultez le fichier de configuration de l'agent dans le guide de l'utilisateur Amazon CloudWatch Logs.

Résolution des problèmes liés à l' CloudWatch agent

Utilisez les informations suivantes pour résoudre les problèmes liés à l' CloudWatch agent.

Rubriques

- Arrêt et redémarrage de l'agent CloudWatch

CloudWatch paramètres de ligne de commande de l'agent

Pour voir la liste complète des paramètres pris en charge par l' CloudWatch agent, entrez ce qui suit sur la ligne de commande de l'ordinateur sur lequel l'agent est installé :

```
amazon-cloudwatch-agent-ctl -help
```

L'installation de l' CloudWatch agent à l'aide de la commande Exécuter échoue

Pour installer l' CloudWatch agent à l'aide de la commande Run de Systems Manager, l'agent SSM sur le serveur cible doit être de version 2.2.93.0 ou ultérieure. Si la version de votre agent SSM Agent n'est pas la bonne, des erreurs peuvent survenir et indiquer les messages suivants :

```
no latest version found for package AmazonCloudWatchAgent on platform linux
```

```
failed to download installation package reliably
```

Pour plus d'informations sur la mise à jour de la version de votre agent SSM, consultez Installation et configuration de l'agent SSM Agent dans le Guide de l'utilisateur AWS Systems Manager .

L' CloudWatch agent ne veut pas démarrer

Si l' CloudWatch agent ne démarre pas, il se peut qu'il y ait un problème dans votre configuration. Les informations de configuration sont journalisées dans le fichier `configuration-validation.log`. Ce fichier se trouve dans `/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` sur les serveurs Linux et dans `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log` sur les serveurs exécutant Windows Server.

Vérifiez que l' CloudWatch agent est en cours d'exécution

Vous pouvez interroger l' CloudWatch agent pour savoir s'il est en cours d'exécution ou s'il est arrêté. Pour le faire à distance, vous pouvez utiliser AWS Systems Manager . Vous pouvez également utiliser la ligne de commande, mais uniquement pour vérifier le serveur local.

Pour demander l'état de l' CloudWatch agent à l'aide de la commande Exécuter

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le panneau de navigation, choisissez Run Command (Fonctionnalité Exécuter la commande).

   -ou-

   Si la page d' AWS Systems Manager accueil s'ouvre, faites défiler la page vers le bas et choisissez Explore Run Command.
3. Sélectionnez Run Command (Exécuter la commande).
4. Dans la liste du document de commande, cliquez sur le bouton situé à côté de AmazonCloudWatch- ManageAgent.
5. Dans la liste Action, choisissez status (état).
6. Pour Source de configuration facultative choisissez par défaut et laissez vide Emplacement de configuration facultatif.
7. Dans la zone Cible, choisissez l'instance à vérifier.
8. Cliquez sur Exécuter.

Si l'agent est en cours d'exécution, la sortie ressemble à ce qui suit.

```
{
       "status": "running",
       "starttime": "2017-12-12T18:41:18",
       "version": "1.73.4"
}
```

Si l'agent est arrêté, le champ `"status"` indique `"stopped"`.

Pour demander l'état de l' CloudWatch agent localement à l'aide de la ligne de commande

- Sur un serveur Linux, saisissez ce qui suit :

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   status
  ```

Sur un serveur exécutant Windows Server, entrez les informations suivantes en PowerShell tant qu'administrateur :

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
 ec2 -a status
```

L' CloudWatch agent ne démarre pas et l'erreur mentionne une région Amazon EC2

Si l'agent ne démarre pas et que le message d'erreur mentionne un point de terminaison dans la région Amazon EC2, il est possible que vous ayez configuré l'agent pour qu'il ait besoin de l'accès au point de terminaison Amazon EC2, mais que vous n'ayez pas accordé cet accès.

Par exemple, si vous spécifiez une valeur pour le paramètre `append_dimensions` du fichier de configuration de l'agent qui dépend des métadonnées Amazon EC2 et que vous utilisez des proxys, vous devez vous assurer que le serveur peut accéder au point de terminaison pour Amazon EC2. Pour plus d'informations sur ces points de terminaison, consultez la rubrique Amazon Elastic Compute Cloud (Amazon EC2) du document Référence générale d'Amazon Web Services.

L' CloudWatch agent ne démarre pas sous Windows Server

Sur Windows Server, l'erreur suivante peut s'afficher :

```
Start-Service : Service 'Amazon CloudWatch Agent (AmazonCloudWatchAgent)' cannot be
 started due to the following
error: Cannot start service AmazonCloudWatchAgent on computer '.'.
At C:\Program Files\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1:113
 char:12
+     $svc | Start-Service
+            ~~~~~~~~~~~~~
    + CategoryInfo          : OpenError:
 (System.ServiceProcess.ServiceController:ServiceController) [Start-Service],
   ServiceCommandException
    + FullyQualifiedErrorId :
 CouldNotStartService,Microsoft.PowerShell.Commands.StartServiceCommand
```

Avant de résoudre ce problème, vérifiez que le service serveur est en cours d'exécution. Cette erreur peut être observée si l'agent essaie de démarrer lorsque le service serveur n'est pas en cours d'exécution.

Si le service serveur est déjà en cours d'exécution, le problème suivant peut être le suivant. Sur certaines installations de Windows Server, le démarrage de l' CloudWatch agent prend plus de 30 secondes. Comme Windows Server n'accorde par défaut que 30 secondes aux services pour démarrer, cela provoque l'échec de l'agent avec une erreur similaire à la suivante :

Pour résoudre ce problème, augmentez le délai d'attente du service. Pour de plus amples informations, consultez Un service ne démarre pas et les événements 7000 et 7011 sont enregistrés dans le journal des événements Windows.

Où sont les métriques ?

Si l' CloudWatch agent est en cours d'exécution mais que vous ne trouvez pas les métriques qu'il a collectées dans le AWS Management Console ou le AWS CLI, vérifiez que vous utilisez le bon espace de noms. L'espace de noms pour les métriques collectées par l'agent est `CWAgent` par défaut. Vous pouvez personnaliser cet espace de noms à l'aide du champ `namespace` dans la section `metrics` du fichier de configuration d'agent. Si vous ne voyez pas les métriques attendues, consultez le fichier de configuration pour confirmer l'espace de noms utilisé.

Lorsque vous téléchargez le package de l' CloudWatch agent pour la première fois, le fichier de configuration de l'agent est`amazon-cloudwatch-agent.json`. Ce fichier est situé dans le répertoire où vous avez exécuté l'assistant de configuration. Il se peut que vous l'ayez déplacé dans un autre répertoire. Si vous utilisez l'assistant de configuration, le fichier de configuration d'agent produit à partir de l'assistant est nommé `config.json`. Pour plus d'informations sur le fichier de configuration, y compris le champ `namespace`, consultez CloudWatch fichier de configuration de l'agent : section Metrics.

L' CloudWatch agent met du temps à s'exécuter dans un conteneur ou enregistre une erreur de limite de sauts

Lorsque vous exécutez l' CloudWatch agent en tant que service de conteneur et que vous souhaitez ajouter des dimensions métriques Amazon EC2 à toutes les métriques collectées par l'agent, les erreurs suivantes peuvent s'afficher dans la version v1.247354.0 de l'agent :

```
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Unable to retrieve Instance
 Metadata Tags. This plugin must only be used on an EC2 instance.
```

```
2022-06-07T03:36:11Z E! [processors.ec2tagger] ec2tagger: Please increase hop limit
 to 2 by following this document https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/
configuring-instance-metadata-options.html#configuring-IMDS-existing-instances.
2022-06-07T03:36:11Z E! [telegraf] Error running agent: could not initialize processor
 ec2tagger: EC2MetadataRequestError: failed to get EC2 instance identity document
caused by: EC2MetadataError: failed to make EC2Metadata request
        status code: 401, request id:
caused by:
```

Cette erreur peut s'afficher si l'agent essaie d'obtenir des métadonnées d'IMDSv2 à l'intérieur d'un conteneur sans limite de saut appropriée. Dans les versions de l'agent antérieures à la v1.247354.0, vous pouvez rencontrer ce problème sans voir le message du journal.

Pour résoudre ce problème, augmentez la limite de saut à 2 en suivant les instructions de Configurer les options de métadonnées d'instance.

J'ai mis à jour la configuration de mon agent mais je ne vois pas les nouvelles métriques ou les nouveaux journaux dans la CloudWatch console

Si vous mettez à jour le fichier de configuration de votre CloudWatch agent, vous devrez utiliser l'**`fetch-config`**option au prochain démarrage de l'agent. Par exemple, si vous avez stocké le fichier mis à jour sur l'ordinateur local, entrez la commande suivante :

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -
s -m ec2 -c file:configuration-file-path
```

CloudWatch fichiers et emplacements des agents

Le tableau suivant répertorie les fichiers installés et utilisés par l' CloudWatch agent, ainsi que leur emplacement sur les serveurs exécutant Linux ou Windows Server.

Fichier	Emplacement Linux	Emplacement Windows Server
Script de contrôle qui contrôle le démarrage, l'arrêt et le redémarrage de l'agent.	`/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl` ou `/usr/bin/amazon-cloudwatch-agent-ctl`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1`

Fichier	Emplacement Linux	Emplacement Windows Server
Fichier journal dans lequel l'agent écrit. Vous devrez peut-être joindre ce document lors de la prise de contact AWS Support.	`/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` ou `/var/log/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log`
Fichier de validation de la configuration de l'agent.	`/opt/aws/amazon-cloudwatch-agent/logs/configuration-validation.log` ou `/var/log/amazon/amazon-cloudwatch-agent/configuration-validation.log`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\configuration-validation.log`
Fichier JSON utilisé pour configurer l'agent immédiatement après sa création par l'Assistant. Pour de plus amples informations, consultez Création du fichier de configuration de CloudWatch l'agent.	`/opt/aws/amazon-cloudwatch-agent/bin/config.json`	`$Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\config.json`
Fichier JSON utilisé pour configurer l'agent si ce fichier de configuration a été téléchargé à partir du Parameter Store.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json` ou `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.json`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.json`

Fichier	Emplacement Linux	Emplacement Windows Server
Le fichier TOML utilisé pour spécifier la région et les informations d'identification devant être utilisées par l'agent, à la place des valeurs par défaut du système.	`/opt/aws/amazon-cloudwatch-agent/etc/common-config.toml` ou `/etc/amazon/amazon-cloudwatch-agent/common-config.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\common-config.toml`
Le fichier TOML contenant le contenu converti du fichier de configuration JSON. Le script `amazon-cloudwatch-agent-ctl` génère ce fichier. Les utilisateurs ne doivent pas modifier directement ce fichier. Cela peut être utile pour vérifier que la traduction JSON vers TOML a été réussie.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.toml` ou `/etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.toml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.toml`
Le fichier YAML contenant le contenu converti du fichier de configuration JSON. Le script `amazon-cloudwatch-agent-ctl` génère ce fichier. Vous ne devez pas modifier directement ce fichier. Ce fichier peut être utile pour vérifier que la traduction JSON vers YAML a été réussie.	`/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.yaml or /etc/amazon/amazon-cloudwatch-agent/amazon-cloudwatch-agent.yaml`	`$Env:ProgramData\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent.yaml`

Recherche d'informations sur les versions des CloudWatch agents

Pour trouver le numéro de version de l' CloudWatch agent sur un serveur Linux, entrez la commande suivante :

```
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a status
```

Pour trouver le numéro de version de l' CloudWatch agent sur Windows Server, entrez la commande suivante :

```
& $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m ec2
 -a status
```

> **Note**
>
> Cette commande est la bonne méthode pour trouver la version de l' CloudWatch agent. Si vous utilisez Programs and Features (Programmes et fonctionnalités) dans le Panneau de configuration, vous verrez un numéro de version incorrect.

Vous pouvez également télécharger un fichier README sur les dernières modifications apportées à l'agent, ainsi qu'un fichier qui indique le numéro de version actuellement disponible pour le téléchargement. Ces fichiers se trouvent aux emplacements suivants :

- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/RELEASE_NOTES` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/RELEASE_NOTES`
- `https://amazoncloudwatch-agent.s3.amazonaws.com/info/latest/CWAGENT_VERSION` ou `https://amazoncloudwatch-agent-`*`region`*`.s3.`*`region`*`.amazonaws.com/info/latest/CWAGENT_VERSION`

Logs générés par l' CloudWatchagent

L'agent génère un journal pendant son exécution. Ce journal comprend des informations de dépannage. Ce journal est le fichier `amazon-cloudwatch-agent.log`. Ce fichier se trouve dans `/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log` sur les serveurs Linux et dans `$Env:ProgramData\Amazon\AmazonCloudWatchAgent\Logs\amazon-cloudwatch-agent.log` sur les serveurs exécutant Windows Server.

Vous pouvez configurer l'agent afin qu'il consigne des détails supplémentaires dans le fichier `amazon-cloudwatch-agent.log`. Dans le fichier de configuration de l'agent, dans la `agent` section, définissez le `debug` champ sur`true`, puis reconfigurez et redémarrez l' CloudWatch agent.

Pour désactiver la journalisation de ces informations supplémentaires, définissez le champ `debug` sur `false`. Ensuite, reconfigurez et redémarrez l'agent. Pour plus d'informations, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

Dans les versions 1.247350.0 et ultérieures de l' CloudWatch agent, vous pouvez éventuellement définir le `aws_sdk_log_level` champ dans la `agent` section du fichier de configuration de l'agent sur une ou plusieurs des options suivantes. Séparez les différentes options avec le caractère `|`.

- `LogDebug`
- `LogDebugWithSigning`
- `LogDebugWithHTTPBody`
- `LogDebugRequestRetries`
- `LogDebugWithEventStreamBody`

Pour plus d'informations sur ces options, consultez LogLevelType.

Arrêt et redémarrage de l'agent CloudWatch

Vous pouvez arrêter l' CloudWatch agent manuellement à l'aide de l'un ou de l'autre AWS Systems Manager ou de la ligne de commande.

Pour arrêter l' CloudWatch agent à l'aide de la commande Exécuter

1. Ouvrez la console Systems Manager à l'adresse https://console.aws.amazon.com/systems-manager/.
2. Dans le panneau de navigation, choisissez Run Command (Fonctionnalité Exécuter la commande).

   -ou-

   Si la page d' AWS Systems Manager accueil s'ouvre, faites défiler la page vers le bas et choisissez Explore Run Command.
3. Sélectionnez Run Command (Exécuter la commande).
4. Dans la liste du document de commande, choisissez AmazonCloudWatch- ManageAgent.
5. Dans la zone Cibles, choisissez l'instance sur laquelle vous avez installé l' CloudWatch agent.
6. Dans la liste Action, choisissez stop.

7. Laissez les champs Optional Configuration Source (Source de configuration facultative) et Optional Configuration Location (Emplacement de configuration facultative) vides.
8. Cliquez sur Exécuter.

Pour arrêter l' CloudWatch agent localement à l'aide de la ligne de commande

- Sur un serveur Linux, saisissez ce qui suit :

  ```
  sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a
   stop
  ```

  Sur un serveur exécutant Windows Server, entrez les informations suivantes en PowerShell tant qu'administrateur :

  ```
  & $Env:ProgramFiles\Amazon\AmazonCloudWatchAgent\amazon-cloudwatch-agent-ctl.ps1 -m
   ec2 -a stop
  ```

Pour redémarrer l'agent, suivez les instructions de la section Démarrez l' CloudWatch agent.

Intégration de métriques dans les journaux

Le format de métrique CloudWatch intégré vous permet de générer des métriques personnalisées de manière asynchrone sous la forme de journaux écrits dans Logs. CloudWatch Vous pouvez intégrer des métriques personnalisées à des données détaillées sur les événements du journal, et extraire CloudWatch automatiquement les métriques personnalisées afin de pouvoir les visualiser et les utiliser pour déclencher des alarmes, afin de détecter les incidents en temps réel. En outre, les événements de journal détaillés associés aux métriques extraites peuvent être interrogés à l'aide de CloudWatch Logs Insights pour fournir des informations approfondies sur les causes profondes des événements opérationnels.

Le format de métrique intégrée vous aide à générer des métriques personnalisées exploitables à partir de ressources éphémères telles que des fonctions et des conteneurs Lambda. En utilisant le format de métrique intégrée pour envoyer des journaux à partir de ces ressources éphémères, vous pouvez désormais facilement créer des métriques personnalisées sans avoir à instrumenter ni à maintenir un code distinct, tout en acquérant de puissantes capacités d'analyse sur vos données de journal.

Aucune configuration n'est requise pour utiliser le format de métrique intégrée. Structurez vos journaux en suivant la spécification du format métrique intégré ou générez-les à l'aide de nos bibliothèques clientes et envoyez-les à CloudWatch Logs à l'aide de l'PutLogEvents API ou de l'CloudWatch agent.

Des frais sont encourus pour l'ingestion et l'archivage des journaux, ainsi que pour les métriques personnalisées générées. Pour en savoir plus, consultez Tarification Amazon CloudWatch.

> **Note**
>
> Soyez prudent lorsque vous configurez votre extraction de métriques car cela affecte votre utilisation de métriques personnalisées et la facture correspondante. Si vous créez involontairement des métriques basées sur des dimensions à cardinalité élevée (telles que `requestId`), le format de métrique intégrée créera par définition une métrique personnalisée correspondant à chaque combinaison de dimensions unique. Pour de plus amples informations, veuillez consulter Dimensions.

Rubriques

- Publication de journaux à l'aide du format de métrique intégrée
- Affichage de vos statistiques et journaux dans la console
- Configuration d'alertes sur les métriques créées avec le format de métrique intégrée

Publication de journaux à l'aide du format de métrique intégrée

Vous pouvez générer des journaux de format de métrique intégrée à l'aide des méthodes suivantes :

- Générez et envoyez les journaux à l'aide des bibliothèques clientes open source.
- Générez manuellement les journaux à l'aide de la spécification de format métrique intégrée, puis utilisez l'CloudWatch agent ou l'PutLogEvents API pour envoyer les journaux.

Rubriques

- Création de journaux au format de métrique intégrée à l'aide des bibliothèques clientes
- Spécifications : format de métrique intégrée
- Utilisation de l' PutLogEventsAPI pour envoyer des journaux au format métrique intégré créés manuellement
- Utilisation de l' CloudWatch agent pour envoyer des journaux au format métrique intégrés
- Utilisation du format métrique intégré avec AWS Distro pour OpenTelemetry

Création de journaux au format de métrique intégrée à l'aide des bibliothèques clientes

Amazon fournit des bibliothèques clientes open source que vous pouvez utiliser pour créer des journaux de format de métrique intégrée. Actuellement, ces bibliothèques sont disponibles pour les langues figurant dans la liste suivante. Des exemples complets de différentes configurations se trouvent dans nos bibliothèques clientes sous /examples.

Les bibliothèques et les instructions pour leur utilisation se trouvent sur Github. Utilisez les liens suivants.

- Node.Js

> **Note**
>
> Pour Node.js, les versions 4.1.1+, 3.0.2+, 2.0.7+ sont requises pour une utilisation avec le format de journal Lambda JSON. L'utilisation de versions précédentes dans de tels environnements Lambda entraînera une perte de métriques.
> Pour plus d'informations, consultez la section Accès aux CloudWatch journaux Amazon pour AWS Lambda.

- Python
- Java
- C#

Les bibliothèques clientes sont conçues pour fonctionner immédiatement avec l' CloudWatch agent. Les journaux au format métrique intégré générés sont envoyés à l' CloudWatch agent, qui les agrège ensuite et les publie pour vous dans CloudWatch Logs.

> **Note**
>
> Lors de l'utilisation de Lambda, aucun agent n'est requis pour envoyer les journaux à. CloudWatch Tout ce qui est enregistré sur STDOUT est envoyé à CloudWatch Logs via l'agent de journalisation Lambda.

Spécifications : format de métrique intégrée

Le format de métrique CloudWatch intégré est une spécification JSON utilisée pour demander à CloudWatch Logs d'extraire automatiquement les valeurs métriques intégrées dans les événements de journaux structurés. Vous pouvez l'utiliser CloudWatch pour représenter graphiquement et créer des alarmes sur les valeurs métriques extraites.

Conventions de spécification de format de métrique intégrée

Les mots-clés « DOIT », « NE DOIT PAS », « OBLIGATOIRE », « DEVRAIT », « NE DEVRAIT PAS », « RECOMMANDÉ », « PEUT » et « FACULTATIF » dans cette spécification de format doivent être interprétés tels que décrits dans Mots-clés RFC2119.

Les termes « JSON », « texte JSON », « valeur JSON », « membre », « élément », « objet », « tableau », « nombre », « chaîne », « booléen », « vrai », « faux » et « nul » dans cette spécification de format doivent être interprétés tels que définis dans la norme JavaScript Object Notation RFC8259.

> **Note**
>
> Si vous prévoyez de créer des alertes sur des métriques créées à l'aide du format de métrique intégrée, consultez les recommandations de la rubrique Configuration d'alertes sur les métriques créées avec le format de métrique intégrée.

Structure du document au format de métrique intégrée

Cette section décrit la structure d'un document au format de métrique intégrée. Les documents au format métrique intégré sont définis dans la norme JavaScript Object Notation RFC8259.

Sauf indication contraire, les objets définis par la présente spécification NE DOIVENT PAS contenir de membres supplémentaires. Les membres non reconnus par cette spécification DOIVENT être ignorés. Les membres définis dans cette spécification sont sensibles à la casse.

Le format métrique intégré est soumis aux mêmes limites que les événements CloudWatch Logs standard et est limité à une taille maximale de 256 Ko.

Avec le format de métrique intégré, vous pouvez suivre le traitement de vos journaux EMF par des métriques qui sont publiées dans l'espace de nom `AWS/Logs` de votre compte. Celles-ci peuvent être utilisées pour suivre les échecs de génération de métriques à partir d'EMF, ainsi que pour savoir si les échecs sont dus à l'analyse ou à la validation. Pour plus de détails, consultez la section Surveillance à l'aide de CloudWatch métriques.

Nœud racine

Le LogEvent message DOIT être un objet JSON valide sans données supplémentaires au début ou à la fin de la chaîne de LogEvent message. Pour plus d'informations sur la LogEvent structure, consultez InputLogEvent.

Les documents au format de métrique intégrée DOIVENT contenir le membre de niveau supérieur suivant sur le nœud racine. Il s'agit d'un objet Objet Métadonnées.

```
{
 "_aws": {
```

```
    "CloudWatchMetrics": [ ... ]
  }
}
```

Le nœud racine DOIT contenir tous les membres Membres cibles définis par les références dans le MetricDirective objet.

Le nœud racine peut contenir tous les autres membres qui ne sont pas inclus dans les exigences ci-dessus. Les valeurs de ces membres DOIVENT être des types JSON valides.

Objet Métadonnées

Le `_aws` membre peut être utilisé pour représenter les métadonnées relatives à la charge utile qui indiquent aux services en aval comment ils doivent traiter le LogEvent. La valeur DOIT être un objet et DOIT contenir les membres suivants :

- CloudWatchMetrics— Un tableau de données MetricDirective objet utilisé pour indiquer CloudWatch d'extraire les métriques du nœud racine du LogEvent.

```
{
  "_aws": {
    "CloudWatchMetrics": [ ... ]
  }
}
```

- Timestamp (Horodatage) — Nombre représentant l'horodatage utilisé pour les métriques extraites de l'événement. Les valeurs DOIVENT être exprimées en tant que nombre de millisecondes après le 1er janv. 1970 00:00:00 UTC.

```
{
  "_aws": {
    "Timestamp": 1559748430481
  }
}
```

MetricDirective objet

L' MetricDirective objet indique aux services en aval qu'ils LogEvent contiennent des métriques qui seront extraites et publiées. CloudWatch MetricDirectives DOIT contenir les membres suivants :

- Namespace — Chaîne représentant l'espace de CloudWatch noms de la métrique.

- Dimensions — DimensionSet réseau.
- Metrics (Métriques) — Tableau d'objets MetricDefinition. Ce tableau NE DOIT PAS contenir plus de 100 MetricDefinition objets.

DimensionSet réseau

A DimensionSet est un tableau de chaînes contenant les clés de dimension qui seront appliquées à toutes les métriques du document. Les valeurs de ce tableau DOIVENT également être des membres sur le nœud racine (appelé) Membres cibles

A NE DimensionSet DOIT PAS contenir plus de 30 clés de dimension. A DimensionSet PEUT être vide.

Le membre cible DOIT avoir une valeur de chaîne. Cette valeur NE DOIT PAS contenir plus de 1024 caractères. Le membre cible définit une dimension qui sera publiée dans le cadre de l'identité de métrique. Chaque DimensionSet utilisateur crée une nouvelle métrique dans CloudWatch. Pour obtenir de plus amples informations sur les dimensions, veuillez consulter Dimension et Dimensions.

```
{
 "_aws": {
   "CloudWatchMetrics": [
     {
       "Dimensions": [ [ "functionVersion" ] ],
       ...
     }
   ]
 },
 "functionVersion": "$LATEST"
}
```

> **Note**
>
> Soyez prudent lorsque vous configurez votre extraction de métriques car cela affecte votre utilisation de métriques personnalisées et la facture correspondante. Si vous créez involontairement des métriques basées sur des dimensions à cardinalité élevée (telles que `requestId`), le format de métrique intégrée créera par définition une métrique personnalisée correspondant à chaque combinaison de dimensions unique. Pour de plus amples informations, veuillez consulter Dimensions.

MetricDefinition objet

A MetricDefinition est un objet qui DOIT contenir le membre suivant :

- Name (Nom) —Chaîne Valeurs de référence vers une métrique Membres cibles. Les cibles des métriques DOIVENT être une valeur numérique ou un tableau de valeurs numériques.

Un MetricDefinition objet PEUT contenir les membres suivants :

- Unit (Unitéà — Valeur de chaîne FACULTATIVE représentant l'unité de mesure pour la métrique correspondante. Les valeurs DEVRAIENT être des unités CloudWatch métriques valides. Pour plus d'informations sur les unités valides, consultez MetricDatum. Si aucune valeur n'est fournie, la valeur par défaut NONE est supposée.
- StorageResolution— Une valeur entière FACULTATIVE représentant la résolution de stockage pour la métrique correspondante. Si cette valeur est définie sur 1, cette métrique est une métrique haute résolution, de sorte que la métrique est CloudWatch stockée avec une résolution inférieure à la minute inférieure à une seconde. Si vous définissez cette valeur sur 60, cette métrique est une résolution standard, qui est CloudWatch stockée à une résolution d'une minute. Les valeurs DEVRAIENT être des résolutions valides CloudWatch prises en charge, 1 ou 60. Si aucune valeur n'est fournie, la valeur par défaut 60 est choisie.

  Pour plus d'informations sur les métriques haute résolution, consultez Métriques haute résolution.

> **Note**
>
> Si vous prévoyez de créer des alertes sur des métriques créées à l'aide du format de métrique intégrée, consultez les recommandations de la rubrique Configuration d'alertes sur les métriques créées avec le format de métrique intégrée.

```
{
  "_aws": {
    "CloudWatchMetrics": [
      {
        "Metrics": [
          {
            "Name": "Time",
            "Unit": "Milliseconds",
```

```
            "StorageResolution": 60
          }
        ],
        ...
      }
    ]
  },
  "Time": 1
}
```

Valeurs de référence

Les valeurs de référence sont des valeurs de chaîne qui font référence aux membres Membres cibles sur le nœud racine. Ces références ne doivent PAS être confondues avec les pointeurs JSON décrits dans RFC6901. Les valeurs cibles ne peuvent pas être imbriquées.

Membres cibles

Les cibles valides DOIVENT être des membres sur le nœud racine et ne peuvent pas être des objets imbriqués. Par exemple, une valeur _reference_ de `"A.a"` DOIT correspondre au membre suivant :

```
{ "A.a" }
```

Elle NE DOIT PAS correspondre au membre imbriqué :

```
{ "A": { "a" } }
```

Les valeurs valides des membres cibles dépendent de ce qui les référence. Une cible de métrique DOIT être une valeur numérique ou un tableau de valeurs numériques. Les cibles de métriques de tableau numérique NE DOIVENT PAS avoir plus de 100 membres. Une cible de dimension DOIT avoir une valeur de chaîne.

Exemple de format de métrique intégrée et schéma JSON

Voici un exemple valide de format de métrique intégré.

```
{
  "_aws": {
    "Timestamp": 1574109732004,
    "CloudWatchMetrics": [
      {
```

```
        "Namespace": "lambda-function-metrics",
        "Dimensions": [["functionVersion"]],
        "Metrics": [
          {
            "Name": "time",
            "Unit": "Milliseconds",
            "StorageResolution": 60
          }
        ]
      }
    ]
  },
  "functionVersion": "$LATEST",
  "time": 100,
  "requestId": "989ffbf8-9ace-4817-a57c-e4dd734019ee"
}
```

Vous pouvez utiliser le schéma suivant pour valider les documents au format de métrique intégré.

```
{
    "type": "object",
    "title": "Root Node",
    "required": [
        "_aws"
    ],
    "properties": {
        "_aws": {
            "$id": "#/properties/_aws",
            "type": "object",
            "title": "Metadata",
            "required": [
                "Timestamp",
                "CloudWatchMetrics"
            ],
            "properties": {
                "Timestamp": {
                    "$id": "#/properties/_aws/properties/Timestamp",
                    "type": "integer",
                    "title": "The Timestamp Schema",
                    "examples": [
                        1565375354953
                    ]
                },
```

```
                "CloudWatchMetrics": {
                    "$id": "#/properties/_aws/properties/CloudWatchMetrics",
                    "type": "array",
                    "title": "MetricDirectives",
                    "items": {
                        "$id": "#/properties/_aws/properties/CloudWatchMetrics/items",
                        "type": "object",
                        "title": "MetricDirective",
                        "required": [
                            "Namespace",
                            "Dimensions",
                            "Metrics"
                        ],
                        "properties": {
                            "Namespace": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Namespace",
                                "type": "string",
                                "title": "CloudWatch Metrics Namespace",
                                "examples": [
                                    "MyApp"
                                ],
                                "pattern": "^(.*)$",
                                "minLength": 1,
                                "maxLength": 1024
                            },
                            "Dimensions": {
                                "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Dimensions",
                                "type": "array",
                                "title": "The Dimensions Schema",
                                "minItems": 1,
                                "items": {
                                    "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items",
                                    "type": "array",
                                    "title": "DimensionSet",
                                    "minItems": 0,
                                    "maxItems": 30,
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Dimensions/items/items",
                                        "type": "string",
                                        "title": "DimensionReference",
```

```
                                        "examples": [
                                            "Operation"
                                        ],
                                        "pattern": "^(.*)$",
                                        "minLength": 1,
                                        "maxLength": 250
}
                                    }
                                },
                                "Metrics": {
                                    "$id": "#/properties/_aws/properties/CloudWatchMetrics/
items/properties/Metrics",
                                    "type": "array",
                                    "title": "MetricDefinitions",
                                    "items": {
                                        "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items",
                                        "type": "object",
                                        "title": "MetricDefinition",
                                        "required": [
                                            "Name"
                                        ],
                                        "properties": {
                                            "Name": {
                                                "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Name",
                                                "type": "string",
                                                "title": "MetricName",
                                                "examples": [
                                                    "ProcessingLatency"
                                                ],
                                                "pattern": "^(.*)$",
                                                "minLength": 1,
                                                "maxLength": 1024
                                            },
                                            "Unit": {
                                                "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/Unit",
                                                "type": "string",
                                                "title": "MetricUnit",
                                                "examples": [
                                                    "Milliseconds"
                                                ],
```

```
                                                "pattern": "^(Seconds|Microseconds|
Milliseconds|Bytes|Kilobytes|Megabytes|Gigabytes|Terabytes|Bits|Kilobits|Megabits|
Gigabits|Terabits|Percent|Count|Bytes\\/Second|Kilobytes\\/Second|Megabytes\\/Second|
Gigabytes\\/Second|Terabytes\\/Second|Bits\\/Second|Kilobits\\/Second|Megabits\\/
Second|Gigabits\\/Second|Terabits\\/Second|Count\\/Second|None)$"
                                            },
                                            "StorageResolution": {
                                                "$id": "#/properties/_aws/properties/
CloudWatchMetrics/items/properties/Metrics/items/properties/StorageResolution",
                                                "type": "integer",
                                                "title": "StorageResolution",
                                                "examples": [
                                                    60
                                                ]
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Utilisation de l' PutLogEventsAPI pour envoyer des journaux au format métrique intégré créés manuellement

Vous pouvez envoyer des journaux au format métrique intégré à CloudWatch Logs à l'aide de l' PutLogEvents API CloudWatch Logs. Lors de l'appel PutLogEvents, vous pouvez éventuellement inclure l'en-tête HTTP suivant pour indiquer à CloudWatch Logs que les métriques doivent être extraites, mais cela n'est plus nécessaire.

```
x-amzn-logs-format: json/emf
```

Voici un exemple complet d'utilisation du AWS SDK pour Java 2.x :

```
package org.example.basicapp;

import software.amazon.awssdk.regions.Region;
```

```
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.InputLogEvent;
import software.amazon.awssdk.services.cloudwatchlogs.model.PutLogEventsRequest;

import java.util.Collections;

public class EmbeddedMetricsExample {
        public static void main(String[] args) {

                final String usage = "To run this example, supply a Region code (eg.
 us-east-1), log group, and stream name as command line arguments"
                                + "Ex: PutLogEvents <region-id> <log-group-name>
 <stream-name>";

                if (args.length != 3) {
                        System.out.println(usage);
                        System.exit(1);
                }

                String regionId = args[0];
                String logGroupName = args[1];
                String logStreamName = args[2];

                CloudWatchLogsClient logsClient =
 CloudWatchLogsClient.builder().region(Region.of(regionId)).build();

                // Build a JSON log using the EmbeddedMetricFormat.
                long timestamp = System.currentTimeMillis();
                String message = "{" +
                                "  \"_aws\": {" +
                                "    \"Timestamp\": " + timestamp  + "," +
                                "    \"CloudWatchMetrics\": [" +
                                "      {" +
                                "        \"Namespace\": \"MyApp\"," +
                                "        \"Dimensions\": [[\"Operation\"], [\"Operation
\", \"Cell\"]]," +
                                "        \"Metrics\": [{ \"Name\": \"ProcessingLatency
\", \"Unit\": \"Milliseconds\", \"StorageResolution\": 60 }]" +
                                "      }" +
                                "    ]" +
                                "  }," +
                                "  \"Operation\": \"Aggregator\"," +
```

```
                                "  \"Cell\": \"001\"," +
                                "  \"ProcessingLatency\": 100" +
                                "}";
                InputLogEvent inputLogEvent = InputLogEvent.builder()
                        .message(message)
                        .timestamp(timestamp)
                        .build();

                // Specify the request parameters.
                PutLogEventsRequest putLogEventsRequest = PutLogEventsRequest.builder()
                        .logEvents(Collections.singletonList(inputLogEvent))
                        .logGroupName(logGroupName)
                        .logStreamName(logStreamName)
                        .build();

                logsClient.putLogEvents(putLogEventsRequest);

                System.out.println("Successfully put CloudWatch log event");
        }

}
```

> **Note**
>
> Avec le format de métrique intégré, vous pouvez suivre le traitement de vos journaux EMF par des métriques qui sont publiées dans l'espace de nom `AWS/Logs` de votre compte. Celles-ci peuvent être utilisées pour suivre les échecs de génération de métriques à partir d'EMF, ainsi que pour savoir si les échecs sont dus à l'analyse ou à la validation. Pour plus de détails, consultez la section Surveillance à l'aide de CloudWatch métriques.

Utilisation de l' CloudWatch agent pour envoyer des journaux au format métrique intégrés

Pour utiliser cette méthode, installez d'abord l' CloudWatch agent pour les services à partir desquels vous souhaitez envoyer des journaux au format métrique intégré, puis vous pouvez commencer à envoyer les événements.

L' CloudWatch agent doit être de version 1.230621.0 ou ultérieure.

> **Note**
>
> Il n'est pas nécessaire d'installer l' CloudWatch agent pour envoyer des journaux à partir des fonctions Lambda.
> Les délais d'expiration de la fonction Lambda ne sont pas gérés automatiquement. Cela signifie que si votre fonction expire avant que les mesures ne soient éliminées, les métriques de cette invocation ne seront pas capturées.

Installation de l' CloudWatchagent

Installez l' CloudWatch agent pour chaque service qui doit envoyer des journaux au format métrique intégré.

Installation de l' CloudWatchagent sur EC2

Installez d'abord l' CloudWatch agent sur l'instance. Pour plus d'informations, consultez Installation de l' CloudWatch agent.

Une fois l'agent installé, configurez-le pour qu'il écoute sur un port UDP ou TCP pour les journaux de format de métrique intégrée. Voici un exemple de cette configuration qui écoute sur le socket par défaut `tcp:25888`. Pour en savoir plus sur la configuration de l'agent, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Installation de l' CloudWatchagent sur Amazon ECS

Le moyen le plus simple de déployer l' CloudWatch agent sur Amazon ECS est de l'exécuter en tant que sidecar, en le définissant dans la même définition de tâche que votre application.

Créer un fichier de configuration de l'agent

Créez le fichier de configuration de votre CloudWatch agent localement. Dans cet exemple, le chemin de fichier relatif sera `amazon-cloudwatch-agent.json`.

Pour en savoir plus sur la configuration de l'agent, consultez Création ou modification manuelle du fichier de configuration de CloudWatch l'agent.

```
{
  "logs": {
    "metrics_collected": {
      "emf": { }
    }
  }
}
```

Configuration push vers le Parameter Store SSM

Entrez la commande suivante pour transférer le fichier de configuration de l' CloudWatch agent vers le magasin de paramètres AWS Systems Manager (SSM).

```
aws ssm put-parameter \
    --name "cwagentconfig" \
    --type "String" \
    --value "`cat amazon-cloudwatch-agent.json`" \
    --region "{{region}}"
```

Configuration de la définition de tâche

Configurez votre définition de tâche pour utiliser l' CloudWatch agent et exposer le port TCP ou UDP. L'exemple de définition de tâche que vous devez utiliser dépend de votre mode de mise en réseau.

Notez que le `webapp` spécifie la variable d'environnement `AWS_EMF_AGENT_ENDPOINT`. Ceci est utilisé par la bibliothèque et doit pointer vers le point de terminaison sur lequel l'agent écoute. En outre, le `cwagent` spécifie le `CW_CONFIG_CONTENT` en tant que paramètre « valueFrom » qui pointe vers la configuration SSM que vous avez créée à l'étape précédente.

Cette section contient un exemple pour le mode pont et un exemple pour le mode hôte ou awsvpc. Pour d'autres exemples de configuration de l' CloudWatch agent sur Amazon ECS, consultez le référentiel d'exemples Github

Voici un exemple pour le mode pont. Lorsque la mise en réseau en mode pont est activée, l'agent doit être lié à votre application à l'aide du paramètre `links` et doit être adressé à l'aide du nom du conteneur.

```
{
```

```
  "containerDefinitions": [
          {
              "name": "webapp",
              "links": [ "cwagent" ],
              "image": "my-org/web-app:latest",
              "memory": 256,
              "cpu": 256,
              "environment": [{
                "name": "AWS_EMF_AGENT_ENDPOINT",
                "value": "tcp://cwagent:25888"
              }],
          },
          {
              "name": "cwagent",
              "mountPoints": [],
              "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
              "memory": 256,
              "cpu": 256,
              "portMappings": [{
                  "protocol": "tcp",
                  "containerPort": 25888
              }],
              "environment": [{
                "name": "CW_CONFIG_CONTENT",
                "valueFrom": "cwagentconfig"
              }],
          }
      ],
}
```

Voici un exemple pour le mode hôte ou awsvpc. Lors de l'exécution sur ces modes réseau, l'agent peut être adressé via `localhost`.

```
{
  "containerDefinitions": [
          {
              "name": "webapp",
              "image": "my-org/web-app:latest",
              "memory": 256,
              "cpu": 256,
              "environment": [{
                "name": "AWS_EMF_AGENT_ENDPOINT",
                "value": "tcp://127.0.0.1:25888"
```

```
            }],
        },
        {
            "name": "cwagent",
            "mountPoints": [],
            "image": "public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest",
            "memory": 256,
            "cpu": 256,
            "portMappings": [{
                "protocol": "tcp",
                "containerPort": 25888
            }],
            "environment": [{
              "name": "CW_CONFIG_CONTENT",
              "valueFrom": "cwagentconfig"
            }],
        }
    ],
}
```

Note

En mode awsvpc, vous devez soit donner une adresse IP publique au VPC (Fargate uniquement), soit configurer une passerelle NAT, soit configurer un point de terminaison VPC Logs. CloudWatch Pour de plus amples informations sur la configuration d'un NAT, veuillez consulter Passerelles NAT. Pour plus d'informations sur la configuration d'un point de terminaison VPC CloudWatch Logs, consultez la section Utilisation des CloudWatch journaux avec les points de terminaison VPC d'interface.
Voici un exemple d'attribution d'une adresse IP publique à une tâche qui utilise le type de lancement Fargate.

```
aws ecs run-task \
--cluster {{cluster-name}} \
--task-definition cwagent-fargate \
--region {{region}} \
--launch-type FARGATE \
--network-configuration
 "awsvpcConfiguration={subnets=[{{subnetId}}],securityGroups=[{{sgId}}],assignPublicIp=ENA
```

Assurer les autorisations

Assurez-vous que le rôle IAM exécutant vos tâches a l'autorisation de lire à partir du magasin de paramètres SSM. Vous pouvez ajouter cette autorisation en joignant la politique d'AmazonSSM. ReadOnlyAccess Pour ce faire, entrez la commande suivante.

```
aws iam attach-role-policy --policy-arn arn:aws:iam::aws:policy/AmazonSSMReadOnlyAccess
 \
--role-name CWAgentECSExecutionRole
```

Installation de l' CloudWatchagent sur Amazon EKS

Certaines parties de ce processus peuvent être ignorées si vous avez déjà installé CloudWatch Container Insights sur ce cluster.

Autorisations

Si vous n'avez pas déjà installé Container Insights, assurez-vous d'abord que vos nœuds Amazon EKS disposent des autorisations IAM appropriées. Ils devraient avoir la CloudWatchAgentServerPolicypièce jointe. Pour plus d'informations, consultez Vérifiez les conditions préalables.

Créez ConfigMap

Créez un ConfigMap pour l'agent. Indique ConfigMap également à l'agent d'écouter sur un port TCP ou UDP. Utilisez ce qui suit ConfigMap.

```
# cwagent-emf-configmap.yaml
apiVersion: v1
data:
  # Any changes here must not break the JSON format
  cwagentconfig.json: |
    {
      "agent": {
        "omit_hostname": true
      },
      "logs": {
        "metrics_collected": {
          "emf": { }
        }
      }
    }
```

```
kind: ConfigMap
metadata:
  name: cwagentemfconfig
  namespace: default
```

Si vous avez déjà installé Container Insights, ajoutez la `"emf": { }` ligne suivante à votre version existante ConfigMap.

Appliquez le ConfigMap

Entrez la commande suivante pour appliquer le ConfigMap.

```
kubectl apply -f cwagent-emf-configmap.yaml
```

Deploy the agent (Déploiement de l'agent)

Pour déployer l' CloudWatch agent sous forme de sidecar, ajoutez-le à la définition de votre pod, comme dans l'exemple suivant.

```
apiVersion: v1
kind: Pod
metadata:
  name: myapp
  namespace: default
spec:
  containers:
    # Your container definitions go here
    - name: web-app
      image: my-org/web-app:latest
    # CloudWatch Agent configuration
    - name: cloudwatch-agent
      image: public.ecr.aws/cloudwatch-agent/cloudwatch-agent:latest
      imagePullPolicy: Always
      resources:
        limits:
          cpu: 200m
          memory: 100Mi
        requests:
          cpu: 200m
          memory: 100Mi
      volumeMounts:
        - name: cwagentconfig
```

```
          mountPath: /etc/cwagentconfig
      ports:
  # this should match the port configured in the ConfigMap
        - protocol: TCP
          hostPort: 25888
          containerPort: 25888
  volumes:
    - name: cwagentconfig
      configMap:
        name: cwagentemfconfig
```

Utilisation de l' CloudWatch agent pour envoyer des journaux au format métrique intégrés

Lorsque l' CloudWatch agent est installé et en cours d'exécution, vous pouvez envoyer les journaux au format métrique intégré via TCP ou UDP. Il y a deux exigences lors de l'envoi des journaux par l'intermédiaire de l'agent :

- Les journaux doivent contenir une clé `LogGroupName` indiquant à l'agent quel groupe de journaux utiliser.
- Chaque événement de journal doit se trouver sur une seule ligne. En d'autres termes, un événement de journal ne peut pas contenir le caractère de nouvelle ligne (\n).

Les événements de journaux doivent également suivre la spécification de format de métrique intégrée. Pour plus d'informations, consultez Spécifications : format de métrique intégrée .

Si vous prévoyez de créer des alertes sur des métriques créées à l'aide du format de métrique intégrée, consultez les recommandations de la rubrique Configuration d'alertes sur les métriques créées avec le format de métrique intégrée.

Voici un exemple d'envoi manuel d'événements du journal à partir d'un shell bash Linux. Vous pouvez plutôt utiliser les interfaces de socket UDP fournies par votre langage de programmation de choix.

```
echo '{"_aws":{"Timestamp":1574109732004,"LogGroupName":"Foo","CloudWatchMetrics":
[{"Namespace":"MyApp","Dimensions":[["Operation"]],"Metrics":
[{"Name":"ProcessingLatency","Unit":"Milliseconds","StorageResolution":60}]}]},"Operation":"Agg
 \
> /dev/udp/0.0.0.0/25888
```

> **Note**
>
> Avec le format de métrique intégré, vous pouvez suivre le traitement de vos journaux EMF par des métriques qui sont publiées dans l'espace de nom `AWS/Logs` de votre compte. Celles-ci peuvent être utilisées pour suivre les échecs de génération de métriques à partir d'EMF, ainsi que pour savoir si les échecs sont dus à l'analyse ou à la validation. Pour plus de détails, consultez la section Surveillance à l'aide de CloudWatch métriques.

Utilisation du format métrique intégré avec AWS Distro pour OpenTelemetry

Vous pouvez utiliser le format métrique intégré dans le cadre du OpenTelemetry projet. OpenTelemetry est une initiative open source qui supprime les frontières et les restrictions entre les formats spécifiques aux fournisseurs pour le traçage, les journaux et les métriques en proposant un ensemble unique de spécifications et d'API. Pour plus d'informations, consultez OpenTelemetry.

L'utilisation du format métrique intégré OpenTelemetry nécessite deux composants : une source de données OpenTelemetry conforme et AWS Distro for OpenTelemetry Collector activé pour une utilisation avec des journaux au format métrique CloudWatch intégrés.

Nous avons préconfiguré les redistributions des OpenTelemetry composants, maintenues par AWS, afin de faciliter au maximum l'intégration. Pour plus d'informations sur l'utilisation OpenTelemetry du format métrique intégré, en plus d'autres AWS services, consultez AWS Distro for OpenTelemetry.

Pour plus d'informations sur la prise en charge et l'utilisation du langage, consultez ObservabilitéAWS sur Github.

Affichage de vos statistiques et journaux dans la console

Après avoir généré des journaux au format métrique intégré qui extraient les métriques, vous pouvez utiliser la CloudWatch console pour afficher les métriques. Les métriques intégrées ont les dimensions que vous avez spécifiées lors de la génération des journaux. En outre, les métriques intégrées que vous avez générées à l'aide des bibliothèques clientes ont les dimensions par défaut suivantes :

- ServiceType
- ServiceName

- LogGroup

Pour afficher les métriques générées à partir des journaux au format de métrique intégrée

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Sélectionnez un espace de noms que vous avez spécifié pour vos métriques intégrées lorsque vous les avez générées. Si vous avez utilisé les bibliothèques clientes pour générer les métriques et que vous n'avez pas spécifié d'espace de noms, sélectionnez aws-embedded-metrics. Il s'agit de l'espace de noms par défaut pour les métriques intégrées générées à l'aide des bibliothèques clientes.
4. Sélectionnez une dimension métrique (par exemple, ServiceName).
5. L'onglet All metrics (Toutes les métriques) affiche toutes les métriques pour cette dimension dans l'espace de noms. Vous pouvez effectuer les actions suivantes :

    a. Pour trier le tableau, utilisez l’en-tête de colonne.

    b. Pour représenter graphiquement une métrique, cochez la case en regard de la métrique. Pour sélectionner toutes les métriques, cochez la case dans la ligne d’en-tête du tableau.

    c. Pour filtrer par ressource, sélectionnez l'ID de ressource, puis Add to search (Ajouter à la recherche).

    d. Pour filtrer par métrique, choisissez le nom de la métrique, puis Add to search (Ajouter à la recherche).

Interrogation des journaux à l'aide de CloudWatch Logs Insights

Vous pouvez interroger les événements de journal détaillés associés aux métriques extraites à l'aide de CloudWatch Logs Insights pour obtenir des informations approfondies sur les causes profondes des événements opérationnels. L'un des avantages de l'extraction de métriques à partir de vos journaux est que vous pouvez filtrer vos journaux ultérieurement en fonction de la métrique unique (nom de métrique et jeu de dimensions unique) et des valeurs de métrique, afin d'obtenir un contexte sur les événements qui ont contribué à la valeur de métrique agrégée

Par exemple, pour obtenir un identifiant de demande concerné ou un identifiant de suivi par rayons X, vous pouvez exécuter la requête suivante dans CloudWatch Logs Insights.

```
filter Latency > 1000 and Operation = "Aggregator"
```

```
| fields RequestId, TraceId
```

Vous pouvez également effectuer une agrégation de temps de requête sur des clés de cardinalité élevée, par exemple en recherchant les clients touchés par un événement. L'exemple suivant illustre ce scénario.

```
filter Latency > 1000 and Operation = "Aggregator"
| stats count() by CustomerId
```

Pour plus d'informations, voir Analyse des données de journal avec CloudWatch Logs Insights

Configuration d'alertes sur les métriques créées avec le format de métrique intégrée

En général, la création d'alertes sur des métriques générées par le format de métrique intégrée suit le même schéma que la création d'alertes sur toute autre métrique. Pour plus d'informations, consultez Utilisation des CloudWatch alarmes Amazon.

La génération de métriques au format métrique intégré dépend du flux de publication de vos journaux, car les journaux doivent être traités par CloudWatch Logs pour être transformés en métriques. Il est donc important que vous publiiez les journaux en temps opportun afin que vos points de données de métriques soient créés pendant la période pendant laquelle les alertes sont évaluées.

Si vous prévoyez d'utiliser un format métrique intégré pour envoyer des métriques haute résolution et créer des alarmes sur ces métriques, nous vous recommandons de vider les journaux dans les CloudWatch journaux à un intervalle de 5 secondes ou moins afin d'éviter tout délai supplémentaire susceptible de provoquer des alarmes en cas de données partielles ou manquantes. Si vous utilisez l' CloudWatch agent, vous pouvez ajuster l'intervalle de vidange en définissant le `force_flush_interval` paramètre dans le fichier de configuration de l' CloudWatch agent. Cette valeur est définie par défaut sur 5 secondes.

Si vous utilisez Lambda sur d'autres plateformes où vous ne pouvez pas contrôler l'intervalle de vidange du journal, pensez à utiliser des alertes « M sur N » pour contrôler le nombre de points de données utilisés pour l'alerte. Pour plus d'informations, voir Évaluation d'une alerte.

AWS services qui publient CloudWatch des statistiques

Les AWS services suivants publient des métriques sur CloudWatch. Pour plus d'informations sur les métriques et les dimensions, consultez la documentation spécifiée.

Service	Espace de noms	Documentation
AWS Amplify	`AWS/AmplifyHosting`	Surveillance
Amazon API Gateway	`AWS/ApiGateway`	Surveillez l'exécution des API avec Amazon CloudWatch
Amazon AppFlow	`AWS/AppFlow`	Surveillance d'Amazon AppFlow avec Amazon CloudWatch
AWS Service de migration d'applications	`AWS/MGN`	Supervision du service de migration des applications avec Amazon CloudWatch
AWS App Runner	`AWS/AppRunner`	Affichage des statistiques de service App Runner signalées à CloudWatch
AppStream 2,0	`AWS/AppStream`	Surveillance des ressources Amazon AppStream 2.0
AWS AppSync	`AWS/AppSync`	CloudWatch Métriques
Amazon Athena	`AWS/Athena`	Surveillance des requêtes Athena à l'aide de métriques CloudWatch
Amazon Aurora	`AWS/RDS`	Métriques Amazon Aurora
AWS Backup	`AWS/Backup`	Surveillance des métriques AWS de sauvegarde avec CloudWatch
Amazon Bedrock	`AWS/Bedrock`	Surveillance d'Amazon Bedrock avec Amazon CloudWatch

Service	Espace de noms	Documentation
AWS Billing and Cost Management	`AWS/Billing`	Surveillance des coûts à l'aide d'alertes et de notifications
Amazon Braket	`AWS/Braket/By Device`	Surveillance d'Amazon Braket avec Amazon CloudWatch
AWS Certificate Manager	`AWS/CertificateManager`	CloudWatch Métriques prises en charge
Autorité de certification privée AWS	`AWS/ACMPrivateCA`	CloudWatch Métriques prises en charge
AWS Chatbot	`AWS/Chatbot`	Surveillance AWS Chatbot avec Amazon CloudWatch
Amazon Chime	`AWS/ChimeVoiceConnector`	Surveillance Amazon Chime avec Amazon CloudWatch
Kit SDK Amazon Chime	`AWS/ChimeSDK`	Métriques de service
AWS Client VPN	`AWS/ClientVPN`	Surveillance avec Amazon CloudWatch
Amazon CloudFront	`AWS/CloudFront`	Surveillance de CloudFront l'activité à l'aide CloudWatch
AWS CloudHSM	`AWS/CloudHSM`	Obtenir CloudWatch des métriques
Amazon CloudSearch	`AWS/CloudSearch`	Surveillance d'un CloudSearch domaine Amazon avec Amazon CloudWatch

Service	Espace de noms	Documentation
AWS CloudTrail	`AWS/CloudTrail`	CloudWatch Métriques prises en charge
CloudWatch agent	`CWAgent` ou un espace de noms personnalisé	Métriques collectées par l' CloudWatch agent
CloudWatch flux métriques	`AWS/CloudWatch/MetricStreams`	Surveillance de vos flux de mesures à l'aide de CloudWatch métriques
CloudWatch RHUM	`AWS/RUM`	CloudWatch métriques que vous pouvez collecter avec CloudWatch RUM
CloudWatch Synthetics	`CloudWatchSynthetics`	CloudWatch statistiques publiées par canaries
Amazon CloudWatch Logs	`AWS/Logs`	Surveillance de l'utilisation à l'aide de CloudWatch métriques
AWS CodeBuild	`AWS/CodeBuild`	Surveillance AWS CodeBuild
CodeGuru Réviseur Amazon		Surveillance du CodeGuru réviseur avec Amazon CloudWatch
Amazon Kendra		Surveiller Amazon Kendra avec Amazon CloudWatch
Amazon CodeWhisperer	`AWS/CodeWhisperer`	Surveillance Amazon CodeWhisperer avec Amazon CloudWatch
Amazon Cognito	`AWS/Cognito`	Surveillance d'Amazon Cognito

Service	Espace de noms	Documentation
Amazon Comprehend	`AWS/Comprehend`	Surveillance des points de Amazon Comprehend terminaison
AWS Config	`AWS/Config`	AWS Config Indicateurs d'utilisation et de réussite
Amazon Connect	`AWS/Connect`	Surveillance d'Amazon Connect dans Amazon CloudWatch Metrics
Amazon Data Lifecycle Manager	`AWS/DataLifecycleManager`	Surveillez vos politiques à l'aide d'Amazon CloudWatch
AWS DataSync	`AWS/DataSync`	Surveillance de votre tâche
Amazon DataZone		Surveillance d'Amazon DataZone avec Amazon CloudWatch
Amazon DevOps Guru	`AWS/DevOps-Guru`	Surveillance Amazon DevOps Guru avec Amazon CloudWatch
AWS Database Migration Service	`AWS/DMS`	AWS DMS Tâches de surveillance
AWS Direct Connect	`AWS/DX`	Surveillance avec Amazon CloudWatch
AWS Directory Service	`AWS/DirectoryService`	Utilisez CloudWatch les métriques Amazon pour déterminer quand ajouter des contrôleurs de domaine
Amazon DocumentDB	`AWS/DocDB`	Métriques Amazon DocumentDB
Amazon DynamoDB	`AWS/DynamoDB`	Métriques et dimensions DynamoDB

Service	Espace de noms	Documentation
DynamoDB Accelerator (DAX)	`AWS/DAX`	Affichage des métriques et dimensions DAX
Amazon EC2	`AWS/EC2`	Surveillance de vos instances à l'aide de CloudWatch
Amazon EC2 Elastic Graphics	`AWS/ElasticGPUs`	Utilisation de CloudWatch métriques pour surveiller Elastic Graphics
Amazon EC2 Spot	`AWS/EC2Spot`	CloudWatch Indicateurs pour Spot Fleet
Amazon EC2 Auto Scaling	`AWS/AutoScaling`	Surveillance de vos groupes et instances Auto Scaling à l'aide de CloudWatch
AWS Elastic Beanstalk	`AWS/ElasticBeanstalk`	Publication de métriques CloudWatch personnalisées Amazon pour un environnement
Amazon Elastic Block Store	`AWS/EBS`	Amazon CloudWatch Metrics pour Amazon EBS
Amazon Elastic Container Registry	`AWS/ECR`	Métriques de référentiel Amazon ECR
Amazon Elastic Container Service	`AWS/ECS`	CloudWatch Métriques Amazon ECS
Amazon ECS via CloudWatch Container Insights	`ECS/ContainerInsights`	Métriques Container Insights pour Amazon ECS

Service	Espace de noms	Documentation
Auto Scaling du cluster Amazon ECS	`AWS/ECS/ManagedScaling`	Auto Scaling de cluster Amazon ECS
AWS Elastic Disaster Recovery		CloudWatch Métriques pour le DRS
Amazon Elastic File System	`AWS/EFS`	Surveillance avec CloudWatch
Amazon Elastic Inference	`AWS/ElasticInference`	Utilisation de CloudWatch métriques pour surveiller Amazon Elastic Inference
Amazon EKS via CloudWatch Container Insights	`ContainerInsights`	Métriques Container Insights pour Amazon EKS et Kubernetes
Elastic Load Balan	`AWS/ApplicationELB`	CloudWatch Mesures pour votre Application Load Balancer
Elastic Load Balan	`AWS/NetworkELB`	CloudWatch Indicateurs pour votre Network Load Balancer
Elastic Load Balan	`AWS/GatewayELB`	CloudWatch Indicateurs pour votre Gateway Load Balancer
Elastic Load Balan	`AWS/ELB`	CloudWatch Indicateurs pour votre Classic Load Balancer
Amazon Elastic Transcoder	`AWS/ElasticTranscoder`	Surveillance avec Amazon CloudWatch

Service	Espace de noms	Documentation
Amazon ElastiCache pour Memcached	`AWS/ElastiCache`	Surveillance de l'utilisation à l'aide de CloudWatch métriques
Amazon ElastiCache pour Redis	`AWS/ElastiCache`	Surveillance de l'utilisation à l'aide de CloudWatch métriques
Amazon OpenSearch Service	`AWS/ES`	Surveillance des métriques OpenSearch du cluster avec Amazon CloudWatch
Amazon EMR	`AWS/ElasticMapReduce`	Surveillez les métriques avec CloudWatch
AWS Elemental MediaConnect	`AWS/MediaConnect`	Surveillance MediaConnect avec Amazon CloudWatch
AWS Elemental MediaConvert	`AWS/MediaConvert`	Utilisation CloudWatch des métriques pour afficher les métriques relatives aux AWS Elemental MediaConvert ressources
AWS Elemental MediaLive	`AWS/MediaLive`	Surveillance de l'activité à l'aide CloudWatch des métriques Amazon
AWS Elemental MediaPackage	`AWS/MediaPackage`	Surveillance AWS Elemental MediaPackage avec Amazon CloudWatch Metrics
AWS Elemental MediaStore	`AWS/MediaStore`	Surveillance AWS Elemental MediaStore avec Amazon CloudWatch Metrics
AWS Elemental MediaTailor	`AWS/MediaTailor`	Surveillance AWS Elemental MediaTailor avec Amazon CloudWatch
Amazon EventBridge	`AWS/Events`	Surveillance d'Amazon EventBridge

Service	Espace de noms	Documentation
Amazon FinSpace		Journalisation et surveillance
Amazon Forecast		CloudWatch Métriques pour Amazon Forecast
Amazon Fraud Detector		Surveillance d'Amazon Fraud Detector avec Amazon CloudWatch
Amazon FSx for Lustre	`AWS/FSx`	Surveillance d'Amazon FSx for Lustre
Amazon FSx pour OpenZFS	`AWS/FSx`	Surveillance avec Amazon CloudWatch
Amazon FSx for Windows File Server	`AWS/FSx`	Surveillance d'Amazon FSx for Windows File Server
Amazon FSx pour ONTAP NetApp	`AWS/FSx`	Surveillance avec Amazon CloudWatch
Amazon FSx pour OpenZFS	`AWS/FSx`	Surveillance avec Amazon CloudWatch
Amazon GameLift	`AWS/GameLift`	Surveillez Amazon GameLift avec CloudWatch
AWS Global Accelerator	`AWS/GlobalAccelerator`	Utiliser Amazon CloudWatch avec AWS Global Accelerator
AWS Glue	`Glue`	Surveillance à AWS Glue l'aide de CloudWatch métriques
AWS Ground Station	`AWS/GroundStation`	Métriques utilisant Amazon CloudWatch

Service	Espace de noms	Documentation
AWS HealthLake	`AWS/HealthLake`	Surveillance HealthLake avec CloudWatch
Amazon Inspector	`AWS/Inspector`	Surveillance d'Amazon Inspector à l'aide CloudWatch
Amazon Interactive Video Service	`AWS/IVS`	Surveillance d'Amazon IVS avec Amazon CloudWatch
Service de vidéo interactive Amazon Chat	`AWS/IVSChat`	Surveillance d'Amazon IVS avec Amazon CloudWatch
AWS IoT	`AWS/IoT`	AWS IoT Métriques et dimensions
AWS IoT Analytics	`AWS/IoTAnalytics`	Espaces de noms, métriques et dimensions
AWS IoT FleetWise	`AWS/IoTFleetWise`	Surveillance de AWS l'IoT FleetWise avec Amazon CloudWatch
AWS IoT SiteWise	`AWS/IoTSiteWise`	Surveillance AWS IoT SiteWise à l'aide des CloudWatch métriques Amazon
AWS IoT TwinMaker	`AWS/IoTTwinMaker`	Surveillance AWS IoT TwinMaker à l'aide des CloudWatch métriques Amazon
AWS IoT 1 clic		Surveillance en AWS IoT 1 clic avec Amazon CloudWatch
AWS Key Management Service	`AWS/KMS`	Surveillance avec CloudWatch

Service	Espace de noms	Documentation
Amazon Keyspaces (pour Apache Cassandra)	`AWS/Cassandra`	Métriques et dimensions Amazon Keyspaces
Amazon Kendra		Surveiller Amazon Kendra avec Amazon CloudWatch
Service géré Amazon pour Apache Flink	`AWS/KinesisAnalytics`	Service géré pour les applications Apache Flink pour SQL : surveillance avec CloudWatch Service géré pour Apache Flink pour Apache Flink : affichage des métriques et dimensions de service géré Amazon pour Apache Flink
Amazon Data Firehose	`AWS/Firehose`	Surveillance de Firehose à l'aide de métriques CloudWatch
Amazon Kinesis Data Streams	`AWS/Kinesis`	Surveillance d'Amazon Kinesis Data Streams avec Amazon CloudWatch
Amazon Kinesis Vi	`AWS/KinesisVideo`	Surveillance des métriques de Kinesis Video Streams avec CloudWatch
AWS Lambda	`AWS/Lambda`	AWS Lambda Métriques
Amazon Lex	`AWS/Lex`	Surveillance d'Amazon Lex avec Amazon CloudWatch

AWS License Manager	`AWSLicenseManager/licenseUsage`
	
	`AWS/LicenseManager/LinuxSubscriptions`
		Surveillance de l'utilisation des licences avec Amazon CloudWatch
		
		Mesures d'utilisation et CloudWatch alarmes Amazon pour les abonnements Linux

Service	Espace de noms	Documentation
Amazon Location Service	`AWS/Location`	Métriques Amazon Location Service exportées vers Amazon CloudWatch
Amazon Lookout for Equipment	`AWS/lookoutequipment`	Surveillance de Lookout for Equipment avec Amazon CloudWatch
Amazon Lookout for Metrics	`AWS/LookoutMetrics`	Surveillance de Lookout for Metrics avec Amazon CloudWatch
Amazon Lookout for Vision	`AWS/LookoutVision`	Surveillance de Lookout for Vision avec Amazon CloudWatch
AWS Modernisation du mainframe		Surveillance de la modernisation des AWS mainframes avec Amazon CloudWatch
Amazon Machine Learning	`AWS/ML`	Surveillance d'Amazon ML à l'aide de CloudWatch métriques
Amazon Managed Blockchain	`AWS/managedblockchain`	Utilisation des métriques Hyperledger Fabric Peer Node sur Amazon Managed Blockchain
Amazon Managed Service for Prometheus	`AWS/Prometheus`	CloudWatch Métriques Amazon
Amazon Managed Streaming for Apache Kafka	`AWS/Kafka`	Surveillance d'Amazon MSK avec Amazon CloudWatch

Service	Espace de noms	Documentation
Amazon Managed Streaming for Apache Kafka	`AWS/Kafka Connect`	Surveillance de MSK Connect
Amazon Managed Workflows for Apache Airflow	`AWS/MWAA`	Mesures relatives aux conteneurs, aux files d'attente et aux bases de données pour Amazon MWAA
Amazon MemoryDB for Redis	`AWS/MemoryDB`	CloudWatch Métriques de surveillance
Amazon MQ	`AWS/AmazonMQ`	Surveillance des courtiers Amazon MQ à l'aide d'Amazon CloudWatch
Amazon Neptune	`AWS/Neptune`	Surveiller Neptune avec CloudWatch
AWS Network Firewall	`AWS/NetworkFirewall`	AWS Network Firewall statistiques sur Amazon CloudWatch
AWS Directeur du réseau	`AWS/NetworkManager`	CloudWatch métriques pour les ressources sur site
Amazon Nimble Studio	`AWS/NimbleStudio`	Surveillance de Nimble Studio avec Amazon CloudWatch
AWS HealthOmics	`AWS/Omics`	Surveillance AWS HealthOmics avec Amazon CloudWatch
AWS OpsWorks	`AWS/OpsWorks`	Surveillance de Stacks à l'aide d'Amazon CloudWatch
AWS Outposts	`AWS/Outposts`	CloudWatch métriques pour AWS Outposts

Service	Espace de noms	Documentation
AWS Panorama	`AWS/PanoramaDeviceMetrics`	Surveillance des appareils et des applications avec Amazon CloudWatch
Amazon Personalize	`AWS/Personalize`	CloudWatch statistiques pour Amazon Personalize
Amazon Pinpoint	`AWS/Pinpoint`	Afficher Amazon Pinpoint les métriques dans CloudWatch
Amazon Polly	`AWS/Polly`	Intégration CloudWatch à Amazon Polly
AWS PrivateLink	`AWS/PrivateLinkEndpoints`	CloudWatch métriques pour AWS PrivateLink
AWS PrivateLink	`AWS/PrivateLinkServices`	CloudWatch métriques pour AWS PrivateLink
AWS 5G privée	`AWS/Private5G`	CloudWatch Métriques Amazon
Amazon QLDB	`AWS/QLDB`	Surveillance des données sur Amazon QuickSight
Amazon QuickSight	`AWS/QuickSight`	Surveillance avec Amazon CloudWatch
Amazon Redshift	`AWS/Redshift`	Données de performances d'Amazon Redshift
Amazon Relational Database Service	`AWS/RDS`	Surveillance des métriques Amazon RDS avec Amazon CloudWatch

Service	Espace de noms	Documentation
Amazon Rekognition	`AWS/Rekognition`	Surveillance de la Rekognition avec Amazon CloudWatch
AWS re:Post Privé	`AWS/rePostPrivate`	Surveillance AWS re:Post privée avec Amazon CloudWatch
AWS RoboMaker	`AWS/Robomaker`	Surveillance AWS RoboMaker avec Amazon CloudWatch
Amazon Route 53	`AWS/Route53`	Surveillance d'Amazon Route 53
Application Recovery Controller Amazon Route 53	`AWS/Route53RecoveryReadiness`	Utilisation d'Amazon CloudWatch avec Application Recovery Controller
Amazon SageMaker	`AWS/SageMaker`	Surveillance SageMaker avec CloudWatch
Amazon SageMaker Model Building Pipelines	`AWS/SageMaker/ModelBuildingPipeline`	SageMaker Métriques des pipelines
AWS Secrets Manager	`AWS/SecretsManager`	Supervision de Secrets Manager avec Amazon CloudWatch
Amazon Security Lake	`AWS/SecurityLake`	CloudWatch métriques pour Amazon Security Lake
Service Catalog	`AWS/ServiceCatalog`	CloudWatch Métriques du Service Catalog
AWS Shield Advanced	`AWS/DDoSProtection`	Surveillance avec CloudWatch

Service	Espace de noms	Documentation
Amazon Simple En	AWS/SES	Récupération des données d'événements Amazon SES depuis CloudWatch
AWS SimSpace Weaver	AWS/simspaceweaver	Surveillance AWS SimSpace Weaver avec Amazon CloudWatch
Amazon Simple Notification Service	AWS/SNS	Surveillance d'Amazon SNS avec CloudWatch
Amazon Simple Queue Service	AWS/SQS	Surveillance des files d'attente Amazon SQS à l'aide de CloudWatch
Amazon S3	AWS/S3	Surveillance des métriques avec Amazon CloudWatch
S3 Storage Lens	AWS/S3/Storage-Lens	Surveillez les métriques de S3 Storage Lens dans CloudWatch
Amazon Simple Workflow Service	AWS/SWF	Métriques Amazon SWF pour CloudWatch
AWS Step Functions	AWS/States	Surveillance des fonctions Step Functions à l'aide CloudWatch
AWS Storage Gateway	AWS/StorageGateway	Utilisation des CloudWatch métriques Amazon
AWS Systems Manager Exécuter la commande	AWS/SSM-RunCommand	Surveillance des métriques d'exécution des commandes à l'aide CloudWatch
Amazon Textract	AWS/Textract	CloudWatch Métriques pour Amazon Textract

Service	Espace de noms	Documentation
Amazon Timestream	`AWS/Timestream`	Métriques et dimensions Timestream
AWS Transfer for SFTP	`AWS/Transfer`	AWS SFTP CloudWatch Métriques
Amazon Transcribe	`AWS/Transcribe`	Surveillance Amazon Transcribe avec Amazon CloudWatch
Amazon Translate	`AWS/Translate`	CloudWatch Mesures et dimensions pour Amazon Translate
AWS Trusted Advisor	`AWS/TrustedAdvisor`	Création d'alarmes Trusted Advisor à l'aide de CloudWatch
Amazon VPC	`AWS/NATGateway`	Surveillance de votre passerelle NAT avec CloudWatch
Amazon VPC	`AWS/TransitGateway`	CloudWatch Indicateurs pour vos passerelles de transport en commun
Amazon VPC	`AWS/VPN`	Surveillance avec CloudWatch
Amazon VPC IP Address Manager (IPAM)	`AWS/IPAM`	Créez des alarmes avec Amazon CloudWatch
AWS WAF	`AWS/WAFV2` pour les AWS WAF ressources `WAF`pour les ressources AWS WAF classiques	Surveillance avec CloudWatch
Amazon WorkMail	`AWS/WorkMail`	Surveillance Amazon WorkMail avec Amazon CloudWatch

Service	Espace de noms	Documentation
Amazon WorkSpaces	`AWS/WorkSpaces`	Surveillez vos CloudWatch indicateurs WorkSpaces d'utilisation
WorkSpaces Site Web d'Amazon	`AWS/WorkSpacesWeb`	Surveillance d'Amazon WorkSpaces Web avec Amazon CloudWatch

AWS métriques d'utilisation

CloudWatch collecte des métriques permettant de suivre l'utilisation de certaines AWS ressources et API. Ces métriques sont publiées dans l'espace de noms `AWS/Usage`. Les métriques d'utilisation vous CloudWatch permettent de gérer l'utilisation de manière proactive en visualisant les métriques dans la CloudWatch console, en créant des tableaux de bord personnalisés, en détectant les changements d'activité grâce à la détection des CloudWatch anomalies et en configurant des alarmes qui vous alertent lorsque l'utilisation approche un seuil.

Certains AWS services intègrent ces indicateurs d'utilisation aux Service Quotas. Pour ces services, vous pouvez les utiliser CloudWatch pour gérer l'utilisation de vos quotas de service par votre compte. Pour plus d'informations, consultez Visualisation de vos quotas de service et définition d'alertes.

Rubriques

Visualisation de vos quotas de service et définition d'alertes

Pour certains AWS services, vous pouvez utiliser les indicateurs d'utilisation pour visualiser votre utilisation actuelle des services sur CloudWatch des graphiques et des tableaux de bord. Vous pouvez utiliser une fonction mathématique CloudWatch métrique pour afficher les quotas de service pour ces ressources sur vos graphiques. Vous pouvez également configurer des alertes qui vous alertent lorsque votre utilisation approche d'un Service Quota. Pour de plus amples informations sur les quotas de service, veuillez consulter Qu'est-ce que Service Quotas ? dans le Guide de l'utilisateur Service Quotas.

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 utiliser ce compte de surveillance pour visualiser les quotas de service et définir des alarmes pour les métriques des comptes sources liés à ce compte de surveillance. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

Actuellement, les services suivants intègrent leurs métriques d'utilisation avec les Service Quotas :

- AWS CloudHSM
- Kit SDK Amazon Chime
- Amazon CloudWatch
- Amazon CloudWatch Logs
- Amazon DynamoDB
- Amazon EC2
- Amazon Elastic Container Registry
- Elastic Load Balancing
- AWS Fargate
- AWS Fault Injection Service
- AWS Service vidéo interactif
- AWS Key Management Service
- Amazon Data Firehose
- Amazon Location Service
- Requête Amazon Managed Blockchain (AMB)
- AWS RoboMaker
- Amazon SageMaker

Visualiser un quota de service et définir éventuellement une alerte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Dans l'onglet Toutes les mesures, choisissez Utilisation, puis Par AWS ressource.

   La liste des métriques d'utilisation des Service Quotas s'affiche.
4. Cochez la case en regard de l'une des métriques.

   Le graphique indique votre utilisation actuelle de cette AWS ressource.
5. Pour ajouter votre quota de service au graphique, procédez comme suit :

   a. Sélectionnez l'onglet Graphed metrics (Graphiques des métriques).

b. Choisissez Math expression (Expression mathématique), puis Start with an empty expression (Commencer par une expression vide). Dans la nouvelle ligne, sous Details (Détails), entrez **`SERVICE_QUOTA(m1)`**.

   Une nouvelle ligne est ajoutée au graphique avec le quota de service de la ressource représentée dans la métrique.

6. Pour afficher votre utilisation actuelle sous forme de pourcentage du quota, ajoutez une nouvelle expression ou modifiez l'expression SERVICE_QUOTA actuelle. La nouvelle expression à utiliser est **`m1/SERVICE_QUOTA(m1)*100`**.
7. (Facultatif) Pour définir une alerte qui vous avertit si vous approchez du quota de service, procédez comme suit :

   a. Sur la ligne avec **`m1/SERVICE_QUOTA(m1)*100`**, sous Actions, choisissez l'icône d'alerte. Elle ressemble à une cloche.

      La page de création d'alerte s'affiche.

   b. Sous Conditions, vérifiez que le Threshold type (Type de seuil) est Static (Statique) et que Whenever Expression1 is (Lorsque Expression1 est) est défini sur Greater (Supérieur). Sous than (à), entrez **`80`**. Cela crée une alerte qui passe à l'état alerte lorsque votre utilisation dépasse 80 % du quota.

   c. Choisissez Next (Suivant).

   d. Sur la page suivante, sélectionnez une rubrique Amazon SNS ou créez-en une, puis choisissez Next (Suivant). La rubrique que vous avez sélectionnée est notifiée lorsque l'alerte passe à l'état alerte.

   e. Sur la page suivante, entrez le nom et la description de l'alerte, puis choisissez Next (Suivant).

   f. Sélectionnez Create alarm (Créer une alerte).

AWS Métriques d'utilisation de l'API

La plupart des API qui prennent en charge la AWS CloudTrail journalisation transmettent également des statistiques d'utilisation à CloudWatch. Les métriques d'utilisation des API vous CloudWatch permettent de gérer de manière proactive l'utilisation des API en visualisant les métriques dans la CloudWatch console, en créant des tableaux de bord personnalisés, en détectant les changements

d'activité grâce à la détection des CloudWatch anomalies et en configurant des alarmes qui alertent lorsque l'utilisation approche un seuil.

Le tableau suivant répertorie les services auxquels les indicateurs d'utilisation des API sont CloudWatch transmis, ainsi que la valeur à utiliser pour que la `Service` dimension affiche les indicateurs d'utilisation de ce service.

Service	Valeur pour la dimension **`Service`**
AWS Identity and Access Management Access Analyzer	`Access Analyzer`
AWS Account Management	`Account Management`
Alexa for Business	`A4B`
Amazon API Gateway	`API Gateway`
AWS App Mesh	`App Mesh`
AWS AppConfig	`AWS AppConfig`
Amazon AppFlow	`AppFlow`
Application Auto Scaling	`Application Auto Scaling`
Application Discovery Service	`Application Discovery Service`
Amazon AppStream	`AppStream`
AppStream Générateur d'Image 2.0	`Image Builder`
Amazon Athena	`Athena`
AWS Audit Manager	`Audit Manager`
AWS Backup	`Backup`
AWS Batch	`Batch`
Amazon Braket	`Braket`

Service	Valeur pour la dimension **`Service`**
AWS Budgets	`Budgets`
AWS Certificate Manager	`Certificate Manager`
Kit SDK Amazon Chime	`ChimeSDK`
Amazon Cloud Directory	`Cloud Directory`
AWS Cloud Map	`Cloud Map`
AWS CloudFormation	`CloudFormation`
AWS CloudHSM	`CloudHSM`
Amazon CloudSearch	`CloudSearch`
AWS CloudShell	`CloudShell`
AWS CloudTrail	`CloudTrail`
Amazon CloudWatch	`CloudWatch`
Amazon CloudWatch Logs	`Logs`
Informations sur les CloudWatch applications Amazon	`CloudWatch Application Insights`
AWS CodeBuild	`CodeBuild`
AWS CodeCommit	`CodeCommit`
Amazon CodeGuru Profiler	`CodeGuru Profiler`
AWS CodePipeline	`CodePipeline`
AWS CodeStar	`CodeStar`
AWS CodeStar Notifications	`CodeStar Notifications`
AWS CodeStar Connexions	`CodeStar Connections`

Service	Valeur pour la dimension **`Service`**
Groupes d'identités Amazon Cognito	`Cognito Identity Pools`
Amazon Cognito Sync	`Cognito Sync`
Amazon Comprehend	`Comprehend`
Amazon Comprehend Medical	`Comprehend Medical`
AWS Compute Optimizer	`ComputeOptimzier`
Amazon Connect	`Connect`
Profils des clients Amazon Connect	`Customer Profiles`
AWS Rapports sur les coûts et l'utilisation	`Cost and Usage Report`
AWS Cost Explorer	`Cost Explorer`
AWS Data Exchange	`Data Exchange`
AWS Gestionnaire du cycle de vie des données	`Data Lifecycle Manager`
AWS Database Migration Service	`Database Migration Service`
AWS DataSync	`DataSync`
AWS DeepLens	`AWS DeepLens`
Amazon Detective	`Detective`
Device Advisor	`Device Advisor`
AWS Direct Connect	`Direct Connect`
AWS Directory Service	`Directory Service`
DynamoDB Accelerator	`DynamoDBAccelerator`
Amazon EC2	`EC2`

Service	Valeur pour la dimension **`Service`**
EC2 Auto Scaling	`EC2 Auto Scaling`
Amazon Elastic Container Registry	`ECR Public`
Amazon Elastic Container Service	`ECS`
Amazon Elastic File System	`EFS`
Amazon Elastic Kubernetes Service	`EKS`
AWS Elastic Beanstalk	`Elastic Beanstalk`
Amazon Elastic Inference	`Elastic Inference`
Elastic Load Balancing	`Elastic Load Balancing`
Amazon EMR	`EMR Containers`
AWS Firewall Manager	`Firewall Manager`
Amazon FSx	`FSx`
Amazon GameLift	`GameLift`
AWS Glue DataBrew	`DataBrew`
Amazon Managed Grafana	`Grafana`
AWS IoT Greengrass	`Greengrass`
AWS Ground Station	`Ground Station`
AWS Health API et notifications	`AWS Health APIs And Notifications`
Amazon Interactive Video Service	`IVS`
AWS IoT Core	`IoT`
AWS IoT 1 clic	`IoT 1-Click`

Service	Valeur pour la dimension **`Service`**
AWS IoT Events	`IoT Events`
AWS IoT RoboRunner	`IoT RoboRunner`
AWS IoT SiteWise	`IoT Sitewise`
AWS IoT Wireless	`IoT Wireless`
Amazon Kendra	`Kendra`
Amazon Keyspaces (pour Apache Cassandra)	`Keyspaces`
Service géré Amazon pour Apache Flink	`Kinesis Analytics`
Amazon Data Firehose	`Firehose`
Kinesis Video Streams	`Kinesis Video Streams`
AWS Key Management Service	`KMS`
AWS Lambda	`Lambda`
AWS Launch Wizard	`Launch Wizard`
Amazon Lex	`Amazon Lex`
Amazon Lightsail	`Lightsail`
Amazon Location Service	`Location`
Amazon Lookout for Vision	`Lookout for Vision`
Amazon Machine Learning	`Amazon Machine Learning`
Amazon Macie	`Macie`
Requête Amazon Managed Blockchain (AMB)	`Amazon Managed Blockchain Query`
AWS Managed Services	`AWS Managed Services`

Service	Valeur pour la dimension **`Service`**
AWS Marketplace Commerce Analytics	`Marketplace Analytics Service`
AWS Elemental MediaConnect	`MediaConnect`
AWS Elemental MediaConvert	`MediaConvert`
AWS Elemental MediaLive	`MediaLive`
AWS Elemental MediaStore	`Mediastore`
AWS Elemental MediaTailor	`MediaTailor`
AWS Mobile Hub	`Mobile Hub`
AWS Network Firewall	`Network Firewall`
AWS OpsWorks	`OpsWorks`
AWS OpsWorks pour la gestion des configurations	`OPsWorks CM`
AWS Outposts	`Outposts`
AWS Organizations	`Organizations`
Analyse des performances d’Amazon RDS	`Performance Insights`
Amazon Pinpoint	`Pinpoint`
AWS Private Certificate Authority	`Private Certificate Authority`
Amazon Managed Service for Prometheus	`Prometheus`
AWS Proton	`Proton`
Amazon Quantum Ledger Database (Amazon QLDB)	`QLDB`
Amazon RDS	`RDS`

Service	Valeur pour la dimension **`Service`**
Amazon Redshift	`Redshift Data API`
Amazon Rekognition	`Rekognition`
AWS Resource Access Manager	`Resource Access Manager`
AWS Resource Groups	`Resource Groups`
AWS Resource Groups Tagging API	`Resource Groups Tagging API`
AWS RoboMaker	`RoboMaker`
Amazon Route 53 Domaines	`Route 53 Domains`
Amazon Route 53 Resolver	`Route 53 Resolver`
Amazon S3	`S3`
Amazon S3 Glacier	`Amazon S3 Glacier`
Amazon SageMaker Runtime	`Sagemaker`
Savings Plans	`Savings Plans`
AWS Secrets Manager	`Secrets Manager`
AWS Security Hub	`Security Hub`
AWS Server Migration Service	`AWS Server Migration Service`
AWS Service Catalog AppRegistry	`Service Catalog AppRegistry`
Service Quotas	`Service Quotas`
AWS Shield	`Shield`
AWS Signataire	`Signer`
Amazon Simple Notification Service	`SNS`

Service	Valeur pour la dimension **`Service`**
Amazon Simple Email Service	`SES`
Amazon Simple Queue Service	`SQS`
Identity Store	`Identity Store`
Storage Gateway	`Storage Gateway`
AWS Support	`Support`
Amazon Simple Workflow Service	`SWF`
Amazon Textract	`Textract`
AWS IoT Things Graph	`ThingsGraph`
Amazon Timestream	`Timestream`
Amazon Transcribe	`Transcribe`
Amazon Translate	`Translate`
Transcription Amazon Transcribe Streaming	`Transcribe Streaming`
AWS Transfer Family	`Transfer`
AWS WAF	`WAF`
Amazon WorkDocs	`Amazon WorkDocs`
Amazon WorkLink	`WorkLink`
Amazon WorkMail	`Amazon WorkMail`
Amazon WorkSpaces	`Workspaces`
AWS X-Ray	`X-Ray`

Certains services signalent également des métriques d'utilisation pour d'autres API. Pour savoir si une API communique des métriques d'utilisation à CloudWatch, utilisez la CloudWatch console pour voir les métriques signalées par ce service dans l'espace de `AWS/Usage` noms.

Pour consulter la liste des API d'un service qui transmettent des statistiques d'utilisation à CloudWatch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Métriques.
3. Dans l'onglet Toutes les mesures, choisissez Utilisation, puis Par AWS ressource.
4. Dans la zone de recherche située à côté de la liste des métriques, saisissez le nom du service. Les métriques sont filtrées par le service que vous avez entré.

CloudWatch métriques d'utilisation

CloudWatch collecte des métriques qui permettent de suivre l'utilisation de certaines AWS ressources. 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 Visualisation de vos quotas de service et définition d'alertes.

Les métriques d'utilisation des quotas de service se trouvent dans l'espace de noms `AWS/Usage` et sont collectées chaque minute.

Actuellement, le seul nom de métrique publié dans cet espace de noms CloudWatch est`CallCount`. Cette métrique est publiée avec les dimensions `Resource`, `Service` et `Type`. La dimension `Resource` spécifie le nom de l'opération d'API suivie. Par exemple, la `CallCount` métrique avec les dimensions `"Service": "CloudWatch"` `"Type": "API"` et `"Resource": "PutMetricData"` indique le nombre de fois que l'opération CloudWatch `PutMetricData` API a été appelée dans votre compte.

La métrique `CallCount` n'a pas d'unité spécifiée. La statistique la plus utile pour la métrique est `SUM`, qui représente le nombre total d'opérations pour une période d'1 minute.

Métriques

Métrique	Description
`CallCount`	Nombre d'opérations spécifiées effectuées dans votre compte.

Dimensions

Dimension	Description
`Service`	Nom du AWS service contenant la ressource. Pour les métriques CloudWatch d'utilisation, la valeur de cette dimension est`CloudWatch` .
`Class`	Classe de ressource suivie. CloudWatch Les métriques d'utilisation des API utilisent cette dimension avec une valeur de`None`.
`Type`	Type de ressource suivi. Actuellement, lorsque la dimension `Service` est `CloudWatch` , la seule valeur valide pour `Type` est `API`.
`Resource`	Nom de l'opération d'API. Les valeurs valides sont notamment les suivantes : `DeleteAlarms` , `DeleteDashboards` , `DescribeAlarmHistory` , `DescribeAlarms` , `GetDashboard` , `GetMetricData` , `GetMetricStatistics` , `ListMetrics` , `PutDashboard` et `PutMetricData`

CloudWatch tutoriels

Les scénarios suivants illustrent les utilisations d'Amazon CloudWatch. Dans le premier scénario, vous utilisez la CloudWatch console pour créer une alarme de facturation qui suit votre AWS consommation et vous indique quand vous avez dépassé un certain seuil de dépenses. Dans le second scénario, plus avancé, vous utilisez le AWS Command Line Interface (AWS CLI) pour publier une seule métrique pour une application hypothétique nommée. GetStarted

Scénarios

- Surveiller vos coûts estimés
- Publication de métriques

Scénario : surveillez vos frais estimés à l'aide de CloudWatch

Dans ce scénario, vous créez une CloudWatch alarme Amazon pour surveiller vos frais estimés. Lorsque vous activez le suivi des frais estimés pour votre AWS compte, les frais estimés sont calculés et envoyés plusieurs fois par jour CloudWatch sous forme de données métriques.

Les données de métriques de facturation sont stockées dans la région USA Est (Virginie du Nord) et reflètent des frais du monde entier. Ces données incluent les frais estimés pour chaque service AWS que vous utilisez, ainsi que le total estimatif global de vos AWS frais.

Vous pouvez choisir de recevoir des alertes par e-mail dès que vos frais dépassent un certain seuil. Ces alertes sont déclenchées par Amazon Simple Notification Service (Amazon SNS) CloudWatch et les messages sont envoyés par ce biais.

> **Note**
>
> Pour plus d'informations sur l'analyse des CloudWatch frais qui vous ont déjà été facturés, consultezCloudWatch facturation et coût.

Tâches

- Étape 1 : activer des alertes de facturation
- Étape 2 : créer une alerte de facturation

Étape 1 : activer des alertes de facturation

Avant de créer une alarme pour vos frais estimés, vous devez activer les alertes de facturation afin de pouvoir surveiller vos AWS frais estimés et créer une alarme à l'aide des données métriques de facturation. Après avoir activé les alertes de facturation, vous ne pouvez pas désactiver la collecte de données, mais vous pouvez supprimer toute alerte de facturation que vous avez créée.

Après avoir activé les alertes de facturation pour la première fois, il faut environ 15 minutes avant de pouvoir afficher des données de facturation et de configurer des alertes de facturation.

Prérequis

- Vous devez être connecté à l'aide des informations d'identification utilisateur root ou en tant qu'utilisateur disposant de l'autorisation d'afficher les informations de facturation.
- Pour les comptes à facturation Consolidée, les données de facturation de chaque compte lié sont disponibles en vous connectant en tant que le compte de règlement. Vous pouvez afficher les données de facturation pour le montant total des coûts estimés et pour les coûts estimés par service pour chaque compte lié, ainsi que pour le compte de regroupement.
- Dans un compte de facturation consolidé, les métriques de compte lié aux membres ne sont capturées que si le compte payeur active la préférence Recevoir des alertes de facturation. Si vous modifiez votre compte gestion/payeur, vous devez activer les alertes de facturation dans le nouveau compte gestion/payeur.
- Le compte ne doit pas faire partie du réseau de partenaires Amazon (APN) car les statistiques de facturation ne sont pas publiées CloudWatch pour les comptes APN. Pour plus d'informations, consultez Réseau de partenaires AWS .

Pour activer la surveillance des coûts estimés

1. Ouvrez la AWS Billing console à l'adresse https://console.aws.amazon.com/billing/.
2. Dans le panneau de navigation, sélectionnez Billing Preferences (Préférences de facturation).
3. Pour Préférences des alertes, choisissez Modifier.

4. Choisissez Recevoir des alertes CloudWatch de facturation.
5. Choisissez Save preferences (Enregistrer des préférences).

Étape 2 : créer une alerte de facturation

> **Important**
>
> Avant de créer une alerte de facturation, vous devez définir votre région sur USA Est (Virginie du Nord). Les données de métriques de facturation sont stockées dans cette région et regroupent des frais du monde entier. Vous devez également activer les alertes de facturation pour votre compte ou dans le compte de gestion/payeur (si vous utilisez la facturation consolidée). Pour plus d'informations, consultez Étape 1 : activer des alertes de facturation.

Dans cette procédure, vous créez une alarme qui envoie une notification lorsque vos frais estimés AWS dépassent un seuil défini.

Pour créer une alarme de facturation 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 Alarms (alertes), puis All alarms (Toutes les alertes).
3. Choisissez Create alarm (Créer une alerte).
4. Choisissez Sélectionner une métrique. Dans Browse (Parcourir), sélectionnez Billing (Facturation), puis Total Estimated Charge (Frais estimés totaux).

   > **Note**
   >
   > Si vous ne voyez pas la métrique Facturation/Total des frais estimés, activez les alertes de facturation et changez votre région pour USA Est (Virginie du Nord). Pour plus d'informations, consultez Activation des alertes de facturation.

5. Cochez la case correspondant à la EstimatedChargesmétrique, puis sélectionnez Select metric.
6. Pour Statistique, choisissez Maximum.
7. Pour Period (Période), choisissez 6 hours (6 heures).
8. Pour Threshold type (Type de seuil), choisissez Static (Statique).
9. Pour Whenever, EstimatedCharges c'est... , choisissez Greater.

10. Pour que... , définissez la valeur à laquelle vous souhaitez déclencher votre alerte. Par exemple, **`200`** USD.

    Les valeurs EstimatedChargesmétriques sont uniquement en dollars américains (USD) et la conversion des devises est assurée par Amazon Services LLC. Pour plus d'informations, voir Qu'est-ce que c'est AWS Billing ? .

    > **Note**
    >
    > Après avoir défini une valeur seuil, le graphique d'aperçu affiche vos frais estimés pour le mois en cours.

11. Choisissez Configuration supplémentaire et procédez comme suit :

    - Pour Datapoints to alarm (Points de données à alarmer), indiquez 1 out of 1 (1 sur 1).
    - Pour Missing data treatment (Traitement des données manquantes), sélectionnez Treat missing data as missing (Traiter les données manquantes comme manquantes).

12. Choisissez Suivant.
13. Sous Notification, assurez-vous que l'option En alarme est sélectionnée. Spécifiez une rubrique Amazon SNS à notifier lorsque l'alerte passe à l'état `ALARM`. La rubrique Amazon SNS peut inclure votre adresse e-mail afin que vous receviez un e-mail lorsque le montant de facturation dépasse le seuil que vous avez spécifié.

    Vous pouvez sélectionner une rubrique Amazon SNS existante, créer une nouvelle rubrique Amazon SNS ou utiliser l'ARN d'une rubrique pour notifier un autre compte. Si vous voulez que votre alarme envoie plusieurs notifications pour le même état de l'alarme ou pour des états de l'alarme différents, sélectionnez Add notification (Ajouter une notification).

14. Choisissez Suivant.
15. Sous Name and description (Nom et description), saisissez un nom pour votre alarme.

    - (Facultatif) Saisissez une description de votre alarme.

16. Choisissez Suivant.
17. Sous Preview and create (Prévisualiser et créer), assurez-vous que votre configuration est correcte, puis sélectionnez Create alarm (Créer l'alarme).

Étape 3 : vérifier l’état de l'alerte

Vérifiez l’état de l'alerte de facturation que vous venez de créer.

Vérifier l’état de l'alerte

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Si nécessaire, changez la région en USA Est (Virginie du Nord). Les données de métriques de facturation sont stockées dans cette région et reflètent des frais du monde entier.
3. Dans le panneau de navigation, cliquez sur Alarms (alertes).
4. Cochez la case en regard de l'alerte. Jusqu'à ce que l'abonnement soit confirmé, il est affiché comme « Confirmation en attente ». Une fois l'abonnement confirmé, actualisez la console pour afficher le statut mis à jour.

Étape 4 : modifier une alerte de facturation

Par exemple, vous souhaiterez peut-être augmenter le montant d'argent que vous dépensez AWS chaque mois de 200$ à 400$. Vous pouvez modifier votre alerte de facturation actuelle et augmenter le montant à dépasser pour déclencher l'alerte.

Modifier une alerte de facturation

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Si nécessaire, changez la région en USA Est (Virginie du Nord). Les données de métriques de facturation sont stockées dans cette région et reflètent des frais du monde entier.
3. Dans le panneau de navigation, cliquez sur Alarms (alertes).
4. Activez la case à cocher en regard de l'alerte, puis choisissez Actions, puis Modify (Modifier).
5. Pour chaque fois que le montant total de mes AWS frais pour le mois dépasse, spécifiez le nouveau montant qui doit être dépassé pour déclencher l'alarme et envoyez une notification par e-mail.
6. Choisissez Save Changes (Enregistrer les modifications).

Étape 5 : supprimer une alerte de facturation

Si vous n'avez plus besoin d'une alerte de facturation, vous pouvez la supprimer.

Pour supprimer une alerte de facturation

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Si nécessaire, changez la région en USA Est (Virginie du Nord). Les données de métriques de facturation sont stockées dans cette région et reflètent des frais du monde entier.
3. Dans le panneau de navigation, cliquez sur Alarms (alertes).
4. Cochez la case en regard de l'alerte, et choisissez Actions puis Supprimer.
5. Lorsque vous êtes invité à confirmer l'opération, choisissez Yes, Delete.

Scénario : publier des métriques sur CloudWatch

Dans ce scénario, vous utilisez le AWS Command Line Interface (AWS CLI) pour publier une seule métrique pour une application hypothétique nommée. GetStarted Si vous ne l'avez pas encore installé et configuré AWS CLI, consultez la section Getting Up with the AWS Command Line Interface dans le guide de AWS Command Line Interface l'utilisateur.

Tâches

- Étape 1 : définir la configuration des données
- Étape 2 : ajouter des métriques à CloudWatch
- Étape 3 : obtenir des statistiques à partir de CloudWatch
- Étape 4 : afficher des graphiques avec la console

Étape 1 : définir la configuration des données

Dans ce scénario, vous publiez des points de données qui suivent la latence des demandes pour l'application. Choisissez des noms explicites pour votre métrique et votre espace de noms. Pour cet exemple, nommez la métrique RequestLatencyet placez tous les points de données dans l'espace de GetStartednoms.

Vous publiez plusieurs points de données pour un total de trois heures de données de latence. Les données brutes comprennent 15 mesures de latence de demande sur plus de trois heures. Chaque mesure est en millisecondes :

- Première heure : 87, 51, 125, 235
- Deuxième heure : 121, 113, 189, 65, 89

- Troisième heure : 100, 47, 133, 98, 100, 328

Vous pouvez publier des données CloudWatch sous forme de points de données uniques ou d'un ensemble agrégé de points de données appelé ensemble de statistiques. Vous pouvez regrouper les métriques afin d'obtenir un niveau de détail aussi précis qu'une minute. Vous pouvez publier les points de données agrégés CloudWatch sous la forme d'un ensemble de statistiques avec quatre clés prédéfinies : `Sum`,`Minimum`,`Maximum`, et`SampleCount`.

Vous publiez les points de données de la première heure en tant que points de données uniques. Pour les données de la deuxième et de la troisième heures, vous regroupez les points de données et publierez un ensemble de statistiques pour chaque heure. Les valeurs de ces clés figurent dans le tableau suivant.

Heure	Données brutes	Somme	Minimum	Maximum	SampleCount
1	87				
1	51				
1	125				
1	235				
2	121, 113, 189, 65, 89	577	65	189	5
3	100, 47, 133, 98, 100, 328	806	47	328	6

Étape 2 : ajouter des métriques à CloudWatch

Après avoir défini la configuration de vos données, vous pouvez commencer à ajouter des données.

Pour publier des points de données sur CloudWatch

1. À l'invite de commandes, exécutez les put-metric-datacommandes suivantes pour ajouter des données pendant la première heure. Remplacez l'horodatage en exemple avec un horodatage de deux heures dans le passé, en heure UTC (Universal Coordinated Time).

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 87 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 51 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 125 --unit Milliseconds
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T20:30:00Z --value 235 --unit Milliseconds
```

2. Ajoutez des données pour la deuxième heure, à l'aide d'un horodatage défini une heure plus tard que la première heure.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--timestamp 2016-10-14T21:30:00Z --statistic-values
 Sum=577,Minimum=65,Maximum=189,SampleCount=5 --unit Milliseconds
```

3. Ajoutez des données pour la troisième heure, en omettant l'horodatage par défaut à l'heure actuelle.

```
aws cloudwatch put-metric-data --metric-name RequestLatency --namespace GetStarted
 \
--statistic-values Sum=806,Minimum=47,Maximum=328,SampleCount=6 --unit Milliseconds
```

Étape 3 : obtenir des statistiques à partir de CloudWatch

Maintenant que vous avez publié des métriques sur CloudWatch, vous pouvez récupérer des statistiques basées sur ces métriques à l'aide de la get-metric-statisticscommande suivante. Assurez-vous de spécifier `--start-time` et `--end-time` assez loin dans le passé afin de couvrir le premier horodatage que vous avez publié.

```
aws cloudwatch get-metric-statistics --namespace GetStarted --metric-name
 RequestLatency --statistics Average \
--start-time 2016-10-14T00:00:00Z --end-time 2016-10-15T00:00:00Z --period 60
```

Voici un exemple de sortie :

```
{
 "Datapoints": [],
 "Label": "Request:Latency"
}
```

Étape 4 : afficher des graphiques avec la console

Après avoir publié des métriques sur CloudWatch, vous pouvez utiliser la CloudWatch console pour afficher des graphiques statistiques.

Pour afficher des graphiques de vos statistiques sur la console

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de Navigation, choisissez Métriques.
3. Dans l'onglet Toutes les mesures, dans le champ de recherche, tapez RequestLatencyet appuyez sur Entrée.
4. Cochez la case correspondant à la RequestLatencymétrique. Un graphique des données de métrique s'affiche dans le volet supérieur.

Pour plus d'informations, voir Graphique des métriques.

Utilisation CloudWatch avec un AWS SDK

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

Documentation SDK	Exemples de code
AWS SDK for C++	AWS SDK for C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK for Go	AWS SDK for Go exemples de code
AWS SDK for Java	AWS SDK for Java exemples de code
AWS SDK for JavaScript	AWS SDK for JavaScript exemples de code
Kit AWS SDK pour Kotlin	Kit AWS SDK pour Kotlin exemples de code
AWS SDK for .NET	AWS SDK for .NET exemples de code
AWS SDK for PHP	AWS SDK for PHP exemples de code
AWS Tools for PowerShell	Outils pour des exemples PowerShell de code
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemples de code
AWS SDK for Ruby	AWS SDK for Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Pour des exemples spécifiques à CloudWatch, voirExemples de code pour CloudWatch l'utilisation des AWS SDK.

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.

Exemples de code pour CloudWatch l'utilisation des AWS SDK

Les exemples de code suivants montrent comment utiliser CloudWatch 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 et dans des exemples interservices.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir une tâche spécifique en appelant plusieurs fonctions au sein d'un même service.

Les Exemples de services croisés sont des exemples d'applications fonctionnant sur plusieurs Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Mise en route

Bonjour CloudWatch

Les exemples de code suivants montrent comment commencer à utiliser CloudWatch.

.NET

AWS SDK for .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 Amazon.CloudWatch;
using Amazon.CloudWatch.Model;
using Microsoft.Extensions.DependencyInjection;
```

```
using Microsoft.Extensions.Hosting;

namespace CloudWatchActions;

public static class HelloCloudWatch
{
    static async Task Main(string[] args)
    {
        // Use the AWS .NET Core Setup package to set up dependency injection for
 the Amazon CloudWatch service.
        // Use your AWS profile name, or leave it blank to use the default
 profile.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonCloudWatch>()
            ).Build();

        // Now the client is available for injection.
        var cloudWatchClient =
 host.Services.GetRequiredService<IAmazonCloudWatch>();

        // You can use await and any of the async methods to get a response.
        var metricNamespace = "AWS/Billing";
        var response = await cloudWatchClient.ListMetricsAsync(new
 ListMetricsRequest
        {
            Namespace = metricNamespace
        });
        Console.WriteLine($"Hello Amazon CloudWatch! Following are some metrics
 available in the {metricNamespace} namespace:");
        Console.WriteLine();
        foreach (var metric in response.Metrics.Take(5))
        {
            Console.WriteLine($"\tMetric: {metric.MetricName}");
            Console.WriteLine($"\tNamespace: {metric.Namespace}");
            Console.WriteLine($"\tDimensions: {string.Join(", ",
 metric.Dimensions.Select(m => $"{m.Name}:{m.Value}"))}");
            Console.WriteLine();
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section Référence des AWS SDK for .NET 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.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;

/**
 * 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 HelloService {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
```

```
            System.out.println(usage);
            System.exit(1);
        }

        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> System.out.println(" Retrieved metric is:
 " + metrics.metricName()));

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section Référence des AWS SDK for Java 2.x 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.

```
/**
Before running this Kotlin 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-kotlin/latest/developer-guide/setup.html
 */
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
           <namespace>
        Where:
           namespace - The namespace to filter against (for example, AWS/EC2).
    """

    if (args.size != 1) {
        println(usage)
        exitProcess(0)
    }

    val namespace = args[0]
    listAllMets(namespace)
}

suspend fun listAllMets(namespaceVal: String?) {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listMetricsPaginated(request)
            .transform { it.metrics?.forEach { obj -> emit(obj) } }
```

```
            .collect { obj ->
                println("Name is ${obj.metricName}")
                println("Namespace is ${obj.namespace}")
            }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section AWS SDK pour la référence de l'API Kotlin.

Exemples de code

- Actions relatives à CloudWatch l'utilisation des AWS SDK
  - Utilisation DeleteAlarms avec un AWS SDK ou une CLI
  - Utilisation DeleteAnomalyDetector avec un AWS SDK ou une CLI
  - Utilisation DeleteDashboards avec un AWS SDK ou une CLI
  - Utilisation DescribeAlarmHistory avec un AWS SDK ou une CLI
  - Utilisation DescribeAlarms avec un AWS SDK ou une CLI
  - Utilisation DescribeAlarmsForMetric avec un AWS SDK ou une CLI
  - Utilisation DescribeAnomalyDetectors avec un AWS SDK ou une CLI
  - Utilisation DisableAlarmActions avec un AWS SDK ou une CLI
  - Utilisation EnableAlarmActions avec un AWS SDK ou une CLI
  - Utilisation GetDashboard avec un AWS SDK ou une CLI
  - Utilisation GetMetricData avec un AWS SDK ou une CLI
  - Utilisation GetMetricStatistics avec un AWS SDK ou une CLI
  - Utilisation GetMetricWidgetImage avec un AWS SDK ou une CLI
  - Utilisation ListDashboards avec un AWS SDK ou une CLI
  - Utilisation ListMetrics avec un AWS SDK ou une CLI
  - Utilisation PutAnomalyDetector avec un AWS SDK ou une CLI
  - Utilisation PutDashboard avec un AWS SDK ou une CLI
  - Utilisation PutMetricAlarm avec un AWS SDK ou une CLI
  - Utilisation PutMetricData avec un AWS SDK ou une CLI
- Scénarios d' CloudWatch utilisation des AWS SDK

- Commencez à utiliser les CloudWatch alarmes à l'aide d'un AWS SDK
- Commencez à utiliser CloudWatch les métriques, les tableaux de bord et les alarmes à l'aide d'un SDK AWS
- Gérez les CloudWatch métriques et les alarmes à l'aide d'un AWS SDK
- Exemples multiservices d' CloudWatch utilisation des SDK AWS
  - Surveillez les performances d'Amazon DynamoDB à l'aide d'un SDK AWS

Actions relatives à CloudWatch l'utilisation des AWS SDK

Les exemples de code suivants montrent comment effectuer des CloudWatch actions individuelles avec AWS les SDK. Ces extraits appellent l' CloudWatch API et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

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

Exemples

- Utilisation DeleteAlarms avec un AWS SDK ou une CLI
- Utilisation DeleteAnomalyDetector avec un AWS SDK ou une CLI
- Utilisation DeleteDashboards avec un AWS SDK ou une CLI
- Utilisation DescribeAlarmHistory avec un AWS SDK ou une CLI
- Utilisation DescribeAlarms avec un AWS SDK ou une CLI
- Utilisation DescribeAlarmsForMetric avec un AWS SDK ou une CLI
- Utilisation DescribeAnomalyDetectors avec un AWS SDK ou une CLI
- Utilisation DisableAlarmActions avec un AWS SDK ou une CLI
- Utilisation EnableAlarmActions avec un AWS SDK ou une CLI
- Utilisation GetDashboard avec un AWS SDK ou une CLI
- Utilisation GetMetricData avec un AWS SDK ou une CLI
- Utilisation GetMetricStatistics avec un AWS SDK ou une CLI
- Utilisation GetMetricWidgetImage avec un AWS SDK ou une CLI
- Utilisation ListDashboards avec un AWS SDK ou une CLI
- Utilisation ListMetrics avec un AWS SDK ou une CLI

- Utilisation PutAnomalyDetector avec un AWS SDK ou une CLI
- Utilisation PutDashboard avec un AWS SDK ou une CLI
- Utilisation PutMetricAlarm avec un AWS SDK ou une CLI
- Utilisation PutMetricData avec un AWS SDK ou une CLI

Utilisation **`DeleteAlarms`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage des alarmes
- Démarrage avec les métriques, tableaux de bord et alertes
- Gérer les mesures et les alertes

.NET

AWS SDK for .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.

```
/// <summary>
/// Delete a list of alarms from CloudWatch.
/// </summary>
/// <param name="alarmNames">A list of names of alarms to delete.</param>
/// <returns>True if successful.</returns>
public async Task<bool> DeleteAlarms(List<string> alarmNames)
{
    var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
        new DeleteAlarmsRequest()
        {
            AlarmNames = alarmNames
```

```
            });

        return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà la section Référence des AWS SDK for .NET API.

C++

SDK pour 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/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DeleteAlarmsRequest.h>
#include <iostream>
```

Supprimer l'alerte.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DeleteAlarmsRequest request;
        request.AddAlarmNames(alarm_name);

        auto outcome = cw.DeleteAlarms(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to delete CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
```

```
            std::cout << "Successfully deleted CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour supprimer une alarme

L'exemple suivant utilise la `delete-alarms` commande pour supprimer l' CloudWatch alarme Amazon nommée « myalarm » :

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

Sortie :

```
This command returns to the prompt if successful.
```

- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà 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.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;
```

```
/**
 * 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 DeleteAlarm {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to delete (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_2;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        deleteCWAlarm(cw, alarmName);
        cw.close();
    }

    public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
        try {
            DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                    .alarmNames(alarmName)
                    .build();

            cw.deleteAlarms(request);
```

```
            System.out.printf("Successfully deleted alarm %s", alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà 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.

Importez le kit SDK et les modules client et appelez l'API.

```
import { DeleteAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DeleteAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};
```

```
export default run();
```

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà la section Référence des AWS SDK for 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.

Importez le kit SDK et les modules client et appelez l'API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmNames: ["Web_Server_CPU_Utilization"],
};

cw.deleteAlarms(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
```

```
    console.log("Success", data);
  }
});
```

- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà la section Référence des AWS SDK for 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 deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
 metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
 metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

- Pour plus de détails sur l'API, consultez DeleteAlarmsle 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.
    lo_cwt->deletealarms(
      it_alarmnames = it_alarm_names
    ).
    MESSAGE 'Alarms deleted.' TYPE 'I'.
  CATCH /aws1/cx_cwtresourcenotfound .
    MESSAGE 'Resource being accessed is not found.' TYPE 'E'.
ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous DeleteAlarmsà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteAnomalyDetector`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Delete a single metric anomaly detector.
    /// </summary>
    /// <param name="anomalyDetector">The anomaly detector to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var deleteAnomalyDetectorResponse = await
_amazonCloudWatch.DeleteAnomalyDetectorAsync(
            new DeleteAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteAnomalyDetectorà la section Référence des AWS SDK for .NET 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.

```
   public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.deleteAnomalyDetector(request);
           System.out.println("Successfully deleted the Anomaly Detector.");

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       } catch (IOException e) {
```

```
            e.printStackTrace();
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteAnomalyDetectorà la section Référence des AWS SDK for Java 2.x 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 deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteAnomalyDetectorà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DeleteDashboards`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
_amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
```

```
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteDashboardsà la section Référence des AWS SDK for .NET 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.

```
   public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
       try {
           DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                   .dashboardNames(dashboardName)
                   .build();
           cw.deleteDashboards(dashboardsRequest);
           System.out.println(dashboardName + " was successfully deleted.");

       } catch (CloudWatchException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous DeleteDashboardsà la section Référence des AWS SDK for Java 2.x 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 deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteDashboardsà la section AWS SDK pour la référence de l'API Kotlin.

PowerShell

Outils pour PowerShell

Exemple 1 : Supprime le tableau de bord spécifié, en le promouvant pour confirmation avant de continuer. Pour contourner la confirmation, ajoutez le commutateur -Force à la commande.

```
Remove-CWDashboard -DashboardName Dashboard1
```

- Pour plus de détails sur l'API, reportez-vous DeleteDashboardsà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeAlarmHistory`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
   /// <summary>
   /// Describe the history of an alarm for a number of days in the past.
   /// </summary>
   /// <param name="alarmName">The name of the alarm.</param>
   /// <param name="historyDays">The number of days in the past.</param>
   /// <returns>The list of alarm history data.</returns>
   public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
   {
       List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
       var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
           new DescribeAlarmHistoryRequest()
           {
               AlarmName = alarmName,
               EndDateUtc = DateTime.UtcNow,
               HistoryItemType = HistoryItemType.StateUpdate,
               StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
           });

       await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
```

```
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmHistoryà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour récupérer l'historique d'une alarme

L'exemple suivant utilise la `describe-alarm-history` commande pour récupérer l'historique de l'CloudWatch alarme Amazon nommée « myalarm » :

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-type
 StateUpdate
```

Sortie :

```
{
    "AlarmHistoryItems": [
        {
            "Timestamp": "2014-04-09T18:59:06.442Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"ALARM\",\"stateReason\":\"testing purposes\"},\"newState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were not
 greater than the threshold (70.0). The most recent datapoints: [38.958,
 40.292].\",\"stateReasonData\":{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}}",
            "HistorySummary": "Alarm updated from ALARM to OK"
        },
        {
```

```
            "Timestamp": "2014-04-09T18:59:05.805Z",
            "HistoryItemType": "StateUpdate",
            "AlarmName": "myalarm",
            "HistoryData": "{\"version\":\"1.0\",\"oldState\":{\"stateValue
\":\"OK\",\"stateReason\":\"Threshold Crossed: 2 datapoints were
 not greater than the threshold (70.0). The most recent datapoints:
 [38.839999999999996, 39.714].\",\"stateReasonData\":{\"version\":
\"1.0\",\"queryDate\":\"2014-03-11T22:45:41.569+0000\",\"startDate\":
\"2014-03-11T22:30:00.000+0000\",\"statistic\":\"Average\",\"period\":300,
\"recentDatapoints\":[38.839999999999996,39.714],\"threshold\":70.0}},\"newState
\":{\"stateValue\":\"ALARM\",\"stateReason\":\"testing purposes\"}}",
            "HistorySummary": "Alarm updated from OK to ALARM"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmHistoryà 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.

```
   public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String alarmName = rootNode.findValue("exampleAlarmName").asText();

           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();
```

```
            DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                    .startDate(start)
                    .endDate(endDate)
                    .alarmName(alarmName)
                    .historyItemType(HistoryItemType.ACTION)
                    .build();

            DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
            List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
            if (historyItems.isEmpty()) {
                System.out.println("No alarm history data found for " + alarmName
+ ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmHistoryà la section Référence des AWS SDK for Java 2.x 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 getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmHistoryà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeAlarms`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage des alarmes
- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
   /// <summary>
   /// Describe the current alarms, optionally filtered by state.
   /// </summary>
   /// <param name="stateValue">Optional filter for alarm state.</param>
   /// <returns>The list of alarm data.</returns>
   public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
null)
   {
       List<MetricAlarm> alarms = new List<MetricAlarm>();
       var paginatedDescribeAlarms =
_amazonCloudWatch.Paginators.DescribeAlarms(
           new DescribeAlarmsRequest()
           {
               StateValue = stateValue
           });

       await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
       {
           alarms.Add(data);
```

```
        }
        return alarms;
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour répertorier les informations relatives à une alarme

L'exemple suivant fait appel à la commande `describe-alarms` pour fournir des informations sur l'alarme nommée « myalarm » :

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

Sortie :

```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:123456789012:alarm:myalarm",
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2012-12-27T00:49:54.032Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:123456789012:myHighCpuAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
```

```
            "AlarmName": "myalarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsà 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.

```
    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();
```

```
            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
 alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsà la section Référence des AWS SDK for Java 2.x 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 describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
```

```
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsà la section AWS SDK pour la référence de l'API Kotlin.

Ruby

Kit SDK pour Ruby

> **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.

```
require "aws-sdk-cloudwatch"

# Lists the names of available Amazon CloudWatch alarms.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   list_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def list_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms
  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts alarm.alarm_name
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsà la section Référence des AWS SDK for Ruby API.

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_cwt->describealarms(                " oo_result is
 returned for testing purposes. "
          it_alarmnames = it_alarm_names
        ).
        MESSAGE 'Alarms retrieved.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeAlarmsForMetric`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes
- Gérer les mesures et les alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà la section Référence des AWS SDK for .NET API.

C++

SDK pour 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/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DescribeAlarmsRequest.h>
#include <aws/monitoring/model/DescribeAlarmsResult.h>
#include <iomanip>
#include <iostream>
```

Décrire les alertes.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::DescribeAlarmsRequest request;
        request.SetMaxRecords(1);

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.DescribeAlarms(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to describe CloudWatch alarms:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left <<
                    std::setw(32) << "Name" <<
```

```
                    std::setw(64) << "Arn" <<
                    std::setw(64) << "Description" <<
                    std::setw(20) << "LastUpdated" <<
                    std::endl;
                header = true;
            }

            const auto &alarms = outcome.GetResult().GetMetricAlarms();
            for (const auto &alarm : alarms)
            {
                std::cout << std::left <<
                    std::setw(32) << alarm.GetAlarmName() <<
                    std::setw(64) << alarm.GetAlarmArn() <<
                    std::setw(64) << alarm.GetAlarmDescription() <<
                    std::setw(20) <<
                    alarm.GetAlarmConfigurationUpdatedTimestamp().ToGmtString(
                        SIMPLE_DATE_FORMAT_STR) <<
                    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 DescribeAlarmsForMetricà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour afficher des informations relatives aux alarmes associées à une métrique

L'exemple suivant fait appel à la commande `describe-alarms-for-metric` pour afficher des informations sur les alarmes associées à la métrique Amazon EC2 CPUUtilization et à l'instance portant l'ID i-0c986c72. :

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

Sortie :


```
{
    "MetricAlarms": [
        {
            "EvaluationPeriods": 10,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm2",
            "StateUpdatedTimestamp": "2013-10-30T03:03:51.479Z",
            "AlarmConfigurationUpdatedTimestamp": "2013-10-30T03:03:50.865Z",
            "ComparisonOperator": "GreaterThanOrEqualToThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:NotifyMe"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2013-10-30T03:03:51.479+0000\",\"startDate\":\"2013-10-30T02:08:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":
[40.698,39.612,42.432,39.796,38.816,42.28,42.854,40.088,40.760000000000005,41.316],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm2",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 10 datapoints were not
 greater than or equal to the threshold (70.0). The most recent datapoints:
 [40.760000000000005, 41.316].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": true,
            "MetricName": "CPUUtilization"
        },
        {
            "EvaluationPeriods": 2,
            "AlarmArn": "arn:aws:cloudwatch:us-
east-1:111122223333:alarm:myHighCpuAlarm",
```

```
            "StateUpdatedTimestamp": "2014-04-09T18:59:06.442Z",
            "AlarmConfigurationUpdatedTimestamp": "2014-04-09T22:26:05.958Z",
            "ComparisonOperator": "GreaterThanThreshold",
            "AlarmActions": [
                "arn:aws:sns:us-east-1:111122223333:HighCPUAlarm"
            ],
            "Namespace": "AWS/EC2",
            "AlarmDescription": "CPU usage exceeds 70 percent",
            "StateReasonData": "{\"version\":\"1.0\",\"queryDate\":
\"2014-04-09T18:59:06.419+0000\",\"startDate\":\"2014-04-09T18:44:00.000+0000\",
\"statistic\":\"Average\",\"period\":300,\"recentDatapoints\":[38.958,40.292],
\"threshold\":70.0}",
            "Period": 300,
            "StateValue": "OK",
            "Threshold": 70.0,
            "AlarmName": "myHighCpuAlarm",
            "Dimensions": [
                {
                    "Name": "InstanceId",
                    "Value": "i-0c986c72"
                }
            ],
            "Statistic": "Average",
            "StateReason": "Threshold Crossed: 2 datapoints were not greater than
 the threshold (70.0). The most recent datapoints: [38.958, 40.292].",
            "InsufficientDataActions": [],
            "OKActions": [],
            "ActionsEnabled": false,
            "MetricName": "CPUUtilization"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà 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.

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
               DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
               hasAlarm = response.hasMetricAlarms();
               retries--;
               Thread.sleep(20000);
               System.out.println(".");
           }
           if (!hasAlarm)
               System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
```

```
            else
                System.out.println("Alarm state found for " + customMetricName +
 ".");

        } catch (CloudWatchException | IOException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà 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.

Importez le kit SDK et les modules client et appelez l'API.

```
import { DescribeAlarmsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DescribeAlarmsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};
```

```
export default run();
```

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà la section Référence des AWS SDK for 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 CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.describeAlarms({ StateValue: "INSUFFICIENT_DATA" }, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    // List the names of all current alarms in the console
    data.MetricAlarms.forEach(function (item, index, array) {
      console.log(item.AlarmName);
    });
  }
});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà la section Référence des AWS SDK for 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 checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
            println(".")
        }
```

```
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
 metric_name)
```

```
        return alarm_iter
```

- Pour plus de détails sur l'API, consultez DescribeAlarmsForMetricle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **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.

```
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @example
#   describe_metric_alarms(Aws::CloudWatch::Client.new(region: 'us-east-1'))
def describe_metric_alarms(cloudwatch_client)
  response = cloudwatch_client.describe_alarms

  if response.metric_alarms.count.positive?
    response.metric_alarms.each do |alarm|
      puts "-" * 16
      puts "Name:           " + alarm.alarm_name
      puts "State value:    " + alarm.state_value
      puts "State reason:   " + alarm.state_reason
      puts "Metric:         " + alarm.metric_name
      puts "Namespace:      " + alarm.namespace
      puts "Statistic:      " + alarm.statistic
      puts "Period:         " + alarm.period.to_s
      puts "Unit:           " + alarm.unit.to_s
      puts "Eval. periods:  " + alarm.evaluation_periods.to_s
      puts "Threshold:      " + alarm.threshold.to_s
      puts "Comp. operator: " + alarm.comparison_operator
```

```
      if alarm.key?(:ok_actions) && alarm.ok_actions.count.positive?
        puts "OK actions:"
        alarm.ok_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:alarm_actions) && alarm.alarm_actions.count.positive?
        puts "Alarm actions:"
        alarm.alarm_actions.each do |a|
          puts "  " + a
        end
      end

      if alarm.key?(:insufficient_data_actions) &&
          alarm.insufficient_data_actions.count.positive?
        puts "Insufficient data actions:"
        alarm.insufficient_data_actions.each do |a|
          puts "  " + a
        end
      end

      puts "Dimensions:"
      if alarm.key?(:dimensions) && alarm.dimensions.count.positive?
        alarm.dimensions.each do |d|
          puts "  Name: " + d.name + ", Value: " + d.value
        end
      else
        puts "  None for this alarm."
      end
    end
  else
    puts "No alarms found."
  end
rescue StandardError => e
  puts "Error getting information about alarms: #{e.message}"
end

# Example usage:
def run_me
  region = ""

  # Print usage information and then stop.
  if ARGV[0] == "--help" || ARGV[0] == "-h"
```

```
    puts "Usage:   ruby cw-ruby-example-show-alarms.rb REGION"
    puts "Example: ruby cw-ruby-example-show-alarms.rb us-east-1"
    exit 1
  # If no values are specified at the command prompt, use these default values.
  elsif ARGV.count.zero?
    region = "us-east-1"
  # Otherwise, use the values as specified at the command prompt.
  else
    region = ARGV[0]
  end

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
  puts "Available alarms:"
  describe_metric_alarms(cloudwatch_client)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous DescribeAlarmsForMetricà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeAnomalyDetectors`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Describe anomaly detectors for a metric and namespace.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The metric of the anomaly detectors.</param>
    /// <returns>The list of detectors.</returns>
    public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
    {
        List<AnomalyDetector> detectors = new List<AnomalyDetector>();
        var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
            new DescribeAnomalyDetectorsRequest()
            {
                MetricName = metricName,
                Namespace = metricNamespace
            });

        await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
        {
            detectors.Add(data);
        }

        return detectors;
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAnomalyDetectorsà la section Référence des AWS SDK for .NET 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.

```
   public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                   .maxResults(10)
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
           List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
           for (AnomalyDetector detector : anomalyDetectorList) {
               System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
               System.out.println("State: " + detector.stateValue());
           }

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
```

```
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeAnomalyDetectorsà la section Référence des AWS SDK for Java 2.x 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 describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeAnomalyDetectorsà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DisableAlarmActions`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage des alarmes
- Gérer les mesures et les alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
```

```
                AlarmNames = alarmNames
            });

        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà la section Référence des AWS SDK for .NET API.

C++

SDK pour 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/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/DisableAlarmActionsRequest.h>
#include <iostream>
```

Désactiver des actions d'alerte.

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::DisableAlarmActionsRequest
 disableAlarmActionsRequest;
        disableAlarmActionsRequest.AddAlarmNames(alarm_name);

        auto disableAlarmActionsOutcome =
 cw.DisableAlarmActions(disableAlarmActionsRequest);
        if (!disableAlarmActionsOutcome.IsSuccess())
        {
            std::cout << "Failed to disable actions for alarm " << alarm_name <<
```

```
                ": " << disableAlarmActionsOutcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully disabled actions for alarm " <<
                alarm_name << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour désactiver les actions d'une alarme

L'exemple suivant fait appel à la commande `disable-alarm-actions` pour désactiver toutes les actions de l'alarme nommée myalarm :

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

Cette commande revient à l'invite en cas de succès.

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà 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.CloudWatchClient;
```

```
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.DisableAlarmActionsRequest;

/**
 * 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 DisableAlarmActions {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to disable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarmName = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        disableActions(cw, alarmName);
        cw.close();
    }

    public static void disableActions(CloudWatchClient cw, String alarmName) {
        try {
            DisableAlarmActionsRequest request =
 DisableAlarmActionsRequest.builder()
                    .alarmNames(alarmName)
```

```
                    .build();

            cw.disableAlarmActions(request);
            System.out.printf("Successfully disabled actions on alarm %s",
 alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà 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.

Importez le kit SDK et les modules client et appelez l'API.

```
import { DisableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new DisableAlarmActionsCommand({
    AlarmNames: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
```

```
    console.error(err);
  }
};

export default run();
```

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà la section Référence des AWS SDK for 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.

Importez le kit SDK et les modules client et appelez l'API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

cw.disableAlarmActions(
  { AlarmNames: ["Web_Server_CPU_Utilization"] },
  function (err, data) {
    if (err) {
```

```
      console.log("Error", err);
    } else {
      console.log("Success", data);
    }
  }
);
```

- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà la section Référence des AWS SDK for 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 disableActions(alarmName: String) {

    val request = DisableAlarmActionsRequest {
        alarmNames = listOf(alarmName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.disableAlarmActions(request)
        println("Successfully disabled actions on alarm $alarmName")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
```

```
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Pour plus de détails sur l'API, consultez DisableAlarmActionsle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **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.

```
# Disables an alarm in Amazon CloudWatch.
#
# Prerequisites.
#
# - The alarm to disable.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm to disable.
# @return [Boolean] true if the alarm was disabled; otherwise, false.
# @example
#   exit 1 unless alarm_actions_disabled?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket'
#   )
def alarm_actions_disabled?(cloudwatch_client, alarm_name)
  cloudwatch_client.disable_alarm_actions(alarm_names: [alarm_name])
  return true
```

```
rescue StandardError => e
  puts "Error disabling alarm actions: #{e.message}"
  return false
end

# Example usage:
def run_me
  alarm_name = "ObjectsInBucket"
  alarm_description = "Objects exist in this bucket for more than 1 day."
  metric_name = "NumberOfObjects"
  # Notify this Amazon Simple Notification Service (Amazon SNS) topic when
  # the alarm transitions to the ALARM state.
  alarm_actions = ["arn:aws:sns:us-
east-1:111111111111:Default_CloudWatch_Alarms_Topic"]
  namespace = "AWS/S3"
  statistic = "Average"
  dimensions = [
    {
      name: "BucketName",
      value: "doc-example-bucket"
    },
    {
      name: "StorageType",
      value: "AllStorageTypes"
    }
  ]
  period = 86_400 # Daily (24 hours * 60 minutes * 60 seconds = 86400 seconds).
  unit = "Count"
  evaluation_periods = 1 # More than one day.
  threshold = 1 # One object.
  comparison_operator = "GreaterThanThreshold" # More than one object.
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  if alarm_created_or_updated?(
    cloudwatch_client,
    alarm_name,
    alarm_description,
    metric_name,
    alarm_actions,
    namespace,
    statistic,
```

```
    dimensions,
    period,
    unit,
    evaluation_periods,
    threshold,
    comparison_operator
  )
    puts "Alarm '#{alarm_name}' created or updated."
  else
    puts "Could not create or update alarm '#{alarm_name}'."
  end

  if alarm_actions_disabled?(cloudwatch_client, alarm_name)
    puts "Alarm '#{alarm_name}' disabled."
  else
    puts "Could not disable alarm '#{alarm_name}'."
  end
end

run_me if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà la section Référence des AWS SDK for Ruby API.

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.

```
    "Disables actions on the specified alarm. "
    TRY.
        lo_cwt->disablealarmactions(
          it_alarmnames = it_alarm_names
        ).
```

```
        MESSAGE 'Alarm actions disabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous DisableAlarmActionsà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`EnableAlarmActions`** avec un AWS SDK ou une CLI

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

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 :

- Gérer les mesures et les alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
```

```
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà la section Référence des AWS SDK for .NET API.

C++

SDK pour 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/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/EnableAlarmActionsRequest.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Activer des actions d'alarme.

```
    Aws::CloudWatch::CloudWatchClient cw;
    Aws::CloudWatch::Model::PutMetricAlarmRequest request;
    request.SetAlarmName(alarm_name);
```

```
    request.SetComparisonOperator(
        Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
    request.SetEvaluationPeriods(1);
    request.SetMetricName("CPUUtilization");
    request.SetNamespace("AWS/EC2");
    request.SetPeriod(60);
    request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
    request.SetThreshold(70.0);
    request.SetActionsEnabled(false);
    request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
    request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);
    request.AddAlarmActions(actionArn);

    Aws::CloudWatch::Model::Dimension dimension;
    dimension.SetName("InstanceId");
    dimension.SetValue(instanceId);
    request.AddDimensions(dimension);

    auto outcome = cw.PutMetricAlarm(request);
    if (!outcome.IsSuccess())
    {
        std::cout << "Failed to create CloudWatch alarm:" <<
            outcome.GetError().GetMessage() << std::endl;
        return;
    }

    Aws::CloudWatch::Model::EnableAlarmActionsRequest enable_request;
    enable_request.AddAlarmNames(alarm_name);

    auto enable_outcome = cw.EnableAlarmActions(enable_request);
    if (!enable_outcome.IsSuccess())
    {
        std::cout << "Failed to enable alarm actions:" <<
            enable_outcome.GetError().GetMessage() << std::endl;
        return;
    }

    std::cout << "Successfully created alarm " << alarm_name <<
        " and enabled actions on it." << std::endl;
```

- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour activer toutes les actions d'une alarme

L'exemple suivant fait appel à la commande `enable-alarm-actions` pour activer toutes les actions de l'alarme nommée myalarm :

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

Cette commande revient à l'invite en cas de succès.

- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà 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.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import
 software.amazon.awssdk.services.cloudwatch.model.EnableAlarmActionsRequest;

/**
 * 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 EnableAlarmActions {
```

```
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <alarmName>

                Where:
                  alarmName - An alarm name to enable (for example, MyAlarm).
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String alarm = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        enableActions(cw, alarm);
        cw.close();
    }

    public static void enableActions(CloudWatchClient cw, String alarm) {
        try {
            EnableAlarmActionsRequest request =
 EnableAlarmActionsRequest.builder()
                    .alarmNames(alarm)
                    .build();

            cw.enableAlarmActions(request);
            System.out.printf("Successfully enabled actions on alarm %s", alarm);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà 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.

Importez le kit SDK et les modules client et appelez l'API.

```
import { EnableAlarmActionsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  const command = new EnableAlarmActionsCommand({
    AlarmNames: [process.env.CLOUDWATCH_ALARM_NAME], // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà la section Référence des AWS SDK for 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.

Importez le kit SDK et les modules client et appelez l'API.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: true,
  AlarmActions: ["ACTION_ARN"],
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
```

```
};

cw.putMetricAlarm(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Alarm action added", data);
    var paramsEnableAlarmAction = {
      AlarmNames: [params.AlarmName],
    };
    cw.enableAlarmActions(paramsEnableAlarmAction, function (err, data) {
      if (err) {
        console.log("Error", err);
      } else {
        console.log("Alarm action enabled", data);
      }
    });
  }
});
```

- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà la section Référence des AWS SDK for 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 enableActions(alarm: String) {

    val request = EnableAlarmActionsRequest {
        alarmNames = listOf(alarm)
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.enableAlarmActions(request)
        println("Successfully enabled actions on alarm $alarm")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def enable_alarm_actions(self, alarm_name, enable):
        """
        Enables or disables actions on the specified alarm. Alarm actions can be
        used to send notifications or automate responses when an alarm enters a
        particular state.

        :param alarm_name: The name of the alarm.
        :param enable: When True, actions are enabled for the alarm. Otherwise,
 they
                       disabled.
```

```
        """
        try:
            alarm = self.cloudwatch_resource.Alarm(alarm_name)
            if enable:
                alarm.enable_actions()
            else:
                alarm.disable_actions()
            logger.info(
                "%s actions for alarm %s.",
                "Enabled" if enable else "Disabled",
                alarm_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't %s actions alarm %s.",
                "enable" if enable else "disable",
                alarm_name,
            )
            raise
```

- Pour plus de détails sur l'API, consultez EnableAlarmActionsle 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.

```
"Enable actions on the specified alarm."
TRY.
    lo_cwt->enablealarmactions(
      it_alarmnames = it_alarm_names
    ).
```

```
        MESSAGE 'Alarm actions enabled.' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_exception).
        DATA(lv_error) = |"{ lo_exception->av_err_code }" - { lo_exception-
>av_err_msg }|.
        MESSAGE lv_error TYPE 'E'.
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous EnableAlarmActionsà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetDashboard`** avec un AWS SDK ou une CLI

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

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
```

```
            });

        return dashboardResponse.DashboardBody;
    }
```

- Pour plus de détails sur l'API, reportez-vous GetDashboardà la section Référence des AWS SDK for .NET API.

PowerShell

Outils pour PowerShell

Exemple 1 : renvoie l'arn dans le corps du tableau de bord spécifié.

```
Get-CWDashboard -DashboardName Dashboard1
```

Sortie :

```
DashboardArn                                              DashboardBody
------------                                              -------------
arn:aws:cloudwatch::123456789012:dashboard/Dashboard1 {...
```

- Pour plus de détails sur l'API, reportez-vous GetDashboardà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetMetricData`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
```

```
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }
```

- Pour plus de détails sur l'API, reportez-vous GetMetricDataà la section Référence des AWS SDK for .NET 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.

```
    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
```

```
            // Set the date.
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }
```

```
        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetMetricDataà la section Référence des AWS SDK for Java 2.x 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 getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
```

```
    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetMetricDataà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetMetricStatistics`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes
- Gérer les mesures et les alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();
```

```
        return billingStatistics;
    }

    /// <summary>
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }
```

- Pour plus de détails sur l'API, reportez-vous GetMetricStatisticsà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour obtenir l’utilisation du processeur par instance EC2

L'exemple suivant fait appel à la commande `get-metric-statistics` pour obtenir l'utilisation du processeur pour une instance EC2 portant l'ID i-abcdef.

```
aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time
 2014-04-08T23:18:00Z --end-time 2014-04-09T23:18:00Z --period 3600 --namespace
 AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-abcdef
```

Sortie :

```
{
    "Datapoints": [
        {
            "Timestamp": "2014-04-09T11:18:00Z",
            "Maximum": 44.79,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T20:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T19:18:00Z",
            "Maximum": 50.85,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T09:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T03:18:00Z",
            "Maximum": 76.84,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T21:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T14:18:00Z",
```

```
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T08:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T16:18:00Z",
            "Maximum": 45.55,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T06:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T13:18:00Z",
            "Maximum": 45.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T05:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T18:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T17:18:00Z",
            "Maximum": 52.08,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T07:18:00Z",
            "Maximum": 47.92,
            "Unit": "Percent"
        },
        {
```

```
            "Timestamp": "2014-04-09T02:18:00Z",
            "Maximum": 51.23,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T12:18:00Z",
            "Maximum": 47.67,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-08T23:18:00Z",
            "Maximum": 46.88,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T10:18:00Z",
            "Maximum": 51.91,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T04:18:00Z",
            "Maximum": 47.13,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T15:18:00Z",
            "Maximum": 48.96,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T00:18:00Z",
            "Maximum": 48.16,
            "Unit": "Percent"
        },
        {
            "Timestamp": "2014-04-09T01:18:00Z",
            "Maximum": 49.18,
            "Unit": "Percent"
        }
    ],
    "Label": "CPUUtilization"
}
```

Spécification de plusieurs dimensions

L'exemple suivant illustre comment spécifier plusieurs dimensions. Chaque dimension est spécifiée sous la forme d'une paire nom/valeur, avec une virgule entre le nom et la valeur. Les différentes dimensions sont séparées par une espace. Si une métrique comprend plusieurs dimensions, vous devez préciser une valeur pour chaque dimension définie.

Pour d'autres exemples d'utilisation de cette `get-metric-statistics` commande, consultez la section Obtenir des statistiques pour une métrique dans le manuel Amazon CloudWatch Developer Guide.

```
aws cloudwatch get-metric-statistics --metric-name Buffers --
namespace MyNameSpace --dimensions Name=InstanceID,Value=i-abcdef
 Name=InstanceType,Value=m1.small --start-time 2016-10-15T04:00:00Z --end-time
 2016-10-19T07:00:00Z --statistics Average --period 60
```

- Pour plus de détails sur l'API, reportez-vous GetMetricStatisticsà 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.

```
   public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
           String metricOption, String date, Dimension myDimension) {
       try {
           Instant start = Instant.parse(date);
           Instant endDate = Instant.now();

           GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                   .endTime(endDate)
                   .startTime(start)
```

```
                    .dimensions(myDimension)
                    .metricName(metVal)
                    .namespace(nameSpace)
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetMetricStatisticsà la section Référence des AWS SDK for Java 2.x 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 getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetMetricStatisticsà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def get_metric_statistics(self, namespace, name, start, end, period,
 stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
 grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
 period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
 must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
 value
                           or maximum value.
        :return: The retrieved statistics for the metric.
```

```
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats
```

- Pour plus de détails sur l'API, consultez GetMetricStatisticsle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`GetMetricWidgetImage`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };

        var metricImageWidgetString =
JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
```

```
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }
```

- Pour plus de détails sur l'API, reportez-vous GetMetricWidgetImageà la section Référence des AWS SDK for .NET 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.

```
   public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
       System.out.println("Getting Image data for custom metric.");
       try {
           String myJSON = "{\n" +
                   "  \"title\": \"Example Metric Graph\",\n" +
                   "  \"view\": \"timeSeries\",\n" +
                   "  \"stacked \": false,\n" +
```

```
                    "  \"period\": 10,\n" +
                    "  \"width\": 1400,\n" +
                    "  \"height\": 600,\n" +
                    "  \"metrics\": [\n" +
                    "    [\n" +
                    "      \"AWS/Billing\",\n" +
                    "      \"EstimatedCharges\",\n" +
                    "      \"Currency\",\n" +
                    "      \"USD\"\n" +
                    "    ]\n" +
                    "  ]\n" +
                    "}";

            GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                    .metricWidget(myJSON)
                    .build();

            GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous GetMetricWidgetImageà la section Référence des AWS SDK for Java 2.x 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 getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}
```

- Pour plus de détails sur l'API, reportez-vous GetMetricWidgetImageà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListDashboards`** avec un AWS SDK ou une CLI

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

.NET

AWS SDK for .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.

```
/// <summary>
/// Get a list of dashboards.
/// </summary>
/// <returns>A list of DashboardEntry objects.</returns>
public async Task<List<DashboardEntry>> ListDashboards()
{
    var results = new List<DashboardEntry>();
    var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
        new ListDashboardsRequest());
    // Get the entire list using the paginator.
    await foreach (var data in paginateDashboards.DashboardEntries)
    {
        results.Add(data);
    }

    return results;
}
```

- Pour plus de détails sur l'API, reportez-vous ListDashboardsà la section Référence des AWS SDK for .NET 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.

```
    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous ListDashboardsà la section Référence des AWS SDK for Java 2.x 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 listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListDashboardsà la section AWS SDK pour la référence de l'API Kotlin.

PowerShell

Outils pour PowerShell

Exemple 1 : renvoie la collection de tableaux de bord pour votre compte.

```
Get-CWDashboardList
```

Sortie :

```
DashboardArn DashboardName LastModified        Size
------------ ------------- ------------        ----
arn:...      Dashboard1    7/6/2017 8:14:15 PM 252
```

Exemple 2 : renvoie la collection de tableaux de bord de votre compte dont le nom commence par le préfixe « dev ».

```
Get-CWDashboardList -DashboardNamePrefix dev
```

- Pour plus de détails sur l'API, reportez-vous ListDashboardsà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListMetrics`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes
- Gérer les mesures et les alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// List metrics available, optionally within a namespace.
    /// </summary>
    /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
```

```
    /// <param name="filter">Optional dimension filter.</param>
    /// <param name="metricName">Optional metric name filter.</param>
    /// <returns>The list of metrics.</returns>
    public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }

        return results;
    }
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section Référence des AWS SDK for .NET API.

C++

SDK pour 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/monitoring/CloudWatchClient.h>
```

```
#include <aws/monitoring/model/ListMetricsRequest.h>
#include <aws/monitoring/model/ListMetricsResult.h>
#include <iomanip>
#include <iostream>
```

Répertoriez les métriques.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::ListMetricsRequest request;

        if (argc > 1)
        {
            request.SetMetricName(argv[1]);
        }

        if (argc > 2)
        {
            request.SetNamespace(argv[2]);
        }

        bool done = false;
        bool header = false;
        while (!done)
        {
            auto outcome = cw.ListMetrics(request);
            if (!outcome.IsSuccess())
            {
                std::cout << "Failed to list CloudWatch metrics:" <<
                    outcome.GetError().GetMessage() << std::endl;
                break;
            }

            if (!header)
            {
                std::cout << std::left << std::setw(48) << "MetricName" <<
                    std::setw(32) << "Namespace" << "DimensionNameValuePairs" <<
                    std::endl;
                header = true;
            }

            const auto &metrics = outcome.GetResult().GetMetrics();
            for (const auto &metric : metrics)
```

```
            {
                std::cout << std::left << std::setw(48) <<
                    metric.GetMetricName() << std::setw(32) <<
                    metric.GetNamespace();
                const auto &dimensions = metric.GetDimensions();
                for (auto iter = dimensions.cbegin();
                    iter != dimensions.cend(); ++iter)
                {
                    const auto &dimkv = *iter;
                    std::cout << dimkv.GetName() << " = " << dimkv.GetValue();
                    if (iter + 1 != dimensions.cend())
                    {
                        std::cout << ", ";
                    }
                }
                std::cout << 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 ListMetricsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour répertorier les métriques pour Amazon SNS

L'exemple `list-metrics` suivant affiche les métriques pour Amazon SNS.

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

Sortie :

```
{
    "Metrics": [
```

```
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "PublishSize"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
```

```
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà 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.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;

/**
 * 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 ListMetrics {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                  <namespace>\s

                Where:
                  namespace - The namespace to filter against (for example, AWS/
EC2).\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        String namespace = args[0];
        Region region = Region.US_EAST_1;
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .build();

        listMets(cw, namespace);
        cw.close();
    }

    public static void listMets(CloudWatchClient cw, String namespace) {
        boolean done = false;
        String nextToken = null;

        try {
            while (!done) {

                ListMetricsResponse response;
                if (nextToken == null) {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .build();

                    response = cw.listMetrics(request);
                } else {
                    ListMetricsRequest request = ListMetricsRequest.builder()
                            .namespace(namespace)
                            .nextToken(nextToken)
                            .build();

                    response = cw.listMetrics(request);
                }

                for (Metric metric : response.metrics()) {
                    System.out.printf("Retrieved metric %s",
metric.metricName());
                    System.out.println();
                }

                if (response.nextToken() == null) {
                    done = true;
                } else {
                    nextToken = response.nextToken();
```

```
                }
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà 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.

Importez le kit SDK et les modules client et appelez l'API.

```
import { ListMetricsCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

export const main = () => {
  // Use the AWS console to see available namespaces and metric names. Custom
 metrics can also be created.
  // https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
viewing_metrics_with_cloudwatch.html
  const command = new ListMetricsCommand({
    Dimensions: [
      {
        Name: "LogGroupName",
      },
    ],
    MetricName: "IncomingLogEvents",
    Namespace: "AWS/Logs",
```

```
  });

  return client.send(command);
};
```

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section Référence des AWS SDK for 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 CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  Dimensions: [
    {
      Name: "LogGroupName" /* required */,
    },
  ],
  MetricName: "IncomingLogEvents",
  Namespace: "AWS/Logs",
```

```
};

cw.listMetrics(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Metrics", JSON.stringify(data.Metrics));
  }
});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section Référence des AWS SDK for 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 listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
```

```
}
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def list_metrics(self, namespace, name, recent=False):
        """
        Gets the metrics within a namespace that have the specified name.
        If the metric has no dimensions, a single metric is returned.
        Otherwise, metrics for all dimensions are returned.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param recent: When True, only metrics that have been active in the last
                       three hours are returned.
        :return: An iterator that yields the retrieved metrics.
        """
        try:
            kwargs = {"Namespace": namespace, "MetricName": name}
            if recent:
```

```
                kwargs["RecentlyActive"] = "PT3H"  # List past 3 hours only
            metric_iter = self.cloudwatch_resource.metrics.filter(**kwargs)
            logger.info("Got metrics for %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't get metrics for %s.%s.", namespace, name)
            raise
        else:
            return metric_iter
```

- Pour plus de détails sur l'API, consultez ListMetricsle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **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.

```
# Lists available metrics for a metric namespace in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric.
# @example
#   list_metrics_for_namespace(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'SITE/TRAFFIC'
#   )
def list_metrics_for_namespace(cloudwatch_client, metric_namespace)
  response = cloudwatch_client.list_metrics(namespace: metric_namespace)

  if response.metrics.count.positive?
    response.metrics.each do |metric|
      puts "  Metric name: #{metric.metric_name}"
      if metric.dimensions.count.positive?
```

```
        puts "    Dimensions:"
        metric.dimensions.each do |dimension|
          puts "      Name: #{dimension.name}, Value: #{dimension.value}"
        end
      else
        puts "No dimensions found."
      end
    end
  else
    puts "No metrics found for namespace '#{metric_namespace}'. " \
      "Note that it could take up to 15 minutes for recently-added metrics " \
      "to become available."
  end
end

# Example usage:
def run_me
  metric_namespace = "SITE/TRAFFIC"
  # Replace us-west-2 with the AWS Region you're using for Amazon CloudWatch.
  region = "us-east-1"

  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)

  # Add three datapoints.
  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisitors",
    "SiteName",
    "example.com",
    5_885.0,
    "Count"
  )

  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "UniqueVisits",
    "SiteName",
    "example.com",
    8_628.0,
    "Count"
  )
```

```
  puts "Continuing..." unless datapoint_added_to_metric?(
    cloudwatch_client,
    metric_namespace,
    "PageViews",
    "PageURL",
    "example.html",
    18_057.0,
    "Count"
  )

  puts "Metrics for namespace '#{metric_namespace}':"
  list_metrics_for_namespace(cloudwatch_client, metric_namespace)
end

run_me if $PROGRAM_NAME == __FILE__
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà la section Référence des AWS SDK for Ruby API.

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.

```
   "The following list-metrics example displays the metrics for Amazon
CloudWatch."
   TRY.
       oo_result = lo_cwt->listmetrics(           " oo_result is returned for
testing purposes. "
         iv_namespace = iv_namespace
       ).
       DATA(lt_metrics) = oo_result->get_metrics( ).
       MESSAGE 'Metrics retrieved.' TYPE 'I'.
     CATCH /aws1/cx_cwtinvparamvalueex .
       MESSAGE 'The specified argument was not valid.' TYPE 'E'.
```

```
    ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous ListMetricsà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutAnomalyDetector`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
    {
        var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
```

```
            new PutAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous PutAnomalyDetectorà la section Référence des AWS SDK for .NET 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.

```
    public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();
```

```
            PutAnomalyDetectorRequest anomalyDetectorRequest =
 PutAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.putAnomalyDetector(anomalyDetectorRequest);
            System.out.println("Added anomaly detector for metric " +
 customMetricName + ".");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutAnomalyDetectorà la section Référence des AWS SDK for Java 2.x 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 addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
```

```
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutAnomalyDetectorà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutDashboard`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes

.NET

AWS SDK for .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.

```
    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
```

```
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
 newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
 was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }
```

- Pour plus de détails sur l'API, reportez-vous PutDashboardà la section Référence des AWS SDK for .NET 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.

```
   public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
       try {
           PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                   .dashboardName(dashboardName)
                   .dashboardBody(readFileAsString(fileName))
                   .build();

           PutDashboardResponse response = cw.putDashboard(dashboardRequest);
           System.out.println(dashboardName + " was successfully created.");
           List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
           if (messages.isEmpty()) {
               System.out.println("There are no messages in the new Dashboard");
           } else {
               for (DashboardValidationMessage message : messages) {
                   System.out.println("Message is: " + message.message());
               }
           }

       } catch (CloudWatchException | IOException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous PutDashboardà la section Référence des AWS SDK for Java 2.x 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 createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutDashboardà la section AWS SDK pour la référence de l'API Kotlin.

PowerShell

Outils pour PowerShell

Exemple 1 : crée ou met à jour le tableau de bord nommé « Dashboard1 » pour inclure deux widgets métriques côte à côte.

```
$dashBody = @"
{
    "widgets":[
        {
             "type":"metric",
             "x":0,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/EC2",
                      "CPUUtilization",
                      "InstanceId",
                      "i-012345"
                   ]
                ],
                "period":300,
                "stat":"Average",
                "region":"us-east-1",
                "title":"EC2 Instance CPU"
             }
        },
        {
             "type":"metric",
             "x":12,
             "y":0,
             "width":12,
             "height":6,
             "properties":{
                "metrics":[
                   [
                      "AWS/S3",
                      "BucketSizeBytes",
                      "BucketName",
                      "MyBucketName"
```

```
                    ]
                ],
                "period":86400,
                "stat":"Maximum",
                "region":"us-east-1",
                "title":"MyBucketName bytes"
            }
        }
    ]
}
"@

Write-CWDashboard -DashboardName Dashboard1 -DashboardBody $dashBody
```

Exemple 2 : crée ou met à jour le tableau de bord, en redirigeant le contenu décrivant le tableau de bord vers l'applet de commande.

```
$dashBody = @"
{
...
}
"@

$dashBody | Write-CWDashboard -DashboardName Dashboard1
```

- Pour plus de détails sur l'API, reportez-vous PutDashboardà la section Référence des AWS Tools for PowerShell applets de commande.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutMetricAlarm`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage des alarmes

- Démarrage avec les métriques, tableaux de bord et alertes
- Gérer les mesures et les alertes

.NET

AWS SDK for .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.

```
   /// <summary>
   /// Add a metric alarm to send an email when the metric passes a threshold.
   /// </summary>
   /// <param name="alarmDescription">A description of the alarm.</param>
   /// <param name="alarmName">The name for the alarm.</param>
   /// <param name="comparison">The type of comparison to use.</param>
   /// <param name="metricName">The name of the metric for the alarm.</param>
   /// <param name="metricNamespace">The namespace of the metric.</param>
   /// <param name="threshold">The threshold value for the alarm.</param>
   /// <param name="alarmActions">Optional actions to execute when in an alarm
state.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
alarmName, ComparisonOperator comparison,
       string metricName, string metricNamespace, double threshold, List<string>
alarmActions = null!)
   {
       try
       {
           var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
               new PutMetricAlarmRequest()
               {
                   AlarmActions = alarmActions,
                   AlarmDescription = alarmDescription,
                   AlarmName = alarmName,
                   ComparisonOperator = comparison,
                   Threshold = threshold,
```

```
                    Namespace = metricNamespace,
                    MetricName = metricName,
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
 has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
 alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
 topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
 append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Inclut les fichiers requis.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricAlarmRequest.h>
#include <iostream>
```

Créez l'alerte pour regarder la mesure.

```
        Aws::CloudWatch::CloudWatchClient cw;
        Aws::CloudWatch::Model::PutMetricAlarmRequest request;
        request.SetAlarmName(alarm_name);
        request.SetComparisonOperator(
            Aws::CloudWatch::Model::ComparisonOperator::GreaterThanThreshold);
        request.SetEvaluationPeriods(1);
        request.SetMetricName("CPUUtilization");
        request.SetNamespace("AWS/EC2");
        request.SetPeriod(60);
        request.SetStatistic(Aws::CloudWatch::Model::Statistic::Average);
        request.SetThreshold(70.0);
        request.SetActionsEnabled(false);
        request.SetAlarmDescription("Alarm when server CPU exceeds 70%");
        request.SetUnit(Aws::CloudWatch::Model::StandardUnit::Seconds);

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("InstanceId");
        dimension.SetValue(instanceId);

        request.AddDimensions(dimension);

        auto outcome = cw.PutMetricAlarm(request);
```

```
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch alarm:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch alarm " << alarm_name
                << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour envoyer un e-mail Amazon Simple Notification Service lorsque l'utilisation du processeur dépasse 70 %

L'exemple suivant fait appel à la commande `put-metric-alarm` pour envoyer un message e-mail Amazon Simple Notification Service lorsque l'utilisation du processeur dépasse 70 % :

```
aws cloudwatch put-metric-alarm --alarm-name cpu-mon --alarm-description "Alarm
 when CPU exceeds 70 percent" --metric-name CPUUtilization --namespace AWS/
EC2 --statistic Average --period 300 --threshold 70 --comparison-operator
 GreaterThanThreshold  --dimensions "Name=InstanceId,Value=i-12345678" --
evaluation-periods 2 --alarm-actions arn:aws:sns:us-east-1:111122223333:MyTopic
 --unit Percent
```

Cette commande revient à l'invite en cas de succès. Si une alarme portant le même nom existe déjà, elle sera remplacée par la nouvelle alarme.

Pour spécifier plusieurs dimensions

L'exemple suivant illustre comment spécifier plusieurs dimensions. Chaque dimension est spécifiée sous la forme d'une paire nom/valeur, avec une virgule entre le nom et la valeur. Les différentes dimensions sont séparées par une espace :

```
aws cloudwatch put-metric-alarm --alarm-name "Default_Test_Alarm3" --alarm-
description "The default example alarm" --namespace "CW EXAMPLE METRICS" --
metric-name Default_Test --statistic Average --period 60 --evaluation-periods 3
 --threshold 50 --comparison-operator GreaterThanOrEqualToThreshold --dimensions
 Name=key1,Value=value1 Name=key2,Value=value2
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
```

```
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Importez le kit SDK et les modules client et appelez l'API.

```
import { PutMetricAlarmCommand } from "@aws-sdk/client-cloudwatch";
```

```
import { client } from "../libs/client.js";

const run = async () => {
  // This alarm triggers when CPUUtilization exceeds 70% for one minute.
  const command = new PutMetricAlarmCommand({
    AlarmName: process.env.CLOUDWATCH_ALARM_NAME, // Set the value of
 CLOUDWATCH_ALARM_NAME to the name of an existing alarm.
    ComparisonOperator: "GreaterThanThreshold",
    EvaluationPeriods: 1,
    MetricName: "CPUUtilization",
    Namespace: "AWS/EC2",
    Period: 60,
    Statistic: "Average",
    Threshold: 70.0,
    ActionsEnabled: false,
    AlarmDescription: "Alarm when server CPU exceeds 70%",
    Dimensions: [
      {
        Name: "InstanceId",
        Value: process.env.EC2_INSTANCE_ID, // Set the value of EC_INSTANCE_ID to
 the Id of an existing Amazon EC2 instance.
      },
    ],
    Unit: "Percent",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS SDK for JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

var params = {
  AlarmName: "Web_Server_CPU_Utilization",
  ComparisonOperator: "GreaterThanThreshold",
  EvaluationPeriods: 1,
  MetricName: "CPUUtilization",
  Namespace: "AWS/EC2",
  Period: 60,
  Statistic: "Average",
  Threshold: 70.0,
  ActionsEnabled: false,
  AlarmDescription: "Alarm when server CPU exceeds 70%",
  Dimensions: [
    {
      Name: "InstanceId",
      Value: "INSTANCE_ID",
    },
  ],
  Unit: "Percent",
};

cw.putMetricAlarm(params, function (err, data) {
```

```
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun putMetricAlarm(alarmNameVal: String, instanceIdVal: String) {

    val dimensionOb = Dimension {
        name = "InstanceId"
        value = instanceIdVal
    }

    val request = PutMetricAlarmRequest {
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanThreshold
        evaluationPeriods = 1
        metricName = "CPUUtilization"
        namespace = "AWS/EC2"
        period = 60
        statistic = Statistic.fromValue("Average")
        threshold = 70.0
        actionsEnabled = false
```

```
        alarmDescription = "An Alarm created by the Kotlin SDK when server CPU
 utilization exceeds 70%"
        unit = StandardUnit.fromValue("Seconds")
        dimensions = listOf(dimensionOb)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(request)
        println("Successfully created an alarm with name $alarmNameVal")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section AWS SDK pour la référence de l'API Kotlin.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
```

```
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
```

```
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm
```

- Pour plus de détails sur l'API, consultez PutMetricAlarmle AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
# Creates or updates an alarm in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param alarm_name [String] The name of the alarm.
# @param alarm_description [String] A description about the alarm.
# @param metric_name [String] The name of the metric associated with the alarm.
# @param alarm_actions [Array] A list of Strings representing the
#   Amazon Resource Names (ARNs) to execute when the alarm transitions to the
#   ALARM state.
# @param namespace [String] The namespace for the metric to alarm on.
# @param statistic [String] The statistic for the metric.
# @param dimensions [Array] A list of dimensions for the metric, specified as
#   Aws::CloudWatch::Types::Dimension.
# @param period [Integer] The number of seconds before re-evaluating the metric.
```

```
# @param unit [String] The unit of measure for the statistic.
# @param evaluation_periods [Integer] The number of periods over which data is
#   compared to the specified threshold.
# @param theshold [Float] The value against which the specified statistic is
 compared.
# @param comparison_operator [String] The arithmetic operation to use when
#   comparing the specified statistic and threshold.
# @return [Boolean] true if the alarm was created or updated; otherwise, false.
# @example
#   exit 1 unless alarm_created_or_updated?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'ObjectsInBucket',
#     'Objects exist in this bucket for more than 1 day.',
#     'NumberOfObjects',
#     ['arn:aws:sns:us-east-1:111111111111:Default_CloudWatch_Alarms_Topic'],
#     'AWS/S3',
#     'Average',
#     [
#       {
#         name: 'BucketName',
#         value: 'doc-example-bucket'
#       },
#       {
#         name: 'StorageType',
#         value: 'AllStorageTypes'
#       }
#     ],
#     86_400,
#     'Count',
#     1,
#     1,
#     'GreaterThanThreshold'
#   )
def alarm_created_or_updated?(
  cloudwatch_client,
  alarm_name,
  alarm_description,
  metric_name,
  alarm_actions,
  namespace,
  statistic,
  dimensions,
  period,
  unit,
```

```
  evaluation_periods,
  threshold,
  comparison_operator
)
  cloudwatch_client.put_metric_alarm(
    alarm_name: alarm_name,
    alarm_description: alarm_description,
    metric_name: metric_name,
    alarm_actions: alarm_actions,
    namespace: namespace,
    statistic: statistic,
    dimensions: dimensions,
    period: period,
    unit: unit,
    evaluation_periods: evaluation_periods,
    threshold: threshold,
    comparison_operator: comparison_operator
  )
  return true
rescue StandardError => e
  puts "Error creating alarm: #{e.message}"
  return false
end
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà la section Référence des AWS SDK for Ruby API.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
    TRY.
        lo_cwt->putmetricalarm(
          iv_alarmname                = iv_alarm_name
```

```
        iv_comparisonoperator        = iv_comparison_operator
        iv_evaluationperiods         = iv_evaluation_periods
        iv_metricname                = iv_metric_name
        iv_namespace                 = iv_namespace
        iv_statistic                 = iv_statistic
        iv_threshold                 = iv_threshold
        iv_actionsenabled            = iv_actions_enabled
        iv_alarmdescription          = iv_alarm_description
        iv_unit                      = iv_unit
        iv_period                    = iv_period
        it_dimensions                = it_dimensions
      ).
      MESSAGE 'Alarm created.' TYPE 'I'.
    CATCH /aws1/cx_cwtlimitexceededfault.
      MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
  ENDTRY.
```

- Pour plus de détails sur l'API, reportez-vous PutMetricAlarmà 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, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutMetricData`** avec un AWS SDK ou une CLI

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

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 :

- Démarrage avec les métriques, tableaux de bord et alertes
- Gérer les mesures et les alertes

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   /// <summary>
   /// Add some metric data using a call to a wrapper class.
   /// </summary>
   /// <param name="customMetricName">The metric name.</param>
   /// <param name="customMetricNamespace">The metric namespace.</param>
   /// <returns></returns>
   private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
       string customMetricNamespace)
   {
       List<MetricDatum> customData = new List<MetricDatum>();
       Random rnd = new Random();

       // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
       var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
       for (int i = 0; i < 10; i++)
       {
           var metricValue = rnd.Next(0, 100);
           customData.Add(
               new MetricDatum
               {
                   MetricName = customMetricName,
                   Value = metricValue,
                   TimestampUtc = utcNowMinus15.AddMinutes(i)
               }
           );
       }

       await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
```

```
        return customData;
    }

    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Joignez les fichiers requis.

```
#include <aws/core/Aws.h>
#include <aws/monitoring/CloudWatchClient.h>
#include <aws/monitoring/model/PutMetricDataRequest.h>
```

```
#include <iostream>
```

Placer des données dans la métrique

```
        Aws::CloudWatch::CloudWatchClient cw;

        Aws::CloudWatch::Model::Dimension dimension;
        dimension.SetName("UNIQUE_PAGES");
        dimension.SetValue("URLS");

        Aws::CloudWatch::Model::MetricDatum datum;
        datum.SetMetricName("PAGES_VISITED");
        datum.SetUnit(Aws::CloudWatch::Model::StandardUnit::None);
        datum.SetValue(data_point);
        datum.AddDimensions(dimension);

        Aws::CloudWatch::Model::PutMetricDataRequest request;
        request.SetNamespace("SITE/TRAFFIC");
        request.AddMetricData(datum);

        auto outcome = cw.PutMetricData(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to put sample metric data:" <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully put sample metric data" << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour publier une métrique personnalisée sur Amazon CloudWatch

L'exemple suivant utilise la `put-metric-data` commande pour publier une métrique personnalisée sur Amazon CloudWatch :

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

Les valeurs de la métrique elle-même sont stockées dans le fichier JSON, `metric.json`.

Voici le contenu de ce fichier :

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

Pour plus d'informations, consultez la section Publication de métriques personnalisées dans le manuel Amazon CloudWatch Developer Guide.

Pour spécifier plusieurs dimensions

L'exemple suivant illustre comment spécifier plusieurs dimensions. Chaque dimension est spécifiée sous la forme d'une paire nom=valeur. Les différentes dimensions sont séparées par une virgule :

```
aws cloudwatch put-metric-data --metric-name Buffers --namespace
 MyNameSpace --unit Bytes --value 231434333 --dimensions
 InstanceID=1-23456789,InstanceType=m1.small
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           // Set an Instant object.
           String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
           Instant instant = Instant.parse(time);

           MetricDatum datum = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1001.00)
                   .timestamp(instant)
                   .build();

           MetricDatum datum2 = MetricDatum.builder()
                   .metricName(customMetricName)
                   .unit(StandardUnit.NONE)
                   .value(1002.00)
                   .timestamp(instant)
                   .build();
```

```
            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Importez le kit SDK et les modules client et appelez l'API.

```
import { PutMetricDataCommand } from "@aws-sdk/client-cloudwatch";
import { client } from "../libs/client.js";

const run = async () => {
  // See https://docs.aws.amazon.com/AmazonCloudWatch/latest/APIReference/
API_PutMetricData.html#API_PutMetricData_RequestParameters
```

```
  // and https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
publishingMetrics.html
  // for more information about the parameters in this command.
  const command = new PutMetricDataCommand({
    MetricData: [
      {
        MetricName: "PAGES_VISITED",
        Dimensions: [
          {
            Name: "UNIQUE_PAGES",
            Value: "URLS",
          },
        ],
        Unit: "None",
        Value: 1.0,
      },
    ],
    Namespace: "SITE/TRAFFIC",
  });

  try {
    return await client.send(command);
  } catch (err) {
    console.error(err);
  }
};

export default run();
```

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchClient } from "@aws-sdk/client-cloudwatch";

export const client = new CloudWatchClient({});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS SDK for JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Load the AWS SDK for Node.js
var AWS = require("aws-sdk");
// Set the region
AWS.config.update({ region: "REGION" });

// Create CloudWatch service object
var cw = new AWS.CloudWatch({ apiVersion: "2010-08-01" });

// Create parameters JSON for putMetricData
var params = {
  MetricData: [
    {
      MetricName: "PAGES_VISITED",
      Dimensions: [
        {
          Name: "UNIQUE_PAGES",
          Value: "URLS",
        },
      ],
      Unit: "None",
      Value: 1.0,
    },
  ],
  Namespace: "SITE/TRAFFIC",
};

cw.putMetricData(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", JSON.stringify(data));
  }
});
```

- Pour de plus amples informations, consultez le Guide du développeur AWS SDK for JavaScript.
- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }
```

```
    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric $customMetricName")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section AWS SDK pour la référence de l'API Kotlin.

PowerShell

Outils pour PowerShell

Exemple 1 : crée un nouvel MetricDatum objet et l'écrit dans Amazon Web Services CloudWatch Metrics.

```
### Create a MetricDatum .NET object
$Metric = New-Object -TypeName Amazon.CloudWatch.Model.MetricDatum
$Metric.Timestamp = [DateTime]::UtcNow
$Metric.MetricName = 'CPU'
$Metric.Value = 50

### Write the metric data to the CloudWatch service
Write-CWMetricData -Namespace instance1 -MetricData $Metric
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS Tools for PowerShell applets de commande.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data(self, namespace, name, value, unit):
        """
        Sends a single data value to CloudWatch for a metric. This metric is
 given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
 name)
            raise
```

Insérez un ensemble de données dans une CloudWatch métrique.

```
class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
 counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
```

```
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise
```

- Pour plus de détails sur l'API, consultez PutMetricDatale AWS manuel de référence de l'API SDK for Python (Boto3).

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
require "aws-sdk-cloudwatch"

# Adds a datapoint to a metric in Amazon CloudWatch.
#
# @param cloudwatch_client [Aws::CloudWatch::Client]
#   An initialized CloudWatch client.
# @param metric_namespace [String] The namespace of the metric to add the
#   datapoint to.
# @param metric_name [String] The name of the metric to add the datapoint to.
# @param dimension_name [String] The name of the dimension to add the
#   datapoint to.
# @param dimension_value [String] The value of the dimension to add the
#   datapoint to.
# @param metric_value [Float] The value of the datapoint.
# @param metric_unit [String] The unit of measurement for the datapoint.
# @return [Boolean]
# @example
#   exit 1 unless datapoint_added_to_metric?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
```

```
#     'SITE/TRAFFIC',
#     'UniqueVisitors',
#     'SiteName',
#     'example.com',
#     5_885.0,
#     'Count'
#   )
def datapoint_added_to_metric?(
  cloudwatch_client,
  metric_namespace,
  metric_name,
  dimension_name,
  dimension_value,
  metric_value,
  metric_unit
)
  cloudwatch_client.put_metric_data(
    namespace: metric_namespace,
    metric_data: [
      {
        metric_name: metric_name,
        dimensions: [
          {
            name: dimension_name,
            value: dimension_value
          }
        ],
        value: metric_value,
        unit: metric_unit
      }
    ]
  )
  puts "Added data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}'."
  return true
rescue StandardError => e
  puts "Error adding data about '#{metric_name}' to namespace " \
    "'#{metric_namespace}': #{e.message}"
  return false
end
```

- Pour plus de détails sur l'API, reportez-vous PutMetricDataà la section Référence des AWS SDK for Ruby API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Scénarios d' CloudWatch utilisation des AWS SDK

Les exemples de code suivants vous montrent comment implémenter des scénarios courants CloudWatch avec AWS les SDK. Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions CloudWatch. Chaque scénario inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter le code.

Exemples

- Commencez à utiliser les CloudWatch alarmes à l'aide d'un AWS SDK
- Commencez à utiliser CloudWatch les métriques, les tableaux de bord et les alarmes à l'aide d'un SDK AWS
- Gérez les CloudWatch métriques et les alarmes à l'aide d'un AWS SDK

Commencez à utiliser les CloudWatch alarmes à l'aide d'un AWS SDK

L’exemple de code suivant illustre comment :

- Créer une alarme.
- Désactivez les actions d'alarme.
- Décrivez une alarme.
- Supprimez une alarme.

SAP ABAP

Kit SDK pour SAP ABAP

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
DATA lt_alarmnames TYPE /aws1/cl_cwtalarmnames_w=>tt_alarmnames.
DATA lo_alarmname TYPE REF TO /aws1/cl_cwtalarmnames_w.

"Create an alarm"
TRY.
    lo_cwt->putmetricalarm(
      iv_alarmname                = iv_alarm_name
      iv_comparisonoperator       = iv_comparison_operator
      iv_evaluationperiods        = iv_evaluation_periods
      iv_metricname               = iv_metric_name
      iv_namespace                = iv_namespace
      iv_statistic                = iv_statistic
      iv_threshold                = iv_threshold
      iv_actionsenabled           = iv_actions_enabled
      iv_alarmdescription         = iv_alarm_description
      iv_unit                     = iv_unit
      iv_period                   = iv_period
      it_dimensions               = it_dimensions
    ).
    MESSAGE 'Alarm created' TYPE 'I'.
  CATCH /aws1/cx_cwtlimitexceededfault.
    MESSAGE 'The request processing has exceeded the limit' TYPE 'E'.
ENDTRY.

"Create an ABAP internal table for the created alarm."
CREATE OBJECT lo_alarmname EXPORTING iv_value = iv_alarm_name.
INSERT lo_alarmname INTO TABLE lt_alarmnames.

"Disable alarm actions."
TRY.
    lo_cwt->disablealarmactions(
```

```
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarm actions disabled' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_disablealarm_exception).
        DATA(lv_disablealarm_error) = |"{ lo_disablealarm_exception-
>av_err_code }" - { lo_disablealarm_exception->av_err_msg }|.
        MESSAGE lv_disablealarm_error TYPE 'E'.
    ENDTRY.

    "Describe alarm using the same ABAP internal table."
    TRY.
        oo_result = lo_cwt->describealarms(                      " oo_result is
 returned for testing purpose "
          it_alarmnames               = lt_alarmnames
        ).
        MESSAGE 'Alarms retrieved' TYPE 'I'.
      CATCH /aws1/cx_rt_service_generic INTO DATA(lo_describealarms_exception).
        DATA(lv_describealarms_error) = |"{ lo_describealarms_exception-
>av_err_code }" - { lo_describealarms_exception->av_err_msg }|.
        MESSAGE lv_describealarms_error TYPE 'E'.
    ENDTRY.

    "Delete alarm."
    TRY.
        lo_cwt->deletealarms(
          it_alarmnames = lt_alarmnames
        ).
        MESSAGE 'Alarms deleted' TYPE 'I'.
      CATCH /aws1/cx_cwtresourcenotfound .
        MESSAGE 'Resource being access is not found.' TYPE 'E'.
    ENDTRY.
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API du kit AWS SDK pour SAP ABAP.
  - DeleteAlarms
  - DescribeAlarms
  - DisableAlarmActions
  - PutMetricAlarm

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Commencez à utiliser CloudWatch les métriques, les tableaux de bord et les alarmes à l'aide d'un SDK AWS

Les exemples de code suivants montrent comment :

- CloudWatch Répertoriez les espaces de noms et les métriques.
- obtenir les statistiques d'une métrique et de la facturation estimée ;
- créer et mettre à jour un tableau de bord ;
- créer et ajouter des données à une métrique ;
- créer et déclencher une alerte, puis consulter l'historique des alertes ;
- créer un détecteur d'anomalies ;
- obtenez une image de métrique, puis nettoyer les ressources.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exécutez un scénario interactif à une invite de commande.

```
public class CloudWatchScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
```

```
    https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

    This .NET example performs the following tasks:
        1. List and select a CloudWatch namespace.
        2. List and select a CloudWatch metric.
        3. Get statistics for a CloudWatch metric.
        4. Get estimated billing statistics for the last week.
        5. Create a new CloudWatch dashboard with two metrics.
        6. List current CloudWatch dashboards.
        7. Create a CloudWatch custom metric and add metric data.
        8. Add the custom metric to the dashboard.
        9. Create a CloudWatch alarm for the custom metric.
       10. Describe current CloudWatch alarms.
       11. Get recent data for the custom metric.
       12. Add data to the custom metric to trigger the alarm.
       13. Wait for an alarm state.
       14. Get history for the CloudWatch alarm.
       15. Add an anomaly detector.
       16. Describe current anomaly detectors.
       17. Get and display a metric image.
       18. Clean up resources.
    */

    private static ILogger logger = null!;
    private static CloudWatchWrapper _cloudWatchWrapper = null!;
    private static IConfiguration _configuration = null!;
    private static readonly List<string> _statTypes = new List<string>
{ "SampleCount", "Average", "Sum", "Minimum", "Maximum" };
    private static SingleMetricAnomalyDetector? anomalyDetector = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for the Amazon service.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonCloudWatch>()
            .AddTransient<CloudWatchWrapper>()
```

```
        )
        .Build();

        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<CloudWatchScenario>();

        _cloudWatchWrapper =
host.Services.GetRequiredService<CloudWatchWrapper>();

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon CloudWatch example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            var selectedNamespace = await SelectNamespace();
            var selectedMetric = await SelectMetric(selectedNamespace);
            await GetAndDisplayMetricStatistics(selectedNamespace,
selectedMetric);
            await GetAndDisplayEstimatedBilling();
            await CreateDashboardWithMetrics();
            await ListDashboards();
            await CreateNewCustomMetric();
            await AddMetricToDashboard();
            await CreateMetricAlarm();
            await DescribeAlarms();
            await GetCustomMetricData();
            await AddMetricDataForAlarm();
            await CheckForMetricAlarm();
            await GetAlarmHistory();
            anomalyDetector = await AddAnomalyDetector();
            await DescribeAnomalyDetectors();
            await GetAndOpenMetricImage();
            await CleanupResources();
        }
        catch (Exception ex)
        {
```

```
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources();
        }

    }

    /// <summary>
    /// Select a namespace.
    /// </summary>
    /// <returns>The selected namespace.</returns>
    private static async Task<string> SelectNamespace()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"1. Select a CloudWatch Namespace from a list of
Namespaces.");
        var metrics = await _cloudWatchWrapper.ListMetrics();
        // Get a distinct list of namespaces.
        var namespaces = metrics.Select(m => m.Namespace).Distinct().ToList();
        for (int i = 0; i < namespaces.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {namespaces[i]}");
        }

        var namespaceChoiceNumber = 0;
        while (namespaceChoiceNumber < 1 || namespaceChoiceNumber >
namespaces.Count)
        {
            Console.WriteLine(
                "Select a namespace by entering a number from the preceding
list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out namespaceChoiceNumber);
        }

        var selectedNamespace = namespaces[namespaceChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedNamespace;
    }

    /// <summary>
    /// Select a metric from a namespace.
    /// </summary>
```

```
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <returns>The metric name.</returns>
    private static async Task<Metric> SelectMetric(string metricNamespace)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"2. Select a CloudWatch metric from a namespace.");

        var namespaceMetrics = await
_cloudWatchWrapper.ListMetrics(metricNamespace);

        for (int i = 0; i < namespaceMetrics.Count && i < 15; i++)
        {
            var dimensionsWithValues = namespaceMetrics[i].Dimensions
                .Where(d => !string.Equals("None", d.Value));
            Console.WriteLine($"\t{i + 1}. {namespaceMetrics[i].MetricName} " +
                              $"{string.Join(", :", dimensionsWithValues.Select(d
=> d.Value))}");
        }

        var metricChoiceNumber = 0;
        while (metricChoiceNumber < 1 || metricChoiceNumber >
namespaceMetrics.Count)
        {
            Console.WriteLine(
                "Select a metric by entering a number from the preceding list:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out metricChoiceNumber);
        }

        var selectedMetric = namespaceMetrics[metricChoiceNumber - 1];

        Console.WriteLine(new string('-', 80));

        return selectedMetric;
    }

    /// <summary>
    /// Get and display metric statistics for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayMetricStatistics(string
metricNamespace, Metric metric)
```

```
   {
       Console.WriteLine(new string('-', 80));
       Console.WriteLine($"3. Get CloudWatch metric statistics for the last
day.");

       for (int i = 0; i < _statTypes.Count; i++)
       {
           Console.WriteLine($"\t{i + 1}. {_statTypes[i]}");
       }

       var statisticChoiceNumber = 0;
       while (statisticChoiceNumber < 1 || statisticChoiceNumber >
_statTypes.Count)
       {
           Console.WriteLine(
               "Select a metric statistic by entering a number from the
preceding list:");
           var choice = Console.ReadLine();
           Int32.TryParse(choice, out statisticChoiceNumber);
       }

       var selectedStatistic = _statTypes[statisticChoiceNumber - 1];
       var statisticsList = new List<string> { selectedStatistic };

       var metricStatistics = await
_cloudWatchWrapper.GetMetricStatistics(metricNamespace, metric.MetricName,
statisticsList, metric.Dimensions, 1, 60);

       if (!metricStatistics.Any())
       {
           Console.WriteLine($"No {selectedStatistic} statistics found for
{metric} in namespace {metricNamespace}.");
       }

       metricStatistics = metricStatistics.OrderBy(s => s.Timestamp).ToList();
       for (int i = 0; i < metricStatistics.Count && i < 10; i++)
       {
           var metricStat = metricStatistics[i];
           var statValue =
metricStat.GetType().GetProperty(selectedStatistic)!.GetValue(metricStat, null);
           Console.WriteLine($"\t{i + 1}. Timestamp
{metricStatistics[i].Timestamp:G} {selectedStatistic}: {statValue}");
       }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get and display estimated billing statistics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task GetAndDisplayEstimatedBilling()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"4. Get CloudWatch estimated billing for the last
week.");

        var billingStatistics = await SetupBillingStatistics();

        for (int i = 0; i < billingStatistics.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. Timestamp
{billingStatistics[i].Timestamp:G} : {billingStatistics[i].Maximum}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get billing statistics using a call to a wrapper class.
    /// </summary>
    /// <returns>A collection of billing statistics.</returns>
    private static async Task<List<Datapoint>> SetupBillingStatistics()
    {
        // Make a request for EstimatedCharges with a period of one day for the
past seven days.
        var billingStatistics = await _cloudWatchWrapper.GetMetricStatistics(
            "AWS/Billing",
            "EstimatedCharges",
            new List<string>() { "Maximum" },
            new List<Dimension>() { new Dimension { Name = "Currency", Value =
"USD" } },
            7,
            86400);

        billingStatistics = billingStatistics.OrderBy(n => n.Timestamp).ToList();
```

```
        return billingStatistics;
    }

    /// <summary>
    /// Create a dashboard with metrics.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CreateDashboardWithMetrics()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"5. Create a new CloudWatch dashboard with metrics.");
        var dashboardName = _configuration["dashboardName"];
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);
        var newDashboardString = JsonSerializer.Serialize(
            newDashboard,
            new JsonSerializerOptions
            {
                DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull
            });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} was created.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List dashboards.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListDashboards()
    {
        Console.WriteLine(new string('-', 80));
```

```
        Console.WriteLine($"6. List the CloudWatch dashboards in the current
account.");

        var dashboards = await _cloudWatchWrapper.ListDashboards();

        for (int i = 0; i < dashboards.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {dashboards[i].DashboardName}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and add data for a new custom metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateNewCustomMetric()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"7. Create and add data for a new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var customData = await PutRandomMetricData(customMetricName,
customMetricNamespace);

        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");

        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Add some metric data using a call to a wrapper class.
    /// </summary>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <returns></returns>
    private static async Task<List<MetricDatum>> PutRandomMetricData(string
customMetricName,
```

```
        string customMetricNamespace)
    {
        List<MetricDatum> customData = new List<MetricDatum>();
        Random rnd = new Random();

        // Add 10 random values up to 100, starting with a timestamp 15 minutes
in the past.
        var utcNowMinus15 = DateTime.UtcNow.AddMinutes(-15);
        for (int i = 0; i < 10; i++)
        {
            var metricValue = rnd.Next(0, 100);
            customData.Add(
                new MetricDatum
                {
                    MetricName = customMetricName,
                    Value = metricValue,
                    TimestampUtc = utcNowMinus15.AddMinutes(i)
                }
            );
        }

        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
        return customData;
    }

    /// <summary>
    /// Add the custom metric to the dashboard.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricToDashboard()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"8. Add the new custom metric to the dashboard.");

        var dashboardName = _configuration["dashboardName"];

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var validationMessages = await SetupDashboard(customMetricNamespace,
customMetricName, dashboardName);
```

```
        Console.WriteLine(validationMessages.Any() ? $"\tValidation messages:" :
null);
        for (int i = 0; i < validationMessages.Count; i++)
        {
            Console.WriteLine($"\t{i + 1}. {validationMessages[i].Message}");
        }
        Console.WriteLine($"\tDashboard {dashboardName} updated with metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }


    /// <summary>
    /// Set up a dashboard using a call to the wrapper class.
    /// </summary>
    /// <param name="customMetricNamespace">The metric namespace.</param>
    /// <param name="customMetricName">The metric name.</param>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A list of validation messages.</returns>
    private static async Task<List<DashboardValidationMessage>> SetupDashboard(
        string customMetricNamespace, string customMetricName, string
dashboardName)
    {
        // Get the dashboard model from configuration.
        var newDashboard = new DashboardModel();
        _configuration.GetSection("dashboardExampleBody").Bind(newDashboard);

        // Add a new metric to the dashboard.
        newDashboard.Widgets.Add(new Widget
        {
            Height = 8,
            Width = 8,
            Y = 8,
            X = 0,
            Type = "metric",
            Properties = new Properties
            {
                Metrics = new List<List<object>>
                    { new() { customMetricNamespace, customMetricName } },
                View = "timeSeries",
                Region = "us-east-1",
                Stat = "Sum",
                Period = 86400,
                YAxis = new YAxis { Left = new Left { Min = 0, Max = 100 } },
```

```
                Title = "Custom Metric Widget",
                LiveData = true,
                Sparkline = true,
                Trend = true,
                Stacked = false,
                SetPeriodToTimeRange = false
            }
        });

        var newDashboardString = JsonSerializer.Serialize(newDashboard,
            new JsonSerializerOptions
            { DefaultIgnoreCondition = JsonIgnoreCondition.WhenWritingNull });
        var validationMessages =
            await _cloudWatchWrapper.PutDashboard(dashboardName,
newDashboardString);

        return validationMessages;
    }

    /// <summary>
    /// Create a CloudWatch alarm for the new metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"9. Create a CloudWatch alarm for the new metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var alarmName = _configuration["exampleAlarmName"];
        var accountId = _configuration["accountId"];
        var region = _configuration["region"];
        var emailTopic = _configuration["emailTopic"];
        var alarmActions = new List<string>();

        if (GetYesNoResponse(
                $"\tAdd an email action for topic {emailTopic} to alarm
{alarmName}? (y/n)"))
        {
            _cloudWatchWrapper.AddEmailAlarmAction(accountId, region, emailTopic,
alarmActions);
        }
```

```
        await _cloudWatchWrapper.PutMetricEmailAlarm(
            "Example metric alarm",
            alarmName,
            ComparisonOperator.GreaterThanOrEqualToThreshold,
            customMetricName,
            customMetricNamespace,
            100,
            alarmActions);

        Console.WriteLine($"\tAlarm {alarmName} added for metric
{customMetricName}.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Describe Alarms.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAlarms()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"10. Describe CloudWatch alarms in the current
account.");

        var alarms = await _cloudWatchWrapper.DescribeAlarms();
        alarms = alarms.OrderByDescending(a => a.StateUpdatedTimestamp).ToList();

        for (int i = 0; i < alarms.Count && i < 10; i++)
        {
            var alarm = alarms[i];
            Console.WriteLine($"\t{i + 1}. {alarm.AlarmName}");
            Console.WriteLine($"\tState: {alarm.StateValue} for
{alarm.MetricName} {alarm.ComparisonOperator} {alarm.Threshold}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get the recent data for the metric.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetCustomMetricData()
```

```
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"11. Get current data for new custom metric.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var accountId = _configuration["accountId"];

        var query = new List<MetricDataQuery>
        {
            new MetricDataQuery
            {
                AccountId = accountId,
                Id = "m1",
                Label = "Custom Metric Data",
                MetricStat = new MetricStat
                {
                    Metric = new Metric
                    {
                        MetricName = customMetricName,
                        Namespace = customMetricNamespace,
                    },
                    Period = 1,
                    Stat = "Maximum"
                }
            }
        };

        var metricData = await _cloudWatchWrapper.GetMetricData(
            20,
            true,
            DateTime.UtcNow.AddMinutes(1),
            20,
            query);

        for (int i = 0; i < metricData.Count; i++)
        {
            for (int j = 0; j < metricData[i].Values.Count; j++)
            {
                Console.WriteLine(
                    $"\tTimestamp {metricData[i].Timestamps[j]:G} Value:
{metricData[i].Values[j]}");
            }
        }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add metric data to trigger an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task AddMetricDataForAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"12. Add metric data to the custom metric to trigger
an alarm.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var nowUtc = DateTime.UtcNow;
        List<MetricDatum> customData = new List<MetricDatum>
        {
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-2)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc.AddMinutes(-1)
            },
            new MetricDatum
            {
                MetricName = customMetricName,
                Value = 101,
                TimestampUtc = nowUtc
            }
        };
        var valuesString = string.Join(',', customData.Select(d => d.Value));
        Console.WriteLine($"\tAdded metric values for for metric
{customMetricName}: \n\t{valuesString}");
        await _cloudWatchWrapper.PutMetricData(customMetricNamespace,
customData);
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Check for a metric alarm using the DescribeAlarmsForMetric action.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CheckForMetricAlarm()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"13. Checking for an alarm state.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
        var hasAlarm = false;
        var retries = 10;
        while (!hasAlarm && retries > 0)
        {
            var alarms = await
_cloudWatchWrapper.DescribeAlarmsForMetric(customMetricNamespace,
customMetricName);
            hasAlarm = alarms.Any(a => a.StateValue == StateValue.ALARM);
            retries--;
            Thread.Sleep(20000);
        }

        Console.WriteLine(hasAlarm
            ? $"\tAlarm state found for {customMetricName}."
            : $"\tNo Alarm state found for {customMetricName} after 10
retries.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get history for an alarm.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAlarmHistory()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"14. Get alarm history.");

        var exampleAlarmName = _configuration["exampleAlarmName"];
```

```
        var alarmHistory = await
_cloudWatchWrapper.DescribeAlarmHistory(exampleAlarmName, 2);

        for (int i = 0; i < alarmHistory.Count; i++)
        {
            var history = alarmHistory[i];
            Console.WriteLine($"\t{i + 1}. {history.HistorySummary}, time
{history.Timestamp:g}");
        }
        if (!alarmHistory.Any())
        {
            Console.WriteLine($"\tNo alarm history data found for
{exampleAlarmName}.");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an anomaly detector.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<SingleMetricAnomalyDetector> AddAnomalyDetector()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"15. Add an anomaly detector.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detector = new SingleMetricAnomalyDetector
        {
            MetricName = customMetricName,
            Namespace = customMetricNamespace,
            Stat = "Maximum"
        };
        await _cloudWatchWrapper.PutAnomalyDetector(detector);
        Console.WriteLine($"\tAdded anomaly detector for metric
{customMetricName}.");

        Console.WriteLine(new string('-', 80));
        return detector;
    }
```

```
    /// <summary>
    /// Describe anomaly detectors.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task DescribeAnomalyDetectors()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"16. Describe anomaly detectors in the current
account.");

        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];

        var detectors = await
_cloudWatchWrapper.DescribeAnomalyDetectors(customMetricNamespace,
customMetricName);

        for (int i = 0; i < detectors.Count; i++)
        {
            var detector = detectors[i];
            Console.WriteLine($"\t{i + 1}.
{detector.SingleMetricAnomalyDetector.MetricName}, state
{detector.StateValue}");
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Fetch and open a metrics image for a CloudWatch metric and namespace.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetAndOpenMetricImage()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("17. Get a metric image from CloudWatch.");

        Console.WriteLine($"\tGetting Image data for custom metric.");
        var customMetricNamespace = _configuration["customMetricNamespace"];
        var customMetricName = _configuration["customMetricName"];
```

```
        var memoryStream = await
_cloudWatchWrapper.GetTimeSeriesMetricImage(customMetricNamespace,
customMetricName, "Maximum", 10);
        var file = _cloudWatchWrapper.SaveMetricImage(memoryStream,
"MetricImages");

        ProcessStartInfo info = new ProcessStartInfo();

        Console.WriteLine($"\tFile saved as {Path.GetFileName(file)}.");
        Console.WriteLine($"\tPress enter to open the image.");
        Console.ReadLine();
        info.FileName = Path.Combine("ms-photos://", file);
        info.UseShellExecute = true;
        info.CreateNoWindow = true;
        info.Verb = string.Empty;

        Process.Start(info);

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up created resources.
    /// </summary>
    /// <param name="metricNamespace">The namespace for metrics.</param>
    /// <param name="metric">The CloudWatch metric.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"18. Clean up resources.");

        var dashboardName = _configuration["dashboardName"];
        if (GetYesNoResponse($"\tDelete dashboard {dashboardName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting dashboard.");
            var dashboardList = new List<string> { dashboardName };
            await _cloudWatchWrapper.DeleteDashboards(dashboardList);
        }

        var alarmName = _configuration["exampleAlarmName"];
        if (GetYesNoResponse($"\tDelete alarm {alarmName}? (y/n)"))
        {
            Console.WriteLine($"\tCleaning up alarms.");
```

```
            var alarms = new List<string> { alarmName };
            await _cloudWatchWrapper.DeleteAlarms(alarms);
        }

        if (GetYesNoResponse($"\tDelete anomaly detector? (y/n)") &&
 anomalyDetector != null)
        {
            Console.WriteLine($"\tCleaning up anomaly detector.");

            await _cloudWatchWrapper.DeleteAnomalyDetector(
                anomalyDetector);
        }

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null &&
                       ynResponse.Equals("y",
                           StringComparison.InvariantCultureIgnoreCase);
        return response;
    }
}
```

Méthodes d'encapsulation utilisées par le scénario pour les CloudWatch actions.

```
/// <summary>
/// Wrapper class for Amazon CloudWatch methods.
/// </summary>
public class CloudWatchWrapper
{
    private readonly IAmazonCloudWatch _amazonCloudWatch;
    private readonly ILogger<CloudWatchWrapper> _logger;
```

```
    /// <summary>
    /// Constructor for the CloudWatch wrapper.
    /// </summary>
    /// <param name="amazonCloudWatch">The injected CloudWatch client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public CloudWatchWrapper(IAmazonCloudWatch amazonCloudWatch,
ILogger<CloudWatchWrapper> logger)

    {
        _logger = logger;
        _amazonCloudWatch = amazonCloudWatch;
    }

    /// <summary>
    /// List metrics available, optionally within a namespace.
    /// </summary>
    /// <param name="metricNamespace">Optional CloudWatch namespace to use when
listing metrics.</param>
    /// <param name="filter">Optional dimension filter.</param>
    /// <param name="metricName">Optional metric name filter.</param>
    /// <returns>The list of metrics.</returns>
    public async Task<List<Metric>> ListMetrics(string? metricNamespace = null,
DimensionFilter? filter = null, string? metricName = null)
    {
        var results = new List<Metric>();
        var paginateMetrics = _amazonCloudWatch.Paginators.ListMetrics(
            new ListMetricsRequest
            {
                Namespace = metricNamespace,
                Dimensions = filter != null ? new List<DimensionFilter>
{ filter } : null,
                MetricName = metricName
            });
        // Get the entire list using the paginator.
        await foreach (var metric in paginateMetrics.Metrics)
        {
            results.Add(metric);
        }

        return results;
    }

    /// <summary>
```

```
    /// Wrapper to get statistics for a specific CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <param name="statistics">The list of statistics to include.</param>
    /// <param name="dimensions">The list of dimensions to include.</param>
    /// <param name="days">The number of days in the past to include.</param>
    /// <param name="period">The period for the data.</param>
    /// <returns>A list of DataPoint objects for the statistics.</returns>
    public async Task<List<Datapoint>> GetMetricStatistics(string
 metricNamespace,
        string metricName, List<string> statistics, List<Dimension> dimensions,
 int days, int period)
    {
        var metricStatistics = await _amazonCloudWatch.GetMetricStatisticsAsync(
            new GetMetricStatisticsRequest()
            {
                Namespace = metricNamespace,
                MetricName = metricName,
                Dimensions = dimensions,
                Statistics = statistics,
                StartTimeUtc = DateTime.UtcNow.AddDays(-days),
                EndTimeUtc = DateTime.UtcNow,
                Period = period
            });

        return metricStatistics.Datapoints;
    }

    /// <summary>
    /// Wrapper to create or add to a dashboard with metrics.
    /// </summary>
    /// <param name="dashboardName">The name for the dashboard.</param>
    /// <param name="dashboardBody">The metric data in JSON for the dashboard.</
param>
    /// <returns>A list of validation messages for the dashboard.</returns>
    public async Task<List<DashboardValidationMessage>> PutDashboard(string
 dashboardName,
        string dashboardBody)
    {
        // Updating a dashboard replaces all contents.
        // Best practice is to include a text widget indicating this dashboard
 was created programmatically.
        var dashboardResponse = await _amazonCloudWatch.PutDashboardAsync(
```

```
            new PutDashboardRequest()
            {
                DashboardName = dashboardName,
                DashboardBody = dashboardBody
            });

        return dashboardResponse.DashboardValidationMessages;
    }


    /// <summary>
    /// Get information on a dashboard.
    /// </summary>
    /// <param name="dashboardName">The name of the dashboard.</param>
    /// <returns>A JSON object with dashboard information.</returns>
    public async Task<string> GetDashboard(string dashboardName)
    {
        var dashboardResponse = await _amazonCloudWatch.GetDashboardAsync(
            new GetDashboardRequest()
            {
                DashboardName = dashboardName
            });

        return dashboardResponse.DashboardBody;
    }


    /// <summary>
    /// Get a list of dashboards.
    /// </summary>
    /// <returns>A list of DashboardEntry objects.</returns>
    public async Task<List<DashboardEntry>> ListDashboards()
    {
        var results = new List<DashboardEntry>();
        var paginateDashboards = _amazonCloudWatch.Paginators.ListDashboards(
            new ListDashboardsRequest());
        // Get the entire list using the paginator.
        await foreach (var data in paginateDashboards.DashboardEntries)
        {
            results.Add(data);
        }

        return results;
    }
```

```
    /// <summary>
    /// Wrapper to add metric data to a CloudWatch metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricData">A data object for the metric data.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricData(string metricNamespace,
        List<MetricDatum> metricData)
    {
        var putDataResponse = await _amazonCloudWatch.PutMetricDataAsync(
            new PutMetricDataRequest()
            {
                MetricData = metricData,
                Namespace = metricNamespace,
            });

        return putDataResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Get an image for a metric graphed over time.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metric">The name of the metric.</param>
    /// <param name="stat">The name of the stat to chart.</param>
    /// <param name="period">The period to use for the chart.</param>
    /// <returns>A memory stream for the chart image.</returns>
    public async Task<MemoryStream> GetTimeSeriesMetricImage(string
metricNamespace, string metric, string stat, int period)
    {
        var metricImageWidget = new
        {
            title = "Example Metric Graph",
            view = "timeSeries",
            stacked = false,
            period = period,
            width = 1400,
            height = 600,
            metrics = new List<List<object>>
                { new() { metricNamespace, metric, new { stat } } }
        };
```

```
        var metricImageWidgetString =
 JsonSerializer.Serialize(metricImageWidget);
        var imageResponse = await _amazonCloudWatch.GetMetricWidgetImageAsync(
            new GetMetricWidgetImageRequest()
            {
                MetricWidget = metricImageWidgetString
            });

        return imageResponse.MetricWidgetImage;
    }

    /// <summary>
    /// Save a metric image to a file.
    /// </summary>
    /// <param name="memoryStream">The MemoryStream for the metric image.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The path to the file.</returns>
    public string SaveMetricImage(MemoryStream memoryStream, string metricName)
    {
        var metricFileName = $"{metricName}_{DateTime.Now.Ticks}.png";
        using var sr = new StreamReader(memoryStream);
        // Writes the memory stream to a file.
        File.WriteAllBytes(metricFileName, memoryStream.ToArray());
        var filePath = Path.Join(AppDomain.CurrentDomain.BaseDirectory,
            metricFileName);
        return filePath;
    }

    /// <summary>
    /// Get data for CloudWatch metrics.
    /// </summary>
    /// <param name="minutesOfData">The number of minutes of data to include.</
param>
    /// <param name="useDescendingTime">True to return the data descending by
 time.</param>
    /// <param name="endDateUtc">The end date for the data, in UTC.</param>
    /// <param name="maxDataPoints">The maximum data points to include.</param>
    /// <param name="dataQueries">Optional data queries to include.</param>
    /// <returns>A list of the requested metric data.</returns>
    public async Task<List<MetricDataResult>> GetMetricData(int minutesOfData,
 bool useDescendingTime, DateTime? endDateUtc = null,
        int maxDataPoints = 0, List<MetricDataQuery>? dataQueries = null)
    {
        var metricData = new List<MetricDataResult>();
```

```
        // If no end time is provided, use the current time for the end time.
        endDateUtc ??= DateTime.UtcNow;
        var timeZoneOffset =
 TimeZoneInfo.Local.GetUtcOffset(endDateUtc.Value.ToLocalTime());
        var startTimeUtc = endDateUtc.Value.AddMinutes(-minutesOfData);
        // The timezone string should be in the format +0000, so use the timezone
 offset to format it correctly.
        var timeZoneString = $"{timeZoneOffset.Hours:D2}
{timeZoneOffset.Minutes:D2}";
        var paginatedMetricData = _amazonCloudWatch.Paginators.GetMetricData(
            new GetMetricDataRequest()
            {
                StartTimeUtc = startTimeUtc,
                EndTimeUtc = endDateUtc.Value,
                LabelOptions = new LabelOptions { Timezone = timeZoneString },
                ScanBy = useDescendingTime ? ScanBy.TimestampDescending :
 ScanBy.TimestampAscending,
                MaxDatapoints = maxDataPoints,
                MetricDataQueries = dataQueries,
            });

        await foreach (var data in paginatedMetricData.MetricDataResults)
        {
            metricData.Add(data);
        }
        return metricData;
    }

    /// <summary>
    /// Add a metric alarm to send an email when the metric passes a threshold.
    /// </summary>
    /// <param name="alarmDescription">A description of the alarm.</param>
    /// <param name="alarmName">The name for the alarm.</param>
    /// <param name="comparison">The type of comparison to use.</param>
    /// <param name="metricName">The name of the metric for the alarm.</param>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="threshold">The threshold value for the alarm.</param>
    /// <param name="alarmActions">Optional actions to execute when in an alarm
 state.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutMetricEmailAlarm(string alarmDescription, string
 alarmName, ComparisonOperator comparison,
        string metricName, string metricNamespace, double threshold, List<string>
 alarmActions = null!)
```

```
    {
        try
        {
            var putEmailAlarmResponse = await
_amazonCloudWatch.PutMetricAlarmAsync(
                new PutMetricAlarmRequest()
                {
                    AlarmActions = alarmActions,
                    AlarmDescription = alarmDescription,
                    AlarmName = alarmName,
                    ComparisonOperator = comparison,
                    Threshold = threshold,
                    Namespace = metricNamespace,
                    MetricName = metricName,
                    EvaluationPeriods = 1,
                    Period = 10,
                    Statistic = new Statistic("Maximum"),
                    DatapointsToAlarm = 1,
                    TreatMissingData = "ignore"
                });
            return putEmailAlarmResponse.HttpStatusCode == HttpStatusCode.OK;
        }
        catch (LimitExceededException lex)
        {
            _logger.LogError(lex, $"Unable to add alarm {alarmName}. Alarm quota
has already been reached.");
        }

        return false;
    }

    /// <summary>
    /// Add specific email actions to a list of action strings for a CloudWatch
alarm.
    /// </summary>
    /// <param name="accountId">The AccountId for the alarm.</param>
    /// <param name="region">The region for the alarm.</param>
    /// <param name="emailTopicName">An Amazon Simple Notification Service (SNS)
topic for the alarm email.</param>
    /// <param name="alarmActions">Optional list of existing alarm actions to
append to.</param>
    /// <returns>A list of string actions for an alarm.</returns>
    public List<string> AddEmailAlarmAction(string accountId, string region,
        string emailTopicName, List<string>? alarmActions = null)
```

```
    {
        alarmActions ??= new List<string>();
        var snsAlarmAction = $"arn:aws:sns:{region}:{accountId}:
{emailTopicName}";
        alarmActions.Add(snsAlarmAction);
        return alarmActions;
    }

    /// <summary>
    /// Describe the current alarms, optionally filtered by state.
    /// </summary>
    /// <param name="stateValue">Optional filter for alarm state.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarms(StateValue? stateValue =
 null)
    {
        List<MetricAlarm> alarms = new List<MetricAlarm>();
        var paginatedDescribeAlarms =
 _amazonCloudWatch.Paginators.DescribeAlarms(
            new DescribeAlarmsRequest()
            {
                StateValue = stateValue
            });

        await foreach (var data in paginatedDescribeAlarms.MetricAlarms)
        {
            alarms.Add(data);
        }
        return alarms;
    }

    /// <summary>
    /// Describe the current alarms for a specific metric.
    /// </summary>
    /// <param name="metricNamespace">The namespace of the metric.</param>
    /// <param name="metricName">The name of the metric.</param>
    /// <returns>The list of alarm data.</returns>
    public async Task<List<MetricAlarm>> DescribeAlarmsForMetric(string
 metricNamespace, string metricName)
    {
        var alarmsResult = await _amazonCloudWatch.DescribeAlarmsForMetricAsync(
            new DescribeAlarmsForMetricRequest()
            {
                Namespace = metricNamespace,
```

```
                MetricName = metricName
            });

        return alarmsResult.MetricAlarms;
    }

    /// <summary>
    /// Describe the history of an alarm for a number of days in the past.
    /// </summary>
    /// <param name="alarmName">The name of the alarm.</param>
    /// <param name="historyDays">The number of days in the past.</param>
    /// <returns>The list of alarm history data.</returns>
    public async Task<List<AlarmHistoryItem>> DescribeAlarmHistory(string
alarmName, int historyDays)
    {
        List<AlarmHistoryItem> alarmHistory = new List<AlarmHistoryItem>();
        var paginatedAlarmHistory =
_amazonCloudWatch.Paginators.DescribeAlarmHistory(
            new DescribeAlarmHistoryRequest()
            {
                AlarmName = alarmName,
                EndDateUtc = DateTime.UtcNow,
                HistoryItemType = HistoryItemType.StateUpdate,
                StartDateUtc = DateTime.UtcNow.AddDays(-historyDays)
            });

        await foreach (var data in paginatedAlarmHistory.AlarmHistoryItems)
        {
            alarmHistory.Add(data);
        }
        return alarmHistory;
    }

    /// <summary>
    /// Delete a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAlarms(List<string> alarmNames)
    {
        var deleteAlarmsResult = await _amazonCloudWatch.DeleteAlarmsAsync(
            new DeleteAlarmsRequest()
            {
                AlarmNames = alarmNames
```

```
            });

        return deleteAlarmsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableAlarmActions(List<string> alarmNames)
    {
        var disableAlarmActionsResult = await
_amazonCloudWatch.DisableAlarmActionsAsync(
            new DisableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return disableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Enable the actions for a list of alarms from CloudWatch.
    /// </summary>
    /// <param name="alarmNames">A list of names of alarms.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableAlarmActions(List<string> alarmNames)
    {
        var enableAlarmActionsResult = await
_amazonCloudWatch.EnableAlarmActionsAsync(
            new EnableAlarmActionsRequest()
            {
                AlarmNames = alarmNames
            });

        return enableAlarmActionsResult.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Add an anomaly detector for a single metric.
    /// </summary>
    /// <param name="anomalyDetector">A single metric anomaly detector.</param>
    /// <returns>True if successful.</returns>
```

```
   public async Task<bool> PutAnomalyDetector(SingleMetricAnomalyDetector
anomalyDetector)
   {
       var putAlarmDetectorResult = await
_amazonCloudWatch.PutAnomalyDetectorAsync(
           new PutAnomalyDetectorRequest()
           {
               SingleMetricAnomalyDetector = anomalyDetector
           });

       return putAlarmDetectorResult.HttpStatusCode == HttpStatusCode.OK;
   }

   /// <summary>
   /// Describe anomaly detectors for a metric and namespace.
   /// </summary>
   /// <param name="metricNamespace">The namespace of the metric.</param>
   /// <param name="metricName">The metric of the anomaly detectors.</param>
   /// <returns>The list of detectors.</returns>
   public async Task<List<AnomalyDetector>> DescribeAnomalyDetectors(string
metricNamespace, string metricName)
   {
       List<AnomalyDetector> detectors = new List<AnomalyDetector>();
       var paginatedDescribeAnomalyDetectors =
_amazonCloudWatch.Paginators.DescribeAnomalyDetectors(
           new DescribeAnomalyDetectorsRequest()
           {
               MetricName = metricName,
               Namespace = metricNamespace
           });

       await foreach (var data in
paginatedDescribeAnomalyDetectors.AnomalyDetectors)
       {
           detectors.Add(data);
       }

       return detectors;
   }

   /// <summary>
   /// Delete a single metric anomaly detector.
   /// </summary>
   /// <param name="anomalyDetector">The anomaly detector to delete.</param>
```

```
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteAnomalyDetector(SingleMetricAnomalyDetector
 anomalyDetector)
    {
        var deleteAnomalyDetectorResponse = await
 _amazonCloudWatch.DeleteAnomalyDetectorAsync(
            new DeleteAnomalyDetectorRequest()
            {
                SingleMetricAnomalyDetector = anomalyDetector
            });

        return deleteAnomalyDetectorResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Delete a list of CloudWatch dashboards.
    /// </summary>
    /// <param name="dashboardNames">List of dashboard names to delete.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDashboards(List<string> dashboardNames)
    {
        var deleteDashboardsResponse = await
 _amazonCloudWatch.DeleteDashboardsAsync(
            new DeleteDashboardsRequest()
            {
                DashboardNames = dashboardNames
            });

        return deleteDashboardsResponse.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory
  - DescribeAlarms
  - DescribeAlarmsForMetric

- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import com.fasterxml.jackson.core.JsonFactory;
import com.fasterxml.jackson.core.JsonParser;
import com.fasterxml.jackson.databind.ObjectMapper;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchClient;
import software.amazon.awssdk.services.cloudwatch.model.AlarmHistoryItem;
import software.amazon.awssdk.services.cloudwatch.model.AlarmType;
import software.amazon.awssdk.services.cloudwatch.model.AnomalyDetector;
import software.amazon.awssdk.services.cloudwatch.model.CloudWatchException;
import software.amazon.awssdk.services.cloudwatch.model.ComparisonOperator;
import
 software.amazon.awssdk.services.cloudwatch.model.DashboardValidationMessage;
import software.amazon.awssdk.services.cloudwatch.model.Datapoint;
import software.amazon.awssdk.services.cloudwatch.model.DeleteAlarmsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DeleteAnomalyDetectorRequest;
```

```
import software.amazon.awssdk.services.cloudwatch.model.DeleteDashboardsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmHistoryResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsForMetricResponse;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsRequest;
import software.amazon.awssdk.services.cloudwatch.model.DescribeAlarmsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.DescribeAnomalyDetectorsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Dimension;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.GetMetricDataResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricStatisticsResponse;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageRequest;
import
 software.amazon.awssdk.services.cloudwatch.model.GetMetricWidgetImageResponse;
import software.amazon.awssdk.services.cloudwatch.model.HistoryItemType;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsRequest;
import software.amazon.awssdk.services.cloudwatch.model.ListMetricsResponse;
import software.amazon.awssdk.services.cloudwatch.model.Metric;
import software.amazon.awssdk.services.cloudwatch.model.MetricAlarm;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataQuery;
import software.amazon.awssdk.services.cloudwatch.model.MetricDataResult;
import software.amazon.awssdk.services.cloudwatch.model.MetricDatum;
import software.amazon.awssdk.services.cloudwatch.model.MetricStat;
import
 software.amazon.awssdk.services.cloudwatch.model.PutAnomalyDetectorRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutDashboardResponse;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricAlarmRequest;
import software.amazon.awssdk.services.cloudwatch.model.PutMetricDataRequest;
import software.amazon.awssdk.services.cloudwatch.model.ScanBy;
import
 software.amazon.awssdk.services.cloudwatch.model.SingleMetricAnomalyDetector;
```

```
import software.amazon.awssdk.services.cloudwatch.model.StandardUnit;
import software.amazon.awssdk.services.cloudwatch.model.Statistic;
import
 software.amazon.awssdk.services.cloudwatch.paginators.ListDashboardsIterable;
import software.amazon.awssdk.services.cloudwatch.paginators.ListMetricsIterable;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.ZoneOffset;
import java.time.ZonedDateTime;
import java.time.format.DateTimeFormatter;
import java.time.temporal.ChronoUnit;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

/**
 * 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
 *
 * To enable billing metrics and statistics for this example, make sure billing
 * alerts are enabled for your account:
 * https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics
 *
 * This Java code example performs the following tasks:
 *
 * 1. List available namespaces from Amazon CloudWatch.
 * 2. List available metrics within the selected Namespace.
 * 3. Get statistics for the selected metric over the last day.
 * 4. Get CloudWatch estimated billing for the last week.
 * 5. Create a new CloudWatch dashboard with metrics.
 * 6. List dashboards using a paginator.
 * 7. Create a new custom metric by adding data for it.
```

```
 * 8. Add the custom metric to the dashboard.
 * 9. Create an alarm for the custom metric.
 * 10. Describe current alarms.
 * 11. Get current data for the new custom metric.
 * 12. Push data into the custom metric to trigger the alarm.
 * 13. Check the alarm state using the action DescribeAlarmsForMetric.
 * 14. Get alarm history for the new alarm.
 * 15. Add an anomaly detector for the custom metric.
 * 16. Describe current anomaly detectors.
 * 17. Get a metric image for the custom metric.
 * 18. Clean up the Amazon CloudWatch resources.
 */
public class CloudWatchScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws IOException {
        final String usage = """

                Usage:
                  <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage> \s

                Where:
                  myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)\s
                  costDateWeek - The start date to use to get AWS/Billinget
 statistics. (For example, 2023-01-11T18:35:24.00Z.)\s
                  dashboardName - The name of the dashboard to create.\s
                  dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)\s
                  dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)\s
                  settings - The location of a JSON file from which various
 values are read. (See Readme file.)\s
                  metricImage - The location of a BMP file that is used to create
 a graph.\s
                """;

        if (args.length != 7) {
            System.out.println(usage);
            System.exit(1);
        }
```

```
        Region region = Region.US_EAST_1;
        String myDate = args[0];
        String costDateWeek = args[1];
        String dashboardName = args[2];
        String dashboardJson = args[3];
        String dashboardAdd = args[4];
        String settings = args[5];
        String metricImage = args[6];

        Double dataPoint = Double.parseDouble("10.0");
        Scanner sc = new Scanner(System.in);
        CloudWatchClient cw = CloudWatchClient.builder()
                .region(region)
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon CloudWatch example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "1. List at least five available unique namespaces from Amazon
CloudWatch. Select one from the list.");
        ArrayList<String> list = listNameSpaces(cw);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + list.get(z));
        }

        String selectedNamespace = "";
        String selectedMetrics = "";
        int num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedNamespace = list.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("2. List available metrics within the selected
namespace and select one from the list.");
        ArrayList<String> metList = listMets(cw, selectedNamespace);
        for (int z = 0; z < 5; z++) {
            int index = z + 1;
            System.out.println("    " + index + ". " + metList.get(z));
        }
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            selectedMetrics = metList.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + selectedMetrics);
        Dimension myDimension = getSpecificMet(cw, selectedNamespace);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get statistics for the selected metric over the
last day.");
        String metricOption = "";
        ArrayList<String> statTypes = new ArrayList<>();
        statTypes.add("SampleCount");
        statTypes.add("Average");
        statTypes.add("Sum");
        statTypes.add("Minimum");
        statTypes.add("Maximum");

        for (int t = 0; t < 5; t++) {
            System.out.println("    " + (t + 1) + ". " + statTypes.get(t));
        }
        System.out.println("Select a metric statistic by entering a number from
the preceding list:");
        num = Integer.parseInt(sc.nextLine());
        if (1 <= num && num <= 5) {
            metricOption = statTypes.get(num - 1);
        } else {
            System.out.println("You did not select a valid option.");
            System.exit(1);
        }
        System.out.println("You selected " + metricOption);
        getAndDisplayMetricStatistics(cw, selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get CloudWatch estimated billing for the last
week.");
        getMetricStatistics(cw, costDateWeek);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Create a new CloudWatch dashboard with metrics.");
        createDashboardWithMetrics(cw, dashboardName, dashboardJson);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. List dashboards using a paginator.");
        listDashboards(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create a new custom metric by adding data to
it.");
        createNewCustomMetric(cw, dataPoint);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Add an additional metric to the dashboard.");
        addMetricToDashboard(cw, dashboardAdd, dashboardName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("9. Create an alarm for the custom metric.");
        String alarmName = createAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Describe ten current alarms.");
        describeAlarms(cw);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Get current data for new custom metric.");
        getCustomMetricData(cw, settings);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("12. Push data into the custom metric to trigger the
alarm.");
        addMetricDataForAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Check the alarm state using the action
DescribeAlarmsForMetric.");
        checkForMetricAlarm(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Get alarm history for the new alarm.");
        getAlarmHistory(cw, settings, myDate);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("15. Add an anomaly detector for the custom metric.");
        addAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("16. Describe current anomaly detectors.");
        describeAnomalyDetectors(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Get a metric image for the custom metric.");
        getAndOpenMetricImage(cw, metricImage);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up the Amazon CloudWatch resources.");
        deleteDashboard(cw, dashboardName);
        deleteCWAlarm(cw, alarmName);
        deleteAnomalyDetector(cw, settings);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Amazon CloudWatch example scenario is
complete.");
        System.out.println(DASHES);
        cw.close();
```

```
   }

   public static void deleteAnomalyDetector(CloudWatchClient cw, String
fileName) {
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();

           SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .stat("Maximum")
                   .build();

           DeleteAnomalyDetectorRequest request =
DeleteAnomalyDetectorRequest.builder()
                   .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                   .build();

           cw.deleteAnomalyDetector(request);
           System.out.println("Successfully deleted the Anomaly Detector.");

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       } catch (IOException e) {
           e.printStackTrace();
       }
   }

   public static void deleteCWAlarm(CloudWatchClient cw, String alarmName) {
       try {
           DeleteAlarmsRequest request = DeleteAlarmsRequest.builder()
                   .alarmNames(alarmName)
                   .build();
```

```
            cw.deleteAlarms(request);
            System.out.println("Successfully deleted alarm " + alarmName);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void deleteDashboard(CloudWatchClient cw, String dashboardName)
{
        try {
            DeleteDashboardsRequest dashboardsRequest =
DeleteDashboardsRequest.builder()
                    .dashboardNames(dashboardName)
                    .build();
            cw.deleteDashboards(dashboardsRequest);
            System.out.println(dashboardName + " was successfully deleted.");

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getAndOpenMetricImage(CloudWatchClient cw, String
fileName) {
        System.out.println("Getting Image data for custom metric.");
        try {
            String myJSON = "{\n" +
                    "  \"title\": \"Example Metric Graph\",\n" +
                    "  \"view\": \"timeSeries\",\n" +
                    "  \"stacked \": false,\n" +
                    "  \"period\": 10,\n" +
                    "  \"width\": 1400,\n" +
                    "  \"height\": 600,\n" +
                    "  \"metrics\": [\n" +
                    "    [\n" +
                    "      \"AWS/Billing\",\n" +
                    "      \"EstimatedCharges\",\n" +
                    "      \"Currency\",\n" +
                    "      \"USD\"\n" +
                    "    ]\n" +
                    "  ]\n" +
```

```
                    "}";

            GetMetricWidgetImageRequest imageRequest =
GetMetricWidgetImageRequest.builder()
                    .metricWidget(myJSON)
                    .build();

            GetMetricWidgetImageResponse response =
cw.getMetricWidgetImage(imageRequest);
            SdkBytes sdkBytes = response.metricWidgetImage();
            byte[] bytes = sdkBytes.asByteArray();
            File outputFile = new File(fileName);
            try (FileOutputStream outputStream = new
FileOutputStream(outputFile)) {
                outputStream.write(bytes);
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAnomalyDetectors(CloudWatchClient cw, String
fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            DescribeAnomalyDetectorsRequest detectorsRequest =
DescribeAnomalyDetectorsRequest.builder()
                    .maxResults(10)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            DescribeAnomalyDetectorsResponse response =
cw.describeAnomalyDetectors(detectorsRequest);
```

```
            List<AnomalyDetector> anomalyDetectorList =
response.anomalyDetectors();
            for (AnomalyDetector detector : anomalyDetectorList) {
                System.out.println("Metric name: " +
detector.singleMetricAnomalyDetector().metricName());
                System.out.println("State: " + detector.stateValue());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void addAnomalyDetector(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            SingleMetricAnomalyDetector singleMetricAnomalyDetector =
SingleMetricAnomalyDetector.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .stat("Maximum")
                    .build();

            PutAnomalyDetectorRequest anomalyDetectorRequest =
PutAnomalyDetectorRequest.builder()
                    .singleMetricAnomalyDetector(singleMetricAnomalyDetector)
                    .build();

            cw.putAnomalyDetector(anomalyDetectorRequest);
            System.out.println("Added anomaly detector for metric " +
customMetricName + ".");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
```

```
            System.exit(1);
        }
    }

    public static void getAlarmHistory(CloudWatchClient cw, String fileName,
String date) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String alarmName = rootNode.findValue("exampleAlarmName").asText();

            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();
            DescribeAlarmHistoryRequest historyRequest =
DescribeAlarmHistoryRequest.builder()
                    .startDate(start)
                    .endDate(endDate)
                    .alarmName(alarmName)
                    .historyItemType(HistoryItemType.ACTION)
                    .build();

            DescribeAlarmHistoryResponse response =
cw.describeAlarmHistory(historyRequest);
            List<AlarmHistoryItem> historyItems = response.alarmHistoryItems();
            if (historyItems.isEmpty()) {
                System.out.println("No alarm history data found for " + alarmName
+ ".");
            } else {
                for (AlarmHistoryItem item : historyItems) {
                    System.out.println("History summary: " +
item.historySummary());
                    System.out.println("Time stamp: " + item.timestamp());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

```
   public static void checkForMetricAlarm(CloudWatchClient cw, String fileName)
{
       try {
           // Read values from the JSON file.
           JsonParser parser = new JsonFactory().createParser(new
File(fileName));
           com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
           String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
           String customMetricName =
rootNode.findValue("customMetricName").asText();
           boolean hasAlarm = false;
           int retries = 10;

           DescribeAlarmsForMetricRequest metricRequest =
DescribeAlarmsForMetricRequest.builder()
                   .metricName(customMetricName)
                   .namespace(customMetricNamespace)
                   .build();

           while (!hasAlarm && retries > 0) {
               DescribeAlarmsForMetricResponse response =
cw.describeAlarmsForMetric(metricRequest);
               hasAlarm = response.hasMetricAlarms();
               retries--;
               Thread.sleep(20000);
               System.out.println(".");
           }
           if (!hasAlarm)
               System.out.println("No Alarm state found for " + customMetricName
+ " after 10 retries.");
           else
               System.out.println("Alarm state found for " + customMetricName +
".");

       } catch (CloudWatchException | IOException | InterruptedException e) {
           System.err.println(e.getMessage());
           System.exit(1);
       }
   }

   public static void addMetricDataForAlarm(CloudWatchClient cw, String
fileName) {
```

```
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1001.00)
                    .timestamp(instant)
                    .build();

            MetricDatum datum2 = MetricDatum.builder()
                    .metricName(customMetricName)
                    .unit(StandardUnit.NONE)
                    .value(1002.00)
                    .timestamp(instant)
                    .build();

            List<MetricDatum> metricDataList = new ArrayList<>();
            metricDataList.add(datum);
            metricDataList.add(datum2);

            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace(customMetricNamespace)
                    .metricData(metricDataList)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric " +
customMetricName);

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void getCustomMetricData(CloudWatchClient cw, String fileName)
{
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();

            // Set the date.
            Instant nowDate = Instant.now();

            long hours = 1;
            long minutes = 30;
            Instant date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(minutes,
                    ChronoUnit.MINUTES);

            Metric met = Metric.builder()
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .build();

            MetricStat metStat = MetricStat.builder()
                    .stat("Maximum")
                    .period(1)
                    .metric(met)
                    .build();

            MetricDataQuery dataQUery = MetricDataQuery.builder()
                    .metricStat(metStat)
                    .id("foo2")
                    .returnData(true)
                    .build();

            List<MetricDataQuery> dq = new ArrayList<>();
```

```
            dq.add(dataQUery);

            GetMetricDataRequest getMetReq = GetMetricDataRequest.builder()
                    .maxDatapoints(10)
                    .scanBy(ScanBy.TIMESTAMP_DESCENDING)
                    .startTime(nowDate)
                    .endTime(date2)
                    .metricDataQueries(dq)
                    .build();

            GetMetricDataResponse response = cw.getMetricData(getMetReq);
            List<MetricDataResult> data = response.metricDataResults();
            for (MetricDataResult item : data) {
                System.out.println("The label is " + item.label());
                System.out.println("The status code is " +
item.statusCode().toString());
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void describeAlarms(CloudWatchClient cw) {
        try {
            List<AlarmType> typeList = new ArrayList<>();
            typeList.add(AlarmType.METRIC_ALARM);

            DescribeAlarmsRequest alarmsRequest = DescribeAlarmsRequest.builder()
                    .alarmTypes(typeList)
                    .maxRecords(10)
                    .build();

            DescribeAlarmsResponse response = cw.describeAlarms(alarmsRequest);
            List<MetricAlarm> alarmList = response.metricAlarms();
            for (MetricAlarm alarm : alarmList) {
                System.out.println("Alarm name: " + alarm.alarmName());
                System.out.println("Alarm description: " +
alarm.alarmDescription());
            }
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
```

```
        }
    }

    public static String createAlarm(CloudWatchClient cw, String fileName) {
        try {
            // Read values from the JSON file.
            JsonParser parser = new JsonFactory().createParser(new
File(fileName));
            com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
            String customMetricNamespace =
rootNode.findValue("customMetricNamespace").asText();
            String customMetricName =
rootNode.findValue("customMetricName").asText();
            String alarmName = rootNode.findValue("exampleAlarmName").asText();
            String emailTopic = rootNode.findValue("emailTopic").asText();
            String accountId = rootNode.findValue("accountId").asText();
            String region = rootNode.findValue("region").asText();

            // Create a List for alarm actions.
            List<String> alarmActions = new ArrayList<>();
            alarmActions.add("arn:aws:sns:" + region + ":" + accountId + ":" +
emailTopic);
            PutMetricAlarmRequest alarmRequest = PutMetricAlarmRequest.builder()
                    .alarmActions(alarmActions)
                    .alarmDescription("Example metric alarm")
                    .alarmName(alarmName)

.comparisonOperator(ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD)
                    .threshold(100.00)
                    .metricName(customMetricName)
                    .namespace(customMetricNamespace)
                    .evaluationPeriods(1)
                    .period(10)
                    .statistic("Maximum")
                    .datapointsToAlarm(1)
                    .treatMissingData("ignore")
                    .build();

            cw.putMetricAlarm(alarmRequest);
            System.out.println(alarmName + " was successfully created!");
            return alarmName;

        } catch (CloudWatchException | IOException e) {
```

```
            System.err.println(e.getMessage());
            System.exit(1);
        }
        return "";
    }

    public static void addMetricToDashboard(CloudWatchClient cw, String fileName,
String dashboardName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully updated.");

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void createNewCustomMetric(CloudWatchClient cw, Double
dataPoint) {
        try {
            Dimension dimension = Dimension.builder()
                    .name("UNIQUE_PAGES")
                    .value("URLS")
                    .build();

            // Set an Instant object.
            String time =
ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT);
            Instant instant = Instant.parse(time);

            MetricDatum datum = MetricDatum.builder()
                    .metricName("PAGES_VISITED")
                    .unit(StandardUnit.NONE)
                    .value(dataPoint)
                    .timestamp(instant)
                    .dimensions(dimension)
                    .build();
```

```
            PutMetricDataRequest request = PutMetricDataRequest.builder()
                    .namespace("SITE/TRAFFIC")
                    .metricData(datum)
                    .build();

            cw.putMetricData(request);
            System.out.println("Added metric values for for metric
PAGES_VISITED");

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void listDashboards(CloudWatchClient cw) {
        try {
            ListDashboardsIterable listRes = cw.listDashboardsPaginator();
            listRes.stream()
                    .flatMap(r -> r.dashboardEntries().stream())
                    .forEach(entry -> {
                        System.out.println("Dashboard name is: " +
entry.dashboardName());
                        System.out.println("Dashboard ARN is: " +
entry.dashboardArn());
                    });

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void createDashboardWithMetrics(CloudWatchClient cw, String
dashboardName, String fileName) {
        try {
            PutDashboardRequest dashboardRequest = PutDashboardRequest.builder()
                    .dashboardName(dashboardName)
                    .dashboardBody(readFileAsString(fileName))
                    .build();

            PutDashboardResponse response = cw.putDashboard(dashboardRequest);
            System.out.println(dashboardName + " was successfully created.");
```

```
            List<DashboardValidationMessage> messages =
response.dashboardValidationMessages();
            if (messages.isEmpty()) {
                System.out.println("There are no messages in the new Dashboard");
            } else {
                for (DashboardValidationMessage message : messages) {
                    System.out.println("Message is: " + message.message());
                }
            }

        } catch (CloudWatchException | IOException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static String readFileAsString(String file) throws IOException {
        return new String(Files.readAllBytes(Paths.get(file)));
    }

    public static void getMetricStatistics(CloudWatchClient cw, String
costDateWeek) {
        try {
            Instant start = Instant.parse(costDateWeek);
            Instant endDate = Instant.now();
            Dimension dimension = Dimension.builder()
                    .name("Currency")
                    .value("USD")
                    .build();

            List<Dimension> dimensionList = new ArrayList<>();
            dimensionList.add(dimension);
            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .metricName("EstimatedCharges")
                    .namespace("AWS/Billing")
                    .dimensions(dimensionList)
                    .statistics(Statistic.MAXIMUM)
                    .startTime(start)
                    .endTime(endDate)
                    .period(86400)
                    .build();
```

```
            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void getAndDisplayMetricStatistics(CloudWatchClient cw, String
nameSpace, String metVal,
            String metricOption, String date, Dimension myDimension) {
        try {
            Instant start = Instant.parse(date);
            Instant endDate = Instant.now();

            GetMetricStatisticsRequest statisticsRequest =
GetMetricStatisticsRequest.builder()
                    .endTime(endDate)
                    .startTime(start)
                    .dimensions(myDimension)
                    .metricName(metVal)
                    .namespace(nameSpace)
                    .period(86400)
                    .statistics(Statistic.fromValue(metricOption))
                    .build();

            GetMetricStatisticsResponse response =
cw.getMetricStatistics(statisticsRequest);
            List<Datapoint> data = response.datapoints();
            if (!data.isEmpty()) {
                for (Datapoint datapoint : data) {
                    System.out
```

```
                            .println("Timestamp: " + datapoint.timestamp() + "
Maximum value: " + datapoint.maximum());
                }
            } else {
                System.out.println("The returned data list is empty");
            }

        } catch (CloudWatchException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static Dimension getSpecificMet(CloudWatchClient cw, String namespace)
{
        try {
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsResponse response = cw.listMetrics(request);
            List<Metric> myList = response.metrics();
            Metric metric = myList.get(0);
            return metric.dimensions().get(0);

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listMets(CloudWatchClient cw, String
namespace) {
        try {
            ArrayList<String> metList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .namespace(namespace)
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> metList.add(metrics.metricName()));
```

```
            return metList;

        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }

    public static ArrayList<String> listNameSpaces(CloudWatchClient cw) {
        try {
            ArrayList<String> nameSpaceList = new ArrayList<>();
            ListMetricsRequest request = ListMetricsRequest.builder()
                    .build();

            ListMetricsIterable listRes = cw.listMetricsPaginator(request);
            listRes.stream()
                    .flatMap(r -> r.metrics().stream())
                    .forEach(metrics -> {
                        String data = metrics.namespace();
                        if (!nameSpaceList.contains(data)) {
                            nameSpaceList.add(data);
                        }
                    });

            return nameSpaceList;
        } catch (CloudWatchException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return null;
    }
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - DeleteAlarms
  - DeleteAnomalyDetector
  - DeleteDashboards
  - DescribeAlarmHistory

- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
 Before running this Kotlin 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-kotlin/latest/developer-guide/setup.html

 To enable billing metrics and statistics for this example, make sure billing
 alerts are enabled for your account:
 https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/
monitor_estimated_charges_with_cloudwatch.html#turning_on_billing_metrics

 This Kotlin code example performs the following tasks:
```

```
 1. List available namespaces from Amazon CloudWatch. Select a namespace from the
 list.
 2. List available metrics within the selected namespace.
 3. Get statistics for the selected metric over the last day.
 4. Get CloudWatch estimated billing for the last week.
 5. Create a new CloudWatch dashboard with metrics.
 6. List dashboards using a paginator.
 7. Create a new custom metric by adding data for it.
 8. Add the custom metric to the dashboard.
 9. Create an alarm for the custom metric.
 10. Describe current alarms.
 11. Get current data for the new custom metric.
 12. Push data into the custom metric to trigger the alarm.
 13. Check the alarm state using the action DescribeAlarmsForMetric.
 14. Get alarm history for the new alarm.
 15. Add an anomaly detector for the custom metric.
 16. Describe current anomaly detectors.
 17. Get a metric image for the custom metric.
 18. Clean up the Amazon CloudWatch resources.
 */

val DASHES: String? = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <myDate> <costDateWeek> <dashboardName> <dashboardJson>
 <dashboardAdd> <settings> <metricImage>

        Where:
            myDate - The start date to use to get metric statistics. (For
 example, 2023-01-11T18:35:24.00Z.)
            costDateWeek - The start date to use to get AWS Billing and Cost
 Management statistics. (For example, 2023-01-11T18:35:24.00Z.)
            dashboardName - The name of the dashboard to create.
            dashboardJson - The location of a JSON file to use to create a
 dashboard. (See Readme file.)
            dashboardAdd - The location of a JSON file to use to update a
 dashboard. (See Readme file.)
            settings - The location of a JSON file from which various values are
 read. (See Readme file.)
            metricImage - The location of a BMP file that is used to create a
 graph.
    """
```

```
    if (args.size != 7) {
        println(usage)
        System.exit(1)
    }

    val myDate = args[0]
    val costDateWeek = args[1]
    val dashboardName = args[2]
    val dashboardJson = args[3]
    val dashboardAdd = args[4]
    val settings = args[5]
    var metricImage = args[6]
    val dataPoint = "10.0".toDouble()
    val inOb = Scanner(System.`in`)

    println(DASHES)
    println("Welcome to the Amazon CloudWatch example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. List at least five available unique namespaces from Amazon
 CloudWatch. Select a CloudWatch namespace from the list.")
    val list: ArrayList<String> = listNameSpaces()
    for (z in 0..4) {
        println("    ${z + 1}. ${list[z]}")
    }

    var selectedNamespace: String
    var selectedMetrics = ""
    var num = inOb.nextLine().toInt()
    println("You selected $num")

    if (1 <= num && num <= 5) {
        selectedNamespace = list[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $selectedNamespace")
    println(DASHES)

    println(DASHES)
    println("2. List available metrics within the selected namespace and select
 one from the list.")
```

```
    val metList = listMets(selectedNamespace)
    for (z in 0..4) {
        println("    ${ z + 1}. ${metList?.get(z)}")
    }
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        selectedMetrics = metList!![num - 1]
    } else {
        println("You did not select a valid option.")
        System.exit(1)
    }
    println("You selected $selectedMetrics")
    val myDimension = getSpecificMet(selectedNamespace)
    if (myDimension == null) {
        println("Error - Dimension is null")
        exitProcess(1)
    }
    println(DASHES)

    println(DASHES)
    println("3. Get statistics for the selected metric over the last day.")
    val metricOption: String
    val statTypes = ArrayList<String>()
    statTypes.add("SampleCount")
    statTypes.add("Average")
    statTypes.add("Sum")
    statTypes.add("Minimum")
    statTypes.add("Maximum")

    for (t in 0..4) {
        println("    ${t + 1}. ${statTypes[t]}")
    }
    println("Select a metric statistic by entering a number from the preceding
list:")
    num = inOb.nextLine().toInt()
    if (1 <= num && num <= 5) {
        metricOption = statTypes[num - 1]
    } else {
        println("You did not select a valid option.")
        exitProcess(1)
    }
    println("You selected $metricOption")
    getAndDisplayMetricStatistics(selectedNamespace, selectedMetrics,
metricOption, myDate, myDimension)
```

```
    println(DASHES)

    println(DASHES)
    println("4. Get CloudWatch estimated billing for the last week.")
    getMetricStatistics(costDateWeek)
    println(DASHES)

    println(DASHES)
    println("5. Create a new CloudWatch dashboard with metrics.")
    createDashboardWithMetrics(dashboardName, dashboardJson)
    println(DASHES)

    println(DASHES)
    println("6. List dashboards using a paginator.")
    listDashboards()
    println(DASHES)

    println(DASHES)
    println("7. Create a new custom metric by adding data to it.")
    createNewCustomMetric(dataPoint)
    println(DASHES)

    println(DASHES)
    println("8. Add an additional metric to the dashboard.")
    addMetricToDashboard(dashboardAdd, dashboardName)
    println(DASHES)

    println(DASHES)
    println("9. Create an alarm for the custom metric.")
    val alarmName: String = createAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("10. Describe 10 current alarms.")
    describeAlarms()
    println(DASHES)

    println(DASHES)
    println("11. Get current data for the new custom metric.")
    getCustomMetricData(settings)
    println(DASHES)

    println(DASHES)
    println("12. Push data into the custom metric to trigger the alarm.")
```

```
    addMetricDataForAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("13. Check the alarm state using the action
 DescribeAlarmsForMetric.")
    checkForMetricAlarm(settings)
    println(DASHES)

    println(DASHES)
    println("14. Get alarm history for the new alarm.")
    getAlarmHistory(settings, myDate)
    println(DASHES)

    println(DASHES)
    println("15. Add an anomaly detector for the custom metric.")
    addAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("16. Describe current anomaly detectors.")
    describeAnomalyDetectors(settings)
    println(DASHES)

    println(DASHES)
    println("17. Get a metric image for the custom metric.")
    getAndOpenMetricImage(metricImage)
    println(DASHES)

    println(DASHES)
    println("18. Clean up the Amazon CloudWatch resources.")
    deleteDashboard(dashboardName)
    deleteAlarm(alarmName)
    deleteAnomalyDetector(settings)
    println(DASHES)

    println(DASHES)
    println("The Amazon CloudWatch example scenario is complete.")
    println(DASHES)
}

suspend fun deleteAnomalyDetector(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
```

```
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val request = DeleteAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAnomalyDetector(request)
        println("Successfully deleted the Anomaly Detector.")
    }
}

suspend fun deleteAlarm(alarmNameVal: String) {
    val request = DeleteAlarmsRequest {
        alarmNames = listOf(alarmNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteAlarms(request)
        println("Successfully deleted alarm $alarmNameVal")
    }
}

suspend fun deleteDashboard(dashboardName: String) {
    val dashboardsRequest = DeleteDashboardsRequest {
        dashboardNames = listOf(dashboardName)
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.deleteDashboards(dashboardsRequest)
        println("$dashboardName was successfully deleted.")
    }
}

suspend fun getAndOpenMetricImage(fileName: String) {
    println("Getting Image data for custom metric.")
```

```
    val myJSON = """{
        "title": "Example Metric Graph",
        "view": "timeSeries",
        "stacked ": false,
        "period": 10,
        "width": 1400,
        "height": 600,
        "metrics": [
            [
            "AWS/Billing",
            "EstimatedCharges",
            "Currency",
            "USD"
            ]
        ]
        }"""

    val imageRequest = GetMetricWidgetImageRequest {
        metricWidget = myJSON
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricWidgetImage(imageRequest)
        val bytes = response.metricWidgetImage
        if (bytes != null) {
            File(fileName).writeBytes(bytes)
        }
    }
    println("You have successfully written data to $fileName")
}

suspend fun describeAnomalyDetectors(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val detectorsRequest = DescribeAnomalyDetectorsRequest {
        maxResults = 10
        metricName = customMetricName
        namespace = customMetricNamespace
    }
```

```
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAnomalyDetectors(detectorsRequest)
        response.anomalyDetectors?.forEach { detector ->
            println("Metric name:
 ${detector.singleMetricAnomalyDetector?.metricName}")
            println("State: ${detector.stateValue}")
        }
    }
}

suspend fun addAnomalyDetector(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    val singleMetricAnomalyDetectorVal = SingleMetricAnomalyDetector {
        metricName = customMetricName
        namespace = customMetricNamespace
        stat = "Maximum"
    }

    val anomalyDetectorRequest = PutAnomalyDetectorRequest {
        singleMetricAnomalyDetector = singleMetricAnomalyDetectorVal
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putAnomalyDetector(anomalyDetectorRequest)
        println("Added anomaly detector for metric $customMetricName.")
    }
}

suspend fun getAlarmHistory(fileName: String, date: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val start = Instant.parse(date)
    val endDateVal = Instant.now()

    val historyRequest = DescribeAlarmHistoryRequest {
        startDate = aws.smithy.kotlin.runtime.time.Instant(start)
```

```
        endDate = aws.smithy.kotlin.runtime.time.Instant(endDateVal)
        alarmName = alarmNameVal
        historyItemType = HistoryItemType.Action
    }

    CloudWatchClient { credentialsProvider = EnvironmentCredentialsProvider();
 region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarmHistory(historyRequest)
        val historyItems = response.alarmHistoryItems
        if (historyItems != null) {
            if (historyItems.isEmpty()) {
                println("No alarm history data found for $alarmNameVal.")
            } else {
                for (item in historyItems) {
                    println("History summary ${item.historySummary}")
                    println("Time stamp: ${item.timestamp}")
                }
            }
        }
    }
}

suspend fun checkForMetricAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    var hasAlarm = false
    var retries = 10

    val metricRequest = DescribeAlarmsForMetricRequest {
        metricName = customMetricName
        namespace = customMetricNamespace
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        while (!hasAlarm && retries > 0) {
            val response = cwClient.describeAlarmsForMetric(metricRequest)
            if (response.metricAlarms?.count()!! > 0) {
                hasAlarm = true
            }
            retries--
            delay(20000)
```

```
            println(".")
        }
        if (!hasAlarm) println("No Alarm state found for $customMetricName after
 10 retries.") else println("Alarm state found for $customMetricName.")
    }
}

suspend fun addMetricDataForAlarm(fileName: String?) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1001.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val datum2 = MetricDatum {
        metricName = customMetricName
        unit = StandardUnit.None
        value = 1002.00
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
    }

    val metricDataList = ArrayList<MetricDatum>()
    metricDataList.add(datum)
    metricDataList.add(datum2)

    val request = PutMetricDataRequest {
        namespace = customMetricNamespace
        metricData = metricDataList
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
```

```
        println("Added metric values for for metric $customMetricName")
    }
}

suspend fun getCustomMetricData(fileName: String) {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode = ObjectMapper().readTree<JsonNode>(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()

    // Set the date.
    val nowDate = Instant.now()
    val hours: Long = 1
    val minutes: Long = 30
    val date2 = nowDate.plus(hours, ChronoUnit.HOURS).plus(
        minutes,
        ChronoUnit.MINUTES
    )

    val met = Metric {
        metricName = customMetricName
        namespace = customMetricNamespace
    }

    val metStat = MetricStat {
        stat = "Maximum"
        period = 1
        metric = met
    }

    val dataQUery = MetricDataQuery {
        metricStat = metStat
        id = "foo2"
        returnData = true
    }

    val dq = ArrayList<MetricDataQuery>()
    dq.add(dataQUery)
    val getMetReq = GetMetricDataRequest {
        maxDatapoints = 10
        scanBy = ScanBy.TimestampDescending
        startTime = aws.smithy.kotlin.runtime.time.Instant(nowDate)
```

```
        endTime = aws.smithy.kotlin.runtime.time.Instant(date2)
        metricDataQueries = dq
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricData(getMetReq)
        response.metricDataResults?.forEach { item ->
            println("The label is ${item.label}")
            println("The status code is ${item.statusCode}")
        }
    }
}

suspend fun describeAlarms() {
    val typeList = ArrayList<AlarmType>()
    typeList.add(AlarmType.MetricAlarm)
    val alarmsRequest = DescribeAlarmsRequest {
        alarmTypes = typeList
        maxRecords = 10
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.describeAlarms(alarmsRequest)
        response.metricAlarms?.forEach { alarm ->
            println("Alarm name: ${alarm.alarmName}")
            println("Alarm description: ${alarm.alarmDescription}")
        }
    }
}

suspend fun createAlarm(fileName: String): String {
    // Read values from the JSON file.
    val parser = JsonFactory().createParser(File(fileName))
    val rootNode: JsonNode = ObjectMapper().readTree(parser)
    val customMetricNamespace =
 rootNode.findValue("customMetricNamespace").asText()
    val customMetricName = rootNode.findValue("customMetricName").asText()
    val alarmNameVal = rootNode.findValue("exampleAlarmName").asText()
    val emailTopic = rootNode.findValue("emailTopic").asText()
    val accountId = rootNode.findValue("accountId").asText()
    val region2 = rootNode.findValue("region").asText()

    // Create a List for alarm actions.
    val alarmActionObs: MutableList<String> = ArrayList()
```

```
    alarmActionObs.add("arn:aws:sns:$region2:$accountId:$emailTopic")
    val alarmRequest = PutMetricAlarmRequest {
        alarmActions = alarmActionObs
        alarmDescription = "Example metric alarm"
        alarmName = alarmNameVal
        comparisonOperator = ComparisonOperator.GreaterThanOrEqualToThreshold
        threshold = 100.00
        metricName = customMetricName
        namespace = customMetricNamespace
        evaluationPeriods = 1
        period = 10
        statistic = Statistic.Maximum
        datapointsToAlarm = 1
        treatMissingData = "ignore"
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricAlarm(alarmRequest)
        println("$alarmNameVal was successfully created!")
        return alarmNameVal
    }
}

suspend fun addMetricToDashboard(fileNameVal: String, dashboardNameVal: String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully updated.")
    }
}

suspend fun createNewCustomMetric(dataPoint: Double) {
    val dimension = Dimension {
        name = "UNIQUE_PAGES"
        value = "URLS"
    }

    // Set an Instant object.
    val time =
 ZonedDateTime.now(ZoneOffset.UTC).format(DateTimeFormatter.ISO_INSTANT)
```

```
    val instant = Instant.parse(time)
    val datum = MetricDatum {
        metricName = "PAGES_VISITED"
        unit = StandardUnit.None
        value = dataPoint
        timestamp = aws.smithy.kotlin.runtime.time.Instant(instant)
        dimensions = listOf(dimension)
    }

    val request = PutMetricDataRequest {
        namespace = "SITE/TRAFFIC"
        metricData = listOf(datum)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.putMetricData(request)
        println("Added metric values for for metric PAGES_VISITED")
    }
}

suspend fun listDashboards() {
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        cwClient.listDashboardsPaginated({})
            .transform { it.dashboardEntries?.forEach { obj -> emit(obj) } }
            .collect { obj ->
                println("Name is ${obj.dashboardName}")
                println("Dashboard ARN is ${obj.dashboardArn}")
            }
    }
}

suspend fun createDashboardWithMetrics(dashboardNameVal: String, fileNameVal:
 String) {
    val dashboardRequest = PutDashboardRequest {
        dashboardName = dashboardNameVal
        dashboardBody = readFileAsString(fileNameVal)
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.putDashboard(dashboardRequest)
        println("$dashboardNameVal was successfully created.")
        val messages = response.dashboardValidationMessages
        if (messages != null) {
            if (messages.isEmpty()) {
```

```
                println("There are no messages in the new Dashboard")
            } else {
                for (message in messages) {
                    println("Message is: ${message.message}")
                }
            }
        }
    }
}

fun readFileAsString(file: String): String {
    return String(Files.readAllBytes(Paths.get(file)))
}

suspend fun getMetricStatistics(costDateWeek: String?) {
    val start = Instant.parse(costDateWeek)
    val endDate = Instant.now()
    val dimension = Dimension {
        name = "Currency"
        value = "USD"
    }

    val dimensionList: MutableList<Dimension> = ArrayList()
    dimensionList.add(dimension)

    val statisticsRequest = GetMetricStatisticsRequest {
        metricName = "EstimatedCharges"
        namespace = "AWS/Billing"
        dimensions = dimensionList
        statistics = listOf(Statistic.Maximum)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        period = 86400
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data: List<Datapoint>? = response.datapoints
        if (data != null) {
            if (!data.isEmpty()) {
                for (datapoint in data) {
                    println("Timestamp:  ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
```

```
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun getAndDisplayMetricStatistics(nameSpaceVal: String, metVal: String,
 metricOption: String, date: String, myDimension: Dimension) {
    val start = Instant.parse(date)
    val endDate = Instant.now()
    val statisticsRequest = GetMetricStatisticsRequest {
        endTime = aws.smithy.kotlin.runtime.time.Instant(endDate)
        startTime = aws.smithy.kotlin.runtime.time.Instant(start)
        dimensions = listOf(myDimension)
        metricName = metVal
        namespace = nameSpaceVal
        period = 86400
        statistics = listOf(Statistic.fromValue(metricOption))
    }

    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.getMetricStatistics(statisticsRequest)
        val data = response.datapoints
        if (data != null) {
            if (data.isNotEmpty()) {
                for (datapoint in data) {
                    println("Timestamp: ${datapoint.timestamp} Maximum value:
 ${datapoint.maximum}")
                }
            } else {
                println("The returned data list is empty")
            }
        }
    }
}

suspend fun listMets(namespaceVal: String?): ArrayList<String>? {
    val metList = ArrayList<String>()
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val reponse = cwClient.listMetrics(request)
        reponse.metrics?.forEach { metrics ->
```

```
            val data = metrics.metricName
            if (!metList.contains(data)) {
                metList.add(data!!)
            }
        }
    }
    return metList
}

suspend fun getSpecificMet(namespaceVal: String?): Dimension? {
    val request = ListMetricsRequest {
        namespace = namespaceVal
    }
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(request)
        val myList = response.metrics
        if (myList != null) {
            return myList[0].dimensions?.get(0)
        }
    }
    return null
}

suspend fun listNameSpaces(): ArrayList<String> {
    val nameSpaceList = ArrayList<String>()
    CloudWatchClient { region = "us-east-1" }.use { cwClient ->
        val response = cwClient.listMetrics(ListMetricsRequest {})
        response.metrics?.forEach { metrics ->
            val data = metrics.namespace
            if (!nameSpaceList.contains(data)) {
                nameSpaceList.add(data!!)
            }
        }
    }
    return nameSpaceList
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Kotlin API reference.
  - DeleteAlarms
  - DeleteAnomalyDetector

- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- GetMetricData
- GetMetricStatistics
- GetMetricWidgetImage
- ListMetrics
- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- PutMetricData

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Gérez les CloudWatch métriques et les alarmes à l'aide d'un AWS SDK

L’exemple de code suivant illustre comment :

- Créez une alarme pour surveiller une CloudWatch métrique.
- placer les données dans une métrique et déclencher l'alerte ;
- récupérer les données de l'alerte ;
- supprimer l'alerte.

Python

SDK pour Python (Boto3)

> **Note**
>
> Il y en a plus sur GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une classe qui englobe les CloudWatch opérations.

```
from datetime import datetime, timedelta
import logging
from pprint import pprint
import random
import time
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)


class CloudWatchWrapper:
    """Encapsulates Amazon CloudWatch functions."""

    def __init__(self, cloudwatch_resource):
        """
        :param cloudwatch_resource: A Boto3 CloudWatch resource.
        """
        self.cloudwatch_resource = cloudwatch_resource


    def put_metric_data_set(self, namespace, name, timestamp, unit, data_set):
        """
        Sends a set of data to CloudWatch for a metric. All of the data in the
 set
        have the same timestamp and unit.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param timestamp: The UTC timestamp for the metric.
        :param unit: The unit of the metric.
```

```
        :param data_set: The set of data to send. This set is a dictionary that
                         contains a list of values and a list of corresponding
counts.
                         The value and count lists must be the same length.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[
                    {
                        "MetricName": name,
                        "Timestamp": timestamp,
                        "Values": data_set["values"],
                        "Counts": data_set["counts"],
                        "Unit": unit,
                    }
                ],
            )
            logger.info("Put data set for metric %s.%s.", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data set for metric %s.%s.",
namespace, name)
            raise


    def create_metric_alarm(
        self,
        metric_namespace,
        metric_name,
        alarm_name,
        stat_type,
        period,
        eval_periods,
        threshold,
        comparison_op,
    ):
        """
        Creates an alarm that watches a metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :param alarm_name: The name of the alarm.
        :param stat_type: The type of statistic the alarm watches.
```

```
        :param period: The period in which metric data are grouped to calculate
                       statistics.
        :param eval_periods: The number of periods that the metric must be over
the
                             alarm threshold before the alarm is set into an
alarmed
                             state.
        :param threshold: The threshold value to compare against the metric
statistic.
        :param comparison_op: The comparison operation used to compare the
threshold
                              against the metric.
        :return: The newly created alarm.
        """
        try:
            metric = self.cloudwatch_resource.Metric(metric_namespace,
metric_name)
            alarm = metric.put_alarm(
                AlarmName=alarm_name,
                Statistic=stat_type,
                Period=period,
                EvaluationPeriods=eval_periods,
                Threshold=threshold,
                ComparisonOperator=comparison_op,
            )
            logger.info(
                "Added alarm %s to track metric %s.%s.",
                alarm_name,
                metric_namespace,
                metric_name,
            )
        except ClientError:
            logger.exception(
                "Couldn't add alarm %s to metric %s.%s",
                alarm_name,
                metric_namespace,
                metric_name,
            )
            raise
        else:
            return alarm


    def put_metric_data(self, namespace, name, value, unit):
```

```
        """
        Sends a single data value to CloudWatch for a metric. This metric is
given
        a timestamp of the current UTC time.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param value: The value of the metric.
        :param unit: The unit of the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            metric.put_data(
                Namespace=namespace,
                MetricData=[{"MetricName": name, "Value": value, "Unit": unit}],
            )
            logger.info("Put data for metric %s.%s", namespace, name)
        except ClientError:
            logger.exception("Couldn't put data for metric %s.%s", namespace,
name)
            raise


    def get_metric_statistics(self, namespace, name, start, end, period,
stat_types):
        """
        Gets statistics for a metric within a specified time span. Metrics are
grouped
        into the specified period.

        :param namespace: The namespace of the metric.
        :param name: The name of the metric.
        :param start: The UTC start time of the time span to retrieve.
        :param end: The UTC end time of the time span to retrieve.
        :param period: The period, in seconds, in which to group metrics. The
period
                       must match the granularity of the metric, which depends on
                       the metric's age. For example, metrics that are older than
                       three hours have a one-minute granularity, so the period
must
                       be at least 60 and must be a multiple of 60.
        :param stat_types: The type of statistics to retrieve, such as average
value
                           or maximum value.
```

```
        :return: The retrieved statistics for the metric.
        """
        try:
            metric = self.cloudwatch_resource.Metric(namespace, name)
            stats = metric.get_statistics(
                StartTime=start, EndTime=end, Period=period,
Statistics=stat_types
            )
            logger.info(
                "Got %s statistics for %s.", len(stats["Datapoints"]),
stats["Label"]
            )
        except ClientError:
            logger.exception("Couldn't get statistics for %s.%s.", namespace,
name)
            raise
        else:
            return stats


    def get_metric_alarms(self, metric_namespace, metric_name):
        """
        Gets the alarms that are currently watching the specified metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        :returns: An iterator that yields the alarms.
        """
        metric = self.cloudwatch_resource.Metric(metric_namespace, metric_name)
        alarm_iter = metric.alarms.all()
        logger.info("Got alarms for metric %s.%s.", metric_namespace,
metric_name)
        return alarm_iter


    def delete_metric_alarms(self, metric_namespace, metric_name):
        """
        Deletes all of the alarms that are currently watching the specified
metric.

        :param metric_namespace: The namespace of the metric.
        :param metric_name: The name of the metric.
        """
        try:
```

```
            metric = self.cloudwatch_resource.Metric(metric_namespace,
 metric_name)
            metric.alarms.delete()
            logger.info(
                "Deleted alarms for metric %s.%s.", metric_namespace, metric_name
            )
        except ClientError:
            logger.exception(
                "Couldn't delete alarms for metric %s.%s.",
                metric_namespace,
                metric_name,
            )
            raise
```

Utilisez la classe wrapper pour placer des données dans une métrique, déclenchez une alerte qui surveille la métrique et récupérez les données de l'alerte.

```
def usage_demo():
    print("-" * 88)
    print("Welcome to the Amazon CloudWatch metrics and alarms demo!")
    print("-" * 88)

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    cw_wrapper = CloudWatchWrapper(boto3.resource("cloudwatch"))

    minutes = 20
    metric_namespace = "doc-example-metric"
    metric_name = "page_views"
    start = datetime.utcnow() - timedelta(minutes=minutes)
    print(
        f"Putting data into metric {metric_namespace}.{metric_name} spanning the
 "
        f"last {minutes} minutes."
    )
    for offset in range(0, minutes):
        stamp = start + timedelta(minutes=offset)
        cw_wrapper.put_metric_data_set(
            metric_namespace,
            metric_name,
```

```
            stamp,
            "Count",
            {
                "values": [
                    random.randint(bound, bound * 2)
                    for bound in range(offset + 1, offset + 11)
                ],
                "counts": [random.randint(1, offset + 1) for _ in range(10)],
            },
        )

    alarm_name = "high_page_views"
    period = 60
    eval_periods = 2
    print(f"Creating alarm {alarm_name} for metric {metric_name}.")
    alarm = cw_wrapper.create_metric_alarm(
        metric_namespace,
        metric_name,
        alarm_name,
        "Maximum",
        period,
        eval_periods,
        100,
        "GreaterThanThreshold",
    )
    print(f"Alarm ARN is {alarm.alarm_arn}.")
    print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Sending data to trigger the alarm. This requires data over the
threshold "
        f"for {eval_periods} periods of {period} seconds each."
    )
    while alarm.state_value == "INSUFFICIENT_DATA":
        print("Sending data for the metric.")
        cw_wrapper.put_metric_data(
            metric_namespace, metric_name, random.randint(100, 200), "Count"
        )
        alarm.load()
        print(f"Current alarm state is: {alarm.state_value}.")
        if alarm.state_value == "INSUFFICIENT_DATA":
            print(f"Waiting for {period} seconds...")
            time.sleep(period)
        else:
```

```
            print("Wait for a minute for eventual consistency of metric data.")
            time.sleep(period)
            if alarm.state_value == "OK":
                alarm.load()
                print(f"Current alarm state is: {alarm.state_value}.")

    print(
        f"Getting data for metric {metric_namespace}.{metric_name} during
timespan "
        f"of {start} to {datetime.utcnow()} (times are UTC)."
    )
    stats = cw_wrapper.get_metric_statistics(
        metric_namespace,
        metric_name,
        start,
        datetime.utcnow(),
        60,
        ["Average", "Minimum", "Maximum"],
    )
    print(
        f"Got {len(stats['Datapoints'])} data points for metric "
        f"{metric_namespace}.{metric_name}."
    )
    pprint(sorted(stats["Datapoints"], key=lambda x: x["Timestamp"]))

    print(f"Getting alarms for metric {metric_name}.")
    alarms = cw_wrapper.get_metric_alarms(metric_namespace, metric_name)
    for alarm in alarms:
        print(f"Alarm {alarm.name} is currently in state {alarm.state_value}.")

    print(f"Deleting alarms for metric {metric_name}.")
    cw_wrapper.delete_metric_alarms(metric_namespace, metric_name)

    print("Thanks for watching!")
    print("-" * 88)
```

- Pour plus d’informations sur l’API, consultez les rubriques suivantes dans AWS SDK for Python (Boto3) API Reference.
  - DeleteAlarms

- DescribeAlarmsForMetric
- DisableAlarmActions
- EnableAlarmActions
- GetMetricStatistics
- ListMetrics
- PutMetricAlarm
- PutMetricData

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples multiservices d' CloudWatch utilisation des SDK AWS

Les exemples d'applications suivants utilisent AWS des SDK à combiner CloudWatch avec d'autres Services AWS. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter l'application.

Exemples

- Surveillez les performances d'Amazon DynamoDB à l'aide d'un SDK AWS

Surveillez les performances d'Amazon DynamoDB à l'aide d'un SDK AWS

L'exemple de code suivant montre comment configurer l'utilisation de DynamoDB par une application pour surveiller les performances.

Java

SDK pour Java 2.x

Cet exemple montre comment configurer une application Java pour surveiller les performances de DynamoDB. L'application envoie des données métriques vers CloudWatch lesquelles vous pouvez surveiller les performances.

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
- DynamoDB

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation CloudWatch avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Sécurité sur Amazon CloudWatch

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 CloudWatch, 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. Il vous explique comment configurer Amazon CloudWatch pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos CloudWatch ressources.

Table des matières

Protection des données sur Amazon CloudWatch

Le modèle de responsabilité AWS partagée de s'applique à la protection des données sur Amazon CloudWatch. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- 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 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

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 Name (Nom). Cela inclut lorsque vous travaillez avec CloudWatch ou d'autres Services AWS utilisateurs de la console, de l'API ou AWS des SDK. AWS CLI 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 en transit

CloudWatch utilise end-to-end le chiffrement des données en transit.

Gestion des identités et des accès pour Amazon CloudWatch

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser CloudWatch les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Utilisation de rôles liés à un service pour Application Insights CloudWatch
- AWS politiques gérées pour Amazon CloudWatch Application Insights
- Référence CloudWatch des autorisations Amazon

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez. CloudWatch

Utilisateur du service : si vous utilisez le CloudWatch service pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de nouvelles CloudWatch fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans CloudWatch, consultezRésolution des problèmes d' CloudWatch identité et d'accès à Amazon.

Administrateur du service — Si vous êtes responsable des CloudWatch ressources de votre entreprise, vous avez probablement un accès complet à CloudWatch. C'est à vous de déterminer les CloudWatch fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec CloudWatch, voirComment Amazon CloudWatch travaille avec IAM.

Administrateur IAM — Si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à. CloudWatch Pour consulter des exemples de politiques CloudWatch basées sur l'identité que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour Amazon CloudWatch

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs

(IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section Signature des demandes AWS d'API dans le guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez Tâches nécessitant des informations d'identification d'utilisateur root dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent

des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.

- Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur la politique relative à la transmission de demandes FAS, consultez Sessions de transmission d'accès.
- Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés au service apparaissent dans votre Compte AWS fichier et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal

(utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de

confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principals (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.

- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans. AWS Organizations AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations .
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez politiques de séance dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment Amazon CloudWatch travaille avec IAM

Avant d'utiliser IAM pour gérer l'accès à CloudWatch, découvrez les fonctionnalités IAM disponibles. CloudWatch

Fonctionnalités IAM que vous pouvez utiliser avec Amazon CloudWatch

Fonction IAM	CloudWatch soutien
Politiques basées sur l'identité	Oui
Politiques basées sur les ressources	Non

Fonction IAM	CloudWatch soutien
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique (spécifiques au service)	Oui
ACL	Non
ABAC (identifications dans les politiques)	Partielle
Informations d'identification temporaires	Oui
Autorisations de principal	Oui
Fonctions de service	Oui
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont CloudWatch les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour CloudWatch

Prend en charge les politiques basées sur l'identité	Oui

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Création de politiques IAM dans le Guide de l'utilisateur IAM.

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou

refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité car celle-ci s'applique à l'utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Exemples de politiques basées sur l'identité pour CloudWatch

Pour consulter des exemples de politiques CloudWatch basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon CloudWatch

Politiques basées sur les ressources au sein de CloudWatch

Prend en charge les politiques basées sur les ressources	Non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L'ajout d'un principal entre comptes à une politique basée sur les ressources ne représente qu'une partie de l'instauration de la relation d'approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l'entité. Toutefois, si une politique basée sur des ressources accorde l'accès à un principal dans le même compte, aucune autre politique basée sur l'identité n'est requise. Pour plus d'informations, consultez Différence entre les rôles IAM et les politiques basées sur une ressource dans le Guide de l'utilisateur IAM.

Actions politiques pour CloudWatch

Prend en charge les actions de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des CloudWatch actions, consultez la section Actions définies par Amazon CloudWatch dans le Service Authorization Reference.

Les actions de politique en CloudWatch cours utilisent le préfixe suivant avant l'action :

```
cloudwatch
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "cloudwatch:action1",
      "cloudwatch:action2"
         ]
```

Pour consulter des exemples de politiques CloudWatch basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon CloudWatch

Ressources politiques pour CloudWatch

Prend en charge les ressources de politique	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Pour consulter la liste des types de CloudWatch ressources et leurs ARN, consultez la section Ressources définies par Amazon CloudWatch dans le Service Authorization Reference. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par Amazon CloudWatch.

Pour consulter des exemples de politiques CloudWatch basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon CloudWatch

Clés de conditions de politique pour CloudWatch

Prend en charge les clés de condition de politique spécifiques au service	Oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de CloudWatch condition, consultez la section Clés de condition pour Amazon CloudWatch dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par Amazon CloudWatch.

Pour consulter des exemples de politiques CloudWatch basées sur l'identité, consultez. Exemples de politiques basées sur l'identité pour Amazon CloudWatch

ACL dans CloudWatch

Prend en charge les listes ACL	Non

Les listes de contrôle d'accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

ABAC avec CloudWatch

Prise en charge d'ABAC (identifications dans les politiques)	Partielle

Le contrôle d'accès basé sur les attributs (ABAC) est une politique d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur l'ABAC, consultez Qu'est-ce que le contrôle d'accès basé sur les attributs (ABAC) ? dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

Utilisation d'informations d'identification temporaires avec CloudWatch

Prend en charge les informations d'identification temporaires	Oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Changement de rôle (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour CloudWatch

Prend en charge les sessions d'accès direct (FAS)	Oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Fonctions du service pour CloudWatch

Prend en charge les fonctions du service	Oui

Une fonction de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM.

Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber CloudWatch les fonctionnalités. Modifiez les rôles de service uniquement lorsque CloudWatch vous recevez des instructions à cet effet.

Exemples de politiques basées sur l'identité pour Amazon CloudWatch

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier CloudWatch des ressources. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, consultez Création de politiques dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par CloudWatch, y compris le format des ARN pour chacun des types de ressources, consultez la section Actions, ressources et clés de condition pour Amazon CloudWatch dans le Service Authorization Reference.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la console CloudWatch

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer CloudWatch des ressources dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politique IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la console CloudWatch

Pour accéder à la CloudWatch console Amazon, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails CloudWatch des ressources de votre Compte AWS. Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la CloudWatch console, associez également la politique CloudWatch `ConsoleAccess` ou la politique `ReadOnly` AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

Autorisations nécessaires pour CloudWatch la console

L'ensemble complet des autorisations requises pour utiliser la CloudWatch console est répertorié ci-dessous. Ces autorisations fournissent un accès complet en écriture et en lecture à la CloudWatch console.

- mise à l'échelle automatique des applications : DescribeScalingPolicies
- mise à l'échelle automatique : DescribeAutoScalingGroups
- mise à l'échelle automatique : DescribePolicies
- traînée nuageuse : DescribeTrails
- surveillance des nuages : DeleteAlarms
- surveillance des nuages : DescribeAlarmHistory
- surveillance des nuages : DescribeAlarms
- surveillance des nuages : GetMetricData
- surveillance des nuages : GetMetricStatistics
- surveillance des nuages : ListMetrics
- surveillance des nuages : PutMetricAlarm
- surveillance des nuages : PutMetricData
- EC2 : DescribeInstances

- EC2 : DescribeTags
- EC2 : DescribeVolumes
- Oui : DescribeElasticsearchDomain
- Oui : ListDomainNames
- événements : DeleteRule
- événements : DescribeRule
- événements : DisableRule
- événements : EnableRule
- événements : ListRules
- événements : PutRule
- iam : AttachRolePolicy
- iam : CreateRole
- iam : GetPolicy
- iam : GetPolicyVersion
- iam : GetRole
- iam : ListAttachedRolePolicies
- iam : ListRoles
- kinésie : DescribeStream
- kinésie : ListStreams
- lambda : AddPermission
- lambda : CreateFunction
- lambda : GetFunctionConfiguration
- lambda : ListAliases
- lambda : ListFunctions
- lambda : ListVersionsByFunction
- lambda : RemovePermission
- journaux : CancelExportTask
- journaux : CreateExportTask
- journaux : CreateLogGroup

- journaux : CreateLogStream
- journaux : DeleteLogGroup
- journaux : DeleteLogStream
- journaux : DeleteMetricFilter
- journaux : DeleteRetentionPolicy
- journaux : DeleteSubscriptionFilter
- journaux : DescribeExportTasks
- journaux : DescribeLogGroups
- journaux : DescribeLogStreams
- journaux : DescribeMetricFilters
- journaux : DescribeQueries
- journaux : DescribeSubscriptionFilters
- journaux : FilterLogEvents
- journaux : GetLogGroupFields
- journaux : GetLogRecord
- journaux : GetLogEvents
- journaux : GetQueryResults
- journaux : PutMetricFilter
- journaux : PutRetentionPolicy
- journaux : PutSubscriptionFilter
- journaux : StartQuery
- journaux : StopQuery
- journaux : TestMetricFilter
- s3 : CreateBucket
- s3 : ListBucket
- sns : CreateTopic
- sns : GetTopicAttributes
- sns : ListSubscriptions
- sns : ListTopics
- sns : SetTopicAttributes

- sns:Subscribe
- sns:Unsubscribe
- sqs : GetQueueAttributes
- sqs : GetQueueUrl
- sqs : ListQueues
- sqs : SetQueueAttributes
- swf : CreateAction
- swf : DescribeAction
- swf : ListActionTemplates
- swf : RegisterAction
- swf : RegisterDomain
- swf : UpdateAction

Pour visualiser la Carte de suivi X-Ray, `AWSXrayReadOnlyAccess` est également requis.

Résolution des problèmes d' CloudWatch identité et d'accès à Amazon

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec CloudWatch IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans CloudWatch
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes CloudWatch ressources

Je ne suis pas autorisé à effectuer une action dans CloudWatch

Si vous recevez une erreur qui indique que vous n'êtes pas autorisé à effectuer une action, vos politiques doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit quand l'utilisateur IAM `mateojackson` tente d'utiliser la console pour afficher des informations détaillées sur une ressource *`my-example-widget`* fictive, mais ne dispose pas des autorisations `cloudwatch:`*`GetWidget`* fictives.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 cloudwatch:GetWidget on resource: my-example-widget
```

Dans ce cas, la politique qui s'applique à l'utilisateur `mateojackson` doit être mise à jour pour autoriser l'accès à la ressource *`my-example-widget`* à l'aide de l'action `cloudwatch:`*`GetWidget`*.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transmettre un rôle CloudWatch.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour effectuer une action dans CloudWatch. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes CloudWatch ressources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques

basées sur les ressources ou les listes de contrôle d’accès (ACL), vous pouvez utiliser ces politiques pour donner l’accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si ces fonctionnalités sont prises CloudWatch en charge, consultezComment Amazon CloudWatch travaille avec IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d’identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d’identité) dans le Guide de l’utilisateur IAM.
- Pour découvrir quelle est la différence entre l’utilisation des rôles et l’utilisation des politiques basées sur les ressources pour l’accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l’utilisateur IAM.

CloudWatch mise à jour des autorisations du tableau

Le 1er mai 2018, les autorisations requises pour accéder aux CloudWatch tableaux de bord AWS ont été modifiées. L'accès au tableau de bord dans la CloudWatch console nécessite désormais des autorisations introduites en 2017 pour prendre en charge les opérations de l'API du tableau de bord :

- surveillance des nuages : GetDashboard
- surveillance des nuages : ListDashboards
- surveillance des nuages : PutDashboard
- surveillance des nuages : DeleteDashboards

Pour accéder aux CloudWatch tableaux de bord, vous avez besoin de l'un des éléments suivants :

- La AdministratorAccesspolitique.
- La CloudWatchFullAccesspolitique.
- Une politique personnalisée qui inclut une ou plusieurs de ces autorisations spécifiques :

- `cloudwatch:GetDashboard` et `cloudwatch:ListDashboards` pour pouvoir afficher des tableaux de bord
- `cloudwatch:PutDashboard` pour pouvoir créer ou modifier des tableaux de bord
- `cloudwatch:DeleteDashboards` pour pouvoir supprimer des tableaux de bord

Pour plus d'informations la modification des autorisations d'un utilisateur IAM à l'aide de politiques, consultez Modification des autorisations pour un utilisateur IAM.

Pour plus d'informations sur CloudWatch les autorisations, consultezRéférence CloudWatch des autorisations Amazon.

Pour plus d'informations sur les opérations de l'API du tableau de bord, consultez PutDashboardle Amazon CloudWatch API Reference.

AWS politiques gérées (prédéfinies) pour CloudWatch

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par. AWS Ces politiques AWS gérées accordent les autorisations nécessaires pour les cas d'utilisation courants afin que vous puissiez éviter d'avoir à rechercher les autorisations nécessaires. Pour plus d’informations, consultez la section Politiques gérées par AWS dans le Guide de l’utilisateur IAM.

Les politiques AWS gérées suivantes, que vous pouvez associer aux utilisateurs de votre compte, sont spécifiques à CloudWatch.

Rubriques

- CloudWatchFullAccessV2
- CloudWatchFullAccess
- CloudWatchReadOnlyAccess
- CloudWatchActionsAccès EC2
- CloudWatchAutomaticDashboardsAccess
- CloudWatchAgentServerPolicy
- CloudWatchAgentAdminPolicy
- AWS politiques gérées (prédéfinies) pour l' CloudWatch observabilité entre comptes
- AWS politiques gérées (prédéfinies) pour CloudWatch Synthetics
- AWS politiques gérées (prédéfinies) pour Amazon CloudWatch RUM

- AWS politiques gérées (prédéfinies) pour CloudWatch Evidently
- AWS politique gérée pour AWS Systems Manager Incident Manager

CloudWatchFullAccessV2

AWS a récemment ajouté la politique IAM gérée par la CloudWatchFullAccessV2. Cette politique accorde un accès complet aux CloudWatch actions et aux ressources et définit également de manière plus appropriée les autorisations accordées pour d'autres services tels qu'Amazon Amazon EC2 Auto Scaling SNS et. Nous vous recommandons de commencer à utiliser cette politique plutôt que d'utiliser CloudWatchFullAccess. AWS prévoit de devenir obsolète CloudWatchFullAccessdans un futur proche.

Il inclut `application-signals:` des autorisations permettant aux utilisateurs d'accéder à toutes les fonctionnalités depuis la CloudWatch console sous Application Signals. Elle inclut certaines `autoscaling:Describe` autorisations afin que les utilisateurs soumis à cette politique puissent voir les actions Auto Scaling associées aux CloudWatch alarmes. Elle inclut certaines `sns` autorisations afin que les utilisateurs soumis à cette politique puissent récupérer, créer des rubriques Amazon SNS et les associer CloudWatch à des alarmes. Elle inclut des autorisations IAM afin que les utilisateurs soumis à cette politique puissent consulter les informations sur les rôles liés aux services associés à. CloudWatch Elle inclut les `oam:ListAttachedLinks` autorisations `oam:ListSinks` et 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.

Il inclut `rumsynthetics`, et `xray` des autorisations permettant aux utilisateurs d'avoir un accès complet à CloudWatch Synthetics CloudWatch et RUM AWS X-Ray, qui sont tous couverts par le service. CloudWatch

Le contenu de la CloudWatchFullAccessV2 est le suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CloudWatchFullAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:*",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribePolicies",
```

```
                "cloudwatch:*",
                "logs:*",
                "sns:CreateTopic",
                "sns:ListSubscriptions",
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics",
                "sns:Subscribe",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks",
                "rum:*",
                "synthetics:*",
                "xray:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CloudWatchApplicationSignalsServiceLinkedRolePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "application-
signals.cloudwatch.amazonaws.com"
                }
            }
        },
        {
            "Sid": "EventsServicePermissions",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
        },
         {
            "Sid": "OAMReadPermissions",
```

```
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchFullAccess

La CloudWatchFullAccesspolitique est sur le point de devenir obsolète. Nous vous recommandons d'arrêter de l'utiliser et d'utiliser la CloudWatchFullAccessversion V2 à la place.

Le contenu de CloudWatchFullAccessest le suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*",
                "cloudwatch:*",
                "logs:*",
                "sns:*",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "oam:ListSinks"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/events.amazonaws.com/
AWSServiceRoleForCloudWatchEvents*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": "events.amazonaws.com"
                }
            }
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        }
    ]
}
```

CloudWatchReadOnlyAccess

La CloudWatchReadOnlyAccesspolitique accorde un accès en lecture seule à. CloudWatch

La politique inclut certaines `logs:` autorisations, de sorte que les utilisateurs dotés de cette politique peuvent utiliser la console pour consulter les informations CloudWatch des journaux et les requêtes de CloudWatch journaux Insights. Cela inclut`autoscaling:Describe*`, afin que les utilisateurs soumis à cette politique puissent voir les actions Auto Scaling associées aux CloudWatch alarmes. Il inclut les `application-signals:` autorisations permettant aux utilisateurs d'utiliser les signaux d'application pour surveiller l'état de leurs services. Elle inclut `application-autoscaling:DescribeScalingPolicies` afin que les utilisateurs dotés de cette politique puissent accéder aux informations sur les politiques Application Auto Scaling. Cela inclut `sns:Get*` et`sns:List*`, afin que les utilisateurs soumis à cette politique puissent récupérer des informations sur les rubriques Amazon SNS qui reçoivent des notifications concernant CloudWatch les alarmes. Elle inclut les `oam:ListAttachedLinks` autorisations `oam:ListSinks` et, de sorte que les utilisateurs soumis à cette politique peuvent utiliser la console pour consulter les données partagées à partir de comptes sources dans le cadre d'une CloudWatch observabilité entre comptes. Il inclut les `iam:GetRole` autorisations permettant aux utilisateurs de vérifier si les signaux CloudWatch d'application ont été configurés.

Il inclut `rumsynthetics`, et `xray` des autorisations permettant aux utilisateurs d'avoir un accès en lecture seule à Synthetics CloudWatch et CloudWatch RUM AWS X-Ray, qui sont tous couverts par le service. CloudWatch

Le contenu de la CloudWatchReadOnlyAccesspolitique est le suivant.

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

```
            "Sid": "CloudWatchReadOnlyAccessPermissions",
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DescribeScalingPolicies",
                "application-signals:BatchGet*",
                "application-signals:Get*",
                "application-signals:List*",
                "autoscaling:Describe*",
                "cloudwatch:BatchGet*",
                "cloudwatch:Describe*",
                "cloudwatch:GenerateQuery",
                "cloudwatch:Get*",
                "cloudwatch:List*",
                "logs:Get*",
                "logs:List*",
                "logs:StartQuery",
                "logs:StopQuery",
                "logs:Describe*",
                "logs:TestMetricFilter",
                "logs:FilterLogEvents",
                "logs:StartLiveTail",
                "logs:StopLiveTail",
                "oam:ListSinks",
                "sns:Get*",
                "sns:List*",
                "rum:BatchGet*",
                "rum:Get*",
                "rum:List*",
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "xray:BatchGet*",
                "xray:Get*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "OAMReadPermissions",
            "Effect": "Allow",
            "Action": [
                "oam:ListAttachedLinks"
            ],
            "Resource": "arn:aws:oam:*:*:sink/*"
        },
```

```
        {
            "Sid": "CloudWatchReadOnlyGetRolePermissions",
            "Effect": "Allow",
            "Action": "iam:GetRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/application-
signals.cloudwatch.amazonaws.com/AWSServiceRoleForCloudWatchApplicationSignals"
        }
     ]
}
```

CloudWatchActionsAccès EC2

La politique CloudWatchActionsEC2Access accorde un accès en lecture seule aux CloudWatch alarmes et aux métriques en plus des métadonnées Amazon EC2. Il accorde également l'accès pour arrêter, interrompre et réinitialiser les actions d'API pour les instances EC2.

Voici le contenu de la politique CloudWatchActionsEC2Access.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:Describe*",
        "ec2:Describe*",
        "ec2:RebootInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": "*"
    }
  ]
}
```

CloudWatchAutomaticDashboardsAccess

La CloudWatch politique CrossAccountAccess gérée est utilisée par le rôle CloudWatch-CrossAccountSharingRole IAM. Ce rôle et cette politique permettent aux utilisateurs de tableaux de bord intercomptes d'afficher des tableaux de bord automatiques dans chaque compte partageant des tableaux de bord.

Voici le contenu de CloudWatchAutomaticDashboardsAccess:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:DescribeAutoScalingGroups",
        "cloudfront:GetDistribution",
        "cloudfront:ListDistributions",
        "dynamodb:DescribeTable",
        "dynamodb:ListTables",
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "elasticache:DescribeCacheClusters",
        "elasticbeanstalk:DescribeEnvironments",
        "elasticfilesystem:DescribeFileSystems",
        "elasticloadbalancing:DescribeLoadBalancers",
        "kinesis:DescribeStream",
        "kinesis:ListStreams",
        "lambda:GetFunction",
        "lambda:ListFunctions",
        "rds:DescribeDBClusters",
        "rds:DescribeDBInstances",
        "resource-groups:ListGroupResources",
        "resource-groups:ListGroups",
        "route53:GetHealthCheck",
        "route53:ListHealthChecks",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sns:ListTopics",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "sqs:ListQueues",
        "synthetics:DescribeCanariesLastRun",
        "tag:GetResources"
      ],
      "Effect": "Allow",
      "Resource": "*"
```

```
    },
    {
      "Action": [
        "apigateway:GET"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:apigateway:*::/restapis*"
      ]
    }
  ]
```

CloudWatchAgentServerPolicy

La CloudWatchAgentServerPolicypolitique peut être utilisée dans les rôles IAM attachés aux instances Amazon EC2 pour permettre à CloudWatch l'agent de lire les informations de l'instance et de les y écrire. CloudWatch Son contenu est le suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchServerPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeVolumes",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "CWASSMServerPermissions",
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
}
```

CloudWatchAgentAdminPolicy

La CloudWatchAgentAdminPolicypolitique peut être utilisée dans les rôles IAM attachés aux instances Amazon EC2. Cette politique permet à l' CloudWatch agent de lire les informations de l'instance et de les écrire CloudWatch, ainsi que d'écrire des informations dans Parameter Store. Son contenu est le suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CWACloudWatchPermissions",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "ec2:DescribeTags",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries"
            ],
            "Resource": "*"
        },
        {
            "Sid": "CWASSMPermissions",
            "Effect": "Allow",
```

```
            "Action": [
                "ssm:GetParameter",
                "ssm:PutParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-*"
        }
    ]
}
```

> **Note**
>
> Vous pouvez consulter ces politiques d'autorisations en vous connectant à la console IAM et en y recherchant des politiques spécifiques.

Vous pouvez également créer vos propres politiques IAM personnalisées pour autoriser les CloudWatch actions et les ressources. Vous pouvez attacher ces politiques personnalisées aux utilisateurs ou groupes IAM qui nécessitent ces autorisations.

AWS politiques gérées (prédéfinies) pour l' CloudWatch observabilité entre comptes

Les politiques décrites dans cette section accordent des autorisations liées à l' CloudWatch observabilité entre comptes. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

CloudWatchCrossAccountSharingConfiguration

La CloudWatchCrossAccountSharingConfigurationpolitique autorise l'accès à la création, à la gestion et à l'affichage des liens d'Observability Access Manager pour le partage de CloudWatch ressources entre comptes. Pour plus d'informations, consultez CloudWatch observabilité entre comptes. Le contenu est le suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:Link",
                "oam:ListLinks"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:DeleteLink",
                "oam:GetLink",
                "oam:TagResource"
            ],
            "Resource": "arn:aws:oam:*:*:link/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "oam:CreateLink",
                "oam:UpdateLink"
            ],
            "Resource": [
                "arn:aws:oam:*:*:link/*",
                "arn:aws:oam:*:*:sink/*"
            ]
        }
    ]
}
```

OAM FullAccess

La FullAccess politique OAM autorise l'accès à la création, à la gestion et à l'affichage des puits et des liens d'Observability Access Manager, qui sont utilisés pour l'observabilité CloudWatch entre comptes.

La FullAccess politique OAM en elle-même ne vous permet pas de partager des données d'observabilité entre des liens. Pour créer un lien permettant de partager des CloudWatch statistiques, vous avez également besoin de l'un CloudWatchFullAccessou de l'autre CloudWatchCrossAccountSharingConfiguration. Pour créer un lien permettant de partager CloudWatch des groupes de journaux Logs, vous avez également besoin de l'un CloudWatchLogsFullAccessou de CloudWatchLogsCrossAccountSharingConfiguration. Pour créer un lien permettant de partager des traces de X-Ray, vous avez également besoin de l'un AWSXRayFullAccessou de l'autre AWSXRayCrossAccountSharingConfiguration.

Pour plus d’informations, consultez CloudWatch observabilité entre comptes. Le contenu est le suivant :

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

OAM ReadOnlyAccess

La ReadOnlyAccess politique OAM accorde un accès en lecture seule aux ressources de l'Observability Access Manager, qui sont utilisées pour l'observabilité entre comptes. CloudWatch Pour plus d'informations, consultez CloudWatch observabilité entre comptes. Le contenu est le suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "oam:Get*",
                "oam:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politiques gérées (prédéfinies) pour CloudWatch Synthetics

Les politiques CloudWatchSyntheticsReadOnlyAccess AWS gérées CloudWatchSyntheticsFullAccesset les politiques peuvent être attribuées aux utilisateurs qui géreront ou utiliseront CloudWatch Synthetics. Les politiques supplémentaires suivantes sont également pertinentes :

- AmazonS3 ReadOnlyAccess et CloudWatchReadOnlyAccess— Ils sont nécessaires pour pouvoir lire toutes les données Synthetics dans la console. CloudWatch
- AWSLambdaReadOnlyAccess— Pour pouvoir consulter le code source utilisé par les canaris.
- CloudWatchSyntheticsFullAccessvous permet de créer des canaris. En outre, pour créer et supprimer des canaris auxquels un nouveau rôle IAM a été créé, vous avez également besoin de la déclaration de politique intégrée suivante :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateRole",
                "iam:DeleteRole",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:AttachRolePolicy",
                "iam:DetachRolePolicy",
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*",
                "arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*"
            ]
        }
    ]
}
```

Important

Accorder à un utilisateur les autorisations `iam:CreateRole`, `iam:DeleteRole`, `iam:CreatePolicy`, `iam:DeletePolicy`, `iam:AttachRolePolicy` et `iam:DetachRolePolicy` permet à cet utilisateur de bénéficier d'un accès administratif complet pour créer, attacher et supprimer des rôles et des politiques dont les ARN correspondent à `arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*` et `arn:aws:iam::*:policy/service-role/CloudWatchSyntheticsPolicy*`. Par exemple, un utilisateur disposant de ces autorisations peut créer une politique disposant d'autorisations complètes pour toutes les

ressources et attacher cette politique à n'importe quel rôle. Sélectionnez attentivement les personnes auxquelles vous accordez ces autorisations.

Pour plus d'informations sur l'association des politiques et l'octroi d'autorisations aux utilisateurs, consultez Modification des autorisations pour un utilisateur IAM et Pour intégrer une politique en ligne pour un utilisateur ou un rôle.

CloudWatchSyntheticsFullAccess

Le contenu de la CloudWatchSyntheticsFullAccesspolitique est le suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutEncryptionConfiguration"
            ],
            "Resource": [
                "arn:aws:s3:::cw-syn-results-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles",
                "s3:ListAllMyBuckets",
                "xray:GetTraceSummaries",
                "xray:BatchGetTraces",
                "apigateway:GET"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:ListBucket"
            ],
            "Resource": "arn:aws:s3:::cw-syn-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::aws-synthetics-library-*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "lambda.amazonaws.com",
                        "synthetics.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
```

```
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchSyntheticsRole*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:CreateFunction",
                "lambda:AddPermission",
                "lambda:PublishVersion",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "lambda:GetFunctionConfiguration",
                "lambda:DeleteFunction"
            ],
```

```
            "Resource": [
                "arn:aws:lambda:*:*:function:cwsyn-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "lambda:GetLayerVersion",
                "lambda:PublishLayerVersion",
                "lambda:DeleteLayerVersion"
            ],
            "Resource": [
                "arn:aws:lambda:*:*:layer:cwsyn-*",
                "arn:aws:lambda:*:*:layer:Synthetics:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcs",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
```

```
                "arn:*:sns:*:*:Synthetics-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:ListAliases"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:DescribeKey"
            ],
            "Resource": "arn:aws:kms:*:*:key/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "arn:aws:kms:*:*:key/*",
            "Condition": {
                "StringLike": {
                    "kms:ViaService": [
                        "s3.*.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

CloudWatchSyntheticsReadOnlyAccess

Le contenu de la CloudWatchSyntheticsReadOnlyAccesspolitique est le suivant.

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

```
                "synthetics:Describe*",
                "synthetics:Get*",
                "synthetics:List*",
                "lambda:GetFunctionConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politiques gérées (prédéfinies) pour Amazon CloudWatch RUM

Les politiques ReadOnlyAccess AWS gérées par le AmazonCloudWatchAmazonCloudWatchRUM FullAccess et le RUM peuvent être attribuées aux utilisateurs qui géreront ou utiliseront le CloudWatch RUM.

AmazonCloudWatchRHUM FullAccess

Voici le contenu de la FullAccess politique AmazonCloudWatchRUM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rum:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-service-role/rum.amazonaws.com/
AWSServiceRoleForRealUserMonitoring"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/RUM-Monitor*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "cognito-identity.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": "arn:aws:cloudwatch:*:*:alarm:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cognito-identity:CreateIdentityPool",
                "cognito-identity:ListIdentityPools",
                "cognito-identity:DescribeIdentityPool",
                "cognito-identity:GetIdentityPoolRoles",
                "cognito-identity:SetIdentityPoolRoles"
            ],
            "Resource": "arn:aws:cognito-identity:*:*:identitypool/*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "logs:CreateLogGroup",
                "logs:DeleteLogGroup",
                "logs:PutRetentionPolicy",
                "logs:CreateLogStream"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:*RUMService*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:DescribeResourcePolicies"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:*:*:log-group::log-stream:*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "synthetics:describeCanaries",
                "synthetics:describeCanariesLastRun"
            ],
            "Resource": "arn:aws:synthetics:*:*:canary:*"
        }
    ]
}
```

AmazonCloudWatchRHUM ReadOnlyAccess

Voici le contenu de la ReadOnlyAccess politique AmazonCloudWatchRUM.

```
{

 "Version": "2012-10-17",
```

```
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "rum:GetAppMonitor",
    "rum:GetAppMonitorData",
    "rum:ListAppMonitors",
    "rum:ListRumMetricsDestinations",
    "rum:BatchGetRumMetricDefinitions"
   ],
   "Resource": "*"
  }
 ]
}
```

AmazonCloudWatchRHUM ServiceRolePolicy

Vous ne pouvez pas associer de AmazonCloudWatchRUM ServiceRolePolicy à vos entités IAM. Cette politique est associée à un rôle lié à un service qui permet à CloudWatch RUM de publier des données de surveillance pour d'autres services pertinents AWS . Pour plus d'informations sur ce rôle lié à un service, consultez Utilisation de rôles liés à un service pour RUM CloudWatch.

Le contenu complet de AmazonCloudWatchRUM ServiceRolePolicy est le suivant.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
```

```
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
    }
   }
  }
 ]
}
```

AWS politiques gérées (prédéfinies) pour CloudWatch Evidently

Les politiques CloudWatchEvidentlyReadOnlyAccess AWS gérées CloudWatchEvidentlyFullAccesset les politiques peuvent être attribuées aux utilisateurs qui géreront ou utiliseront CloudWatch Evidently.

CloudWatchEvidentlyFullAccess

Le contenu de la CloudWatchEvidentlyFullAccesspolitique est le suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "evidently:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole"
            ],
            "Resource": [
                "arn:aws:iam::*:role/service-role/CloudWatchRUMEvidentlyRole-*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "arn:aws:s3:::*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:ListTagsForResource"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms",
                "cloudwatch:TagResource",
                "cloudwatch:UnTagResource"
            ],
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudtrail:LookupEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm"
            ],
```

```
            "Resource": [
                "arn:aws:cloudwatch:*:*:alarm:Evidently-Alarm-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:ListTopics"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:ListSubscriptionsByTopic"
            ],
            "Resource": [
                "arn:*:sns:*:*:Evidently-*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

CloudWatchEvidentlyReadOnlyAccess

Le contenu de la CloudWatchEvidentlyReadOnlyAccesspolitique est le suivant.

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

```
            "Effect": "Allow",
            "Action": [
                "evidently:GetExperiment",
                "evidently:GetFeature",
                "evidently:GetLaunch",
                "evidently:GetProject",
                "evidently:GetSegment",
                "evidently:ListExperiments",
                "evidently:ListFeatures",
                "evidently:ListLaunches",
                "evidently:ListProjects",
                "evidently:ListSegments",
                "evidently:ListSegmentReferencs"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politique gérée pour AWS Systems Manager Incident Manager

La AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicypolitique est associée à un rôle lié au service qui permet CloudWatch de déclencher des incidents dans AWS Systems Manager Incident Manager en votre nom. Pour plus d'informations, consultez Autorisations de rôle liées au service pour les CloudWatch alarmes (actions de Systems Manager Incident Manager).

La politique a l'autorisation suivante :

- Incidents SMS : StartIncident

Exemples de politiques gérées par le client

Dans cette section, vous trouverez des exemples de politiques utilisateur qui accordent des autorisations pour diverses CloudWatch actions. Ces politiques fonctionnent lorsque vous utilisez l' CloudWatch API, AWS les SDK ou le AWS CLI.

Exemples

- Exemple 1 : Autoriser l'accès complet de l'utilisateur à CloudWatch
- Exemple 2 : Autoriser l'accès en lecture seule à CloudWatch
- Exemple 3 : Arrêter une instance Amazon EC2 ou y mettre fin

Exemple 1 : Autoriser l'accès complet de l'utilisateur à CloudWatch

Pour accorder un accès complet à un utilisateur CloudWatch, vous pouvez lui accorder la politique CloudWatchFullAccessgérée au lieu de créer une politique gérée par le client. Le contenu du est CloudWatchFullAccessrépertorié dansCloudWatchFullAccess.

Exemple 2 : Autoriser l'accès en lecture seule à CloudWatch

La politique suivante permet à un utilisateur d'accéder en lecture seule aux CloudWatch actions, aux CloudWatch métriques, aux données des journaux CloudWatch et aux données Amazon SNS relatives aux alarmes d'Amazon EC2 Auto Scaling et de les consulter.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "autoscaling:Describe*",
        "cloudwatch:Describe*",
        "cloudwatch:Get*",
        "cloudwatch:List*",
        "logs:Get*",
        "logs:Describe*",
        "logs:StartQuery",
        "logs:StopQuery",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "logs:StartLiveTail",
        "logs:StopLiveTail",
        "sns:Get*",
        "sns:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Exemple 3 : Arrêter une instance Amazon EC2 ou y mettre fin

La politique suivante autorise une action CloudWatch d'alarme pour arrêter ou mettre fin à une instance EC2. Dans l'exemple ci-dessous, les DescribeAlarms actions GetMetricData ListMetrics,

et sont facultatives. Il est recommandé d'inclure ces actions pour vous assurer que vous avez correctement arrêté l'instance ou que vous y avez mis fin.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:DescribeAlarms"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "ec2:DescribeInstanceStatus",
        "ec2:DescribeInstances",
        "ec2:StopInstances",
        "ec2:TerminateInstances"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

CloudWatch mises à jour des politiques AWS gérées

Consultez les détails des mises à jour des politiques AWS gérées CloudWatch 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 Historique du CloudWatch document.

Modification	Description	Date
CloudWatchFullAccessV2 — Mise à jour d'une politique existante	CloudWatch a mis à jour la politique nommée CloudWatchFullAccessV2. Le champ d'application de la `CloudWatchFullAccessPermissions` politique a été mis à jour pour `application-signals:*` permettre aux utilisateurs d'utiliser les signaux d'CloudWatchapplication pour visualiser, étudier et diagnostiquer les problèmes liés à l'état de santé de leurs services.	20 mai 2024
CloudWatchReadOnlyAccess – Mise à jour d’une politique existante	CloudWatch a mis à jour la politique nommée CloudWatchReadOnlyAccess. Le champ d'application de la `CloudWatchReadOnlyAccessPermissions` politique a été mis à jour pour ajouter `application-signals:BatchGet* application-signals:List*` , et `application-signals:Get*` afin que les utilisateurs puissent utiliser les signaux d'CloudWatchapplication pour visualiser, étudier et diagnostiquer les problèmes liés à l'état de santé de leurs services.	20 mai 2024

Modification	Description	Date
	Le champ d'application de `CloudWatchReadOnly GetRolePermissions` a été mis à jour pour ajouter l'`iam:GetRole` action afin que les utilisateurs puissent vérifier si CloudWatch Application Signals est configuré.	
CloudWatchApplicationSignalsServiceRolePolicy – Mise à jour d'une politique existante	CloudWatch a mis à jour la politique nommée CloudWatchApplicationSignalsServiceRolePolicy. La portée des `logs:GetQueryResults` autorisations `logs:StartQuery` et a été modifiée pour ajouter les `arn:aws:logs:*:*:log-group:/aws/application-signals/data:*` ARN `arn:aws:logs:*:*:log-group:/aws/appsignals/*:*` et afin d'activer les signaux d'application sur un plus grand nombre d'architectures.	18 avril 2024

Modification	Description	Date
CloudWatchApplicationSignalsServiceRolePolicy – Mise à jour d’une politique existante	CloudWatch a modifié la portée d'une autorisation dans CloudWatchApplicationSignalsServiceRolePolicy. L'étendue de l'`cloudwatch:GetMetricData` autorisation a été modifiée pour `*` permettre à Application Signals de récupérer des métriques à partir de sources situées dans des comptes liés.	8 avril 2024
CloudWatchAgentServerPolicy – Mise à jour d’une politique existante	CloudWatch a ajouté des autorisations à CloudWatchAgentServerPolicy. Les `logs:PutRetentionPolicy` autorisations`xray:PutTraceSegments` ,`xray:PutTelemetryRecords` , `xray:GetSamplingRules` `xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` et ont été ajoutées afin que l'CloudWatch agent puisse publier des traces X-Ray et modifier les périodes de conservation des groupes de journaux.	12 février 2024

Modification	Description	Date
CloudWatchAgentAdminPolicy – Mise à jour d'une politique existante	CloudWatch a ajouté des autorisations à CloudWatchAgentAdminPolicy. Les `logs:PutRetentionPolicy` autorisations `xray:PutTraceSegments` , `xray:PutTelemetryRecords` , `xray:GetSamplingRules` `xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` et ont été ajoutées afin que l'CloudWatch agent puisse publier des traces X-Ray et modifier les périodes de conservation des groupes de journaux.	12 février 2024

Modification	Description	Date
CloudWatchFullAccessV2 — Mise à jour d'une politique existante	CloudWatch autorisations ajoutées à la CloudWatc hFullAccessV2. Les autorisations existantes pour les actions CloudWatch Synthetics, X-Ray CloudWatch et RUM ainsi que de nouvelles CloudWatch autorisations pour les signaux d'applica tion ont été ajoutées afin que les utilisateurs dotés de cette politique CloudWatch puissent gérer les signaux d'application. L'autorisation de créer le rôle lié au service CloudWatc h Application Signals a été ajoutée pour permettre à CloudWatch Application Signals de découvrir les données de télémétrie dans les journaux, les métriques, les traces et les balises.	5 décembre 2023

Modification	Description	Date
CloudWatchReadOnlyAccess – Mise à jour d’une politique existante	CloudWatch a ajouté des autorisations à CloudWatc hReadOnlyAccess. Les autorisations de lecture seule existantes pour les actions Synthetics CloudWatc h , X-Ray et CloudWatch RUM ainsi que de nouvelles autorisations de lecture seule pour les signaux d' CloudWatc h application ont été ajoutées afin que les utilisateurs soumis à cette politique puissent trier et diagnostiquer les problèmes de santé de leurs services tels que signalés par Application Signals. CloudWatch L'`cloudwatch:Generat eQuery` autorisation a été ajoutée afin que les utilisate urs dotés de cette politique puissent générer une chaîne de requête CloudWatch Metrics Insights à partir d'une invite en langage naturel.	5 décembre 2023

Modification	Description	Date
CloudWatchApplicationSignalsServiceRolePolicy : nouvelle politique	CloudWatch a ajouté une nouvelle politique CloudWatchApplicationSignalsServiceRolePolicy. Il CloudWatchApplicationSignalsServiceRolePolicy accorde à une fonctionnalité à venir l'autorisation de collecter CloudWatch les données des journaux, les données de trace X-Ray, CloudWatch les données métriques et les données de marquage.	9 novembre 2023
AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy : nouvelle politique	CloudWatch a ajouté une nouvelle politique AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy. Le AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyautorise CloudWatch à récupérer les métriques Performance Insights à partir de bases de données en votre nom.	20 septembre 2023

Modification	Description	Date
CloudWatchReadOnlyAccess – Mise à jour d'une politique existante	CloudWatch a ajouté une autorisation à CloudWatchReadOnlyAccess. L'autorisation `application-autoscaling:DescribeScalingPolicies` a été ajoutée pour que les utilisateurs dotés de cette politique puissent accéder aux informations sur les politiques Application Auto Scaling.	14 septembre 2023
CloudWatchFullAccessV2 — Nouvelle politique	CloudWatch a ajouté une nouvelle politique CloudWatchFullAccessV2. La CloudWatchFullAccessV2 accorde un accès complet aux CloudWatch actions et aux ressources tout en définissant mieux les autorisations accordées à d'autres services tels qu'Amazon Amazon EC2 Auto Scaling SNS et. Pour plus d'informations, reportez-vous à la section CloudWatchFullAccessV2.	1er août 2023

Modification	Description	Date
AWSServiceRoleForInternetMonitor – Mise à jour d'une politique existante	Amazon CloudWatch Internet Monitor a ajouté de nouvelles autorisations pour surveiller les ressources de Network Load Balancer. Les autorisations `elasticloadbalancing:DescribeLoadBalancers` et `ec2:DescribeNetworkInterfaces` sont requises pour que Moniteur Internet puisse surveiller le trafic Network Load Balancer des clients en analysant les journaux de flux pour détecter les ressources NLB. Pour plus d'informations, consultez Utilisation d'Amazon CloudWatch Internet Monitor.	15 juillet 2023

Modification	Description	Date
CloudWatchReadOnlyAccess – Mise à jour d'une politique existante	CloudWatch a ajouté des autorisations à CloudWatc hReadOnlyAccess. Les `logs:StopLiveTail` autorisations `logs:Star tLiveTail` et ont été ajoutées afin que les utilisate urs soumis à cette politique puissent utiliser la console pour démarrer et arrêter CloudWatch les sessions Logs Live Tail. Pour plus d'informa tions, veuillez consulter Utilisation de Live Tail pour visualiser les journaux en temps quasi réel.	6 juin 2023
CloudWatchCrossAcc ountSharingConfiguration : nouvelle politique	CloudWatch a ajouté une nouvelle politique pour vous permettre de gérer les liens d'observabilité CloudWatch entre comptes qui partagent CloudWatch des métriques. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.	27 novembre 2022

Modification	Description	Date
OAM FullAccess — Nouvelle politique	CloudWatch a ajouté une nouvelle politique pour vous permettre de gérer entièrement les liens et CloudWatch les puits d'observabilité entre comptes. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.	27 novembre 2022
OAM ReadOnlyAccess — Nouvelle politique	CloudWatch a ajouté une nouvelle politique pour vous permettre de consulter les informations sur les liens et CloudWatch les puits d'observabilité entre comptes. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.	27 novembre 2022

Modification	Description	Date
CloudWatchFullAccess – Mise à jour d'une politique existante	CloudWatch a ajouté des autorisations à CloudWatchFullAccess. Les `oam:ListAttachedLinks` autorisations `oam:ListSinks` et ont été ajoutées afin que les utilisateurs soumis à cette politique puissent utiliser la console pour consulter les données partagées à partir de comptes sources dans le cadre d'une CloudWatch observabilité entre comptes.	27 novembre 2022
CloudWatchReadOnlyAccess – Mise à jour d'une politique existante	CloudWatch a ajouté des autorisations à CloudWatchReadOnlyAccess. Les `oam:ListAttachedLinks` autorisations `oam:ListSinks` et ont été ajoutées afin que les utilisateurs soumis à cette politique puissent utiliser la console pour consulter les données partagées à partir de comptes sources dans le cadre d'une CloudWatch observabilité entre comptes.	27 novembre 2022

Modification	Description	Date
AmazonCloudWatchRUM ServiceRolePolicy — Mise à jour d'une politique existante	CloudWatch RUM a mis à jour une clé de condition dans AmazonCloudWatchRUM ServiceRolePolicy. La clé de `"Condition": { "StringEquals": { "cloudwatch:namespace": "AWS/RUM" } }` condition a été modifiée comme suit afin que CloudWatch RUM puisse envoyer des métriques personnalisées à des espaces de noms de métriques personnalisés. `"Condition": { "StringLike": { "cloudwatch:namespace": [ "RUM/CustomMetrics/*", "AWS/RUM" ] } }`	2 février 2023

Modification	Description	Date
AmazonCloudWatchRUMReadOnlyAccess — Politique mise à jour	CloudWatch a ajouté des autorisations à la ReadOnlyAccess politique AmazonCloudWatchRUM. Les `rum:BatchGetRumMetricsDefinitions` autorisations `rum:ListRumMetricsDestinations` et ont été ajoutées afin que CloudWatch RUM puisse envoyer des métriques étendues à CloudWatch et Evidently.	27 octobre 2022
AmazonCloudWatchRUMServiceRolePolicy — Mise à jour d'une politique existante	CloudWatch RUM a ajouté des autorisations à AmazonCloudWatchRUMServiceRolePolicy. L'`cloudwatch:PutMetricData` autorisation a été ajoutée afin que CloudWatch RUM puisse envoyer des métriques étendues à CloudWatch.	26 octobre 2022

Modification	Description	Date
CloudWatchEvidentlyReadOnlyAccess – Mise à jour d'une politique existante	CloudWatch Des autorisations ont évidemment été ajoutées à. CloudWatchEvidentlyReadOnlyAccess Les autorisations `evidently:GetSegment` , `evidently:ListSegments` et `evidently:ListSegmentReferences` ont été ajoutées pour que les utilisateurs ayant cette politique puissent voir les segments d'audience Evidently qui ont été créés.	12 août 2022
CloudWatchSyntheticsFullAccess – Mise à jour d'une politique existante	CloudWatch Synthetics a ajouté des autorisations à. CloudWatchSyntheticsFullAccess Les `lambda:DeleteLayerVersion` autorisations `lambda:DeleteFunction` et ont été ajoutées afin que CloudWatch Synthetics puisse supprimer les ressources associées lorsqu'un canari est supprimé. L'autorisation `iam:ListAttachedRolePolicies` a été ajoutée afin que les clients puissent afficher les politiques attachées au rôle IAM d'un canary.	6 mai 2022

Modification	Description	Date
AmazonCloudWatchRUM FullAccess — Nouvelle politique	CloudWatch a ajouté une nouvelle politique pour permettre la gestion complète du CloudWatch RUM. CloudWatch RUM vous permet d'effectuer une véritable surveillance des utilisateurs de votre application Web. Pour plus d'informations, consultez Utiliser du CloudWatch rhum.	29 novembre 2021
AmazonCloudWatchRUMReadOnlyAccess — Nouvelle politique	CloudWatch a ajouté une nouvelle politique pour permettre l'accès en lecture seule au CloudWatch RUM. CloudWatch RUM vous permet d'effectuer une véritable surveillance des utilisateurs de votre application Web. Pour plus d'informations, consultez Utiliser du CloudWatch rhum.	29 novembre 2021

Modification	Description	Date
CloudWatchEvidentlyFullAccess : nouvelle politique	CloudWatch a ajouté une nouvelle politique pour permettre la gestion complète d' CloudWatchEvidently. CloudWatch Vous permet évidemment de réaliser des tests A/B sur vos applications Web et de les déployer progressivement. Pour plus d'informations, consultez Réalisez des lancements et des expériences A/B avec Evidently CloudWatch .	29 novembre 2021
CloudWatchEvidentlyReadOnlyAccess : nouvelle politique	CloudWatch a ajouté une nouvelle politique pour permettre l'accès en lecture seule à CloudWatch Evidently. CloudWatch Vous permet évidemment de réaliser des tests A/B sur vos applications Web et de les déployer progressivement. Pour plus d'informations, consultez Réalisez des lancements et des expériences A/B avec Evidently CloudWatch .	29 novembre 2021

Modification	Description	Date
AWSServiceRoleForCloudWatchRUM— Nouvelle politique gérée	CloudWatch a ajouté une politique pour un nouveau rôle lié au service afin de permettre au CloudWatch RUM de publier des données de surveillance à d'autres services pertinents. AWS	29 novembre 2021

Modification	Description	Date
CloudWatchSyntheticsFullAccess – Mise à jour d'une politique existante	CloudWatch Synthetics a ajouté des autorisations CloudWatchSyntheticsFullAccessà une autorisation et en a également modifié la portée. L'`kms:ListAliases` autorisation a été ajoutée afin que les utilisateurs puissent répertorier AWS KMS les clés disponibles pouvant être utilisées pour chiffrer les artefacts Canary. Le`kms:DescribeKey` l'autorisation a été ajoutée afin que les utilisateurs puissent voir les détails des clés qui seront utilisées pour chiffrer des artefacts Canary. Et le`kms:Decrypt` l'autorisation a été ajoutée pour permettre aux utilisateurs de déchiffrer les artefacts Canary. Cette capacité de déchiffrement est limitée à une utilisation sur les ressources des compartiments Amazon S3. Le`Resource`portée de la`s3:GetBucketLocation` l'autorisation a été modifiée à partir de`*`pour`arn:aws:s3:::*` .	29 septembre 2021

Modification	Description	Date
CloudWatchSyntheticsFullAccess – Mise à jour d'une politique existante	CloudWatch Synthetics a ajouté une autorisation pour. CloudWatchSyntheticsFullAccess L'autorisation `lambda:UpdateFunctionCode` a été ajoutée afin que les utilisateurs disposant de cette politique puissent modifier la version d'exécution de scripts Canary.	20 juillet 2021
AWSCloudWatchAlarms_ActionSSMIncidentsServiceRolePolicy— Nouvelle politique gérée	CloudWatch a ajouté une nouvelle politique IAM gérée pour permettre de CloudWatch créer des incidents dans AWS Systems Manager Incident Manager.	10 mai 2021
CloudWatchAutomaticDashboardsAccess – Mise à jour d'une politique existante	CloudWatch a ajouté une autorisation à la politique CloudWatchAutomaticDashboardsAccessgérée. L'`synthetics:DescribeCanariesLastRun` autorisation a été ajoutée à cette politique pour permettre aux utilisateurs du tableau de bord multi-comptes de voir des informations sur CloudWatch Synthetics Canary Runs.	20 avril 2021

Modification	Description	Date
CloudWatch a commencé à suivre les modifications	CloudWatch a commencé à suivre les modifications apportées AWS à ses politiques gérées.	14 avril 2021

Utilisation de clés de condition pour limiter l'accès aux espaces de CloudWatch noms

Utilisez les clés de condition IAM pour limiter les utilisateurs à publier des métriques uniquement dans les CloudWatch espaces de noms que vous spécifiez.

Autorisation de publication dans un seul espace de noms

La politique suivante contraint l'utilisateur à publier des métriques uniquement dans l'espace de noms appelé `MyCustomNamespace`.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Resource": "*",
        "Action": "cloudwatch:PutMetricData",
        "Condition": {
            "StringEquals": {
                "cloudwatch:namespace": "MyCustomNamespace"
            }
        }
    }
}
```

Exclusion de publication dans un espace de noms

La politique suivante permet à l'utilisateur de publier des métriques dans n'importe quel espace de noms, à l'exception de `CustomNamespace2`.

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

```
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData"
        },
        {
            "Effect": "Deny",
            "Resource": "*",
            "Action": "cloudwatch:PutMetricData",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "CustomNamespace2"
                }
            }
        }
    ]
}
```

Utilisation de clés de condition pour limiter l'accès des utilisateurs Contributor Insights aux groupes de journaux

Pour créer une règle dans Contributor Insights et voir ses résultats, un utilisateur doit disposer de l'option autorisation `cloudwatch:PutInsightRule`. Par défaut, un utilisateur disposant de cette autorisation peut créer une règle Contributor Insights qui évalue n'importe quel groupe de CloudWatch journaux dans Logs, puis en voit les résultats. Les résultats peuvent contenir des données de contributeur pour ces groupes de journaux.

Vous pouvez créer des politiques IAM avec des clés de condition pour accorder aux utilisateurs l'autorisation d'écrire des règles Contributor Insights pour certains groupes de journaux, tout en les empêchant d'écrire des règles pour et d'afficher ces données à partir d'autres groupes de journaux.

Pour plus d'informations sur l'élément `Condition` dans les politiques IAM, consultez Éléments de politique JSON IAM : condition.

Autoriser l'accès aux règles d'écriture et afficher les résultats pour certains groupes de journaux uniquement

La politique suivante permet à l'utilisateur d'écrire des règles et d'afficher les résultats pour le groupe de journaux nommé `AllowedLogGroup` et tous les groupes de journaux dont le nom commence par `AllowedWildCard`. Il n'accorde pas l'accès aux règles d'écriture ou à l'affichage des résultats des règles pour les autres groupes de journaux.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCertainLogGroups",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "AllowedLogGroup",
                        "AllowedWildcard*"
                    ]
                }
            }
        }
    ]
}
```

Interdire les règles d'écriture pour des groupes de journaux spécifiques, mais autoriser l'écriture de règles pour tous les autres groupes de journaux

La politique suivante refuse explicitement à l'utilisateur l'accès pour écrire des règles et afficher les résultats des règles pour le groupe de journaux nommé `ExplicitlyDeniedLogGroup`, mais permet d'écrire des règles et d'afficher les résultats des règles pour tous les autres groupes de journaux.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowInsightRulesOnLogGroupsByDefault",
            "Effect": "Allow",
            "Action": "cloudwatch:PutInsightRule",
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*"

        },
        {
            "Sid": "ExplicitDenySomeLogGroups",
            "Effect": "Deny",
            "Action": "cloudwatch:PutInsightRule",
```

```
            "Resource": "arn:aws:cloudwatch:*:*:insight-rule/*",
            "Condition": {
                "ForAllValues:StringEqualsIgnoreCase": {
                    "cloudwatch:requestInsightRuleLogGroups": [
                        "/test/alpine/ExplicitlyDeniedLogGroup"
                    ]
                }
            }
        }
    ]
}
```

Utilisation des clés de condition pour limiter les actions d'alarme

Lorsque les CloudWatch alarmes changent d'état, elles peuvent effectuer différentes actions, telles que l'arrêt et la mise hors service des instances EC2 et les actions de Systems Manager. Ces actions peuvent être lancées lorsque l'alarme change à n'importe quel état, y compris ALARM, OK ou INSUFFICIENT_DATA.

Utilisation de la clé de condition `cloudwatch:AlarmActions` pour permettre à un utilisateur de créer des alarmes qui ne peuvent effectuer que les actions que vous spécifiez lorsque l'état de l'alarme change. Par exemple, vous pouvez autoriser un utilisateur à créer des alarmes qui ne peuvent effectuer que des actions qui ne sont pas des actions EC2.

Autoriser un utilisateur à créer des alarmes qui peuvent uniquement envoyer des notifications Amazon SNS ou effectuer des actions du Systems Manager

La politique suivante limite l'utilisateur à créer des alarmes qui peuvent uniquement envoyer des notifications Amazon SNS et à effectuer des actions du Systems Manager. L'utilisateur ne peut créer aucune alarme qui exécute les actions EC2.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateAlarmsThatCanPerformOnlySNSandSSMActions",
            "Effect": "Allow",
            "Action": "cloudwatch:PutMetricAlarm",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringLike": {
                    "cloudwatch:AlarmActions": [
```

```
                    "arn:aws:sns:*",
                    "arn:aws:ssm:*"
                ]
            }
        }
    }
  ]
}
```

Utilisation des rôles liés aux services pour CloudWatch

Amazon CloudWatch utilise des rôles AWS Identity and Access Management liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à. CloudWatch Les rôles liés au service sont prédéfinis par CloudWatch et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Un rôle lié au service CloudWatch permet de configurer des CloudWatch alarmes qui peuvent mettre fin à, arrêter ou redémarrer une instance Amazon EC2 sans que vous ayez à ajouter manuellement les autorisations nécessaires. Un autre rôle lié à un service permet à un compte de surveillance d'accéder aux CloudWatch données d'autres comptes que vous spécifiez, afin de créer des tableaux de bord inter-comptes interrégionaux.

CloudWatch définit les autorisations de ces rôles liés aux services et, sauf indication contraire, seul CloudWatch peut assumer le rôle. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer les rôles uniquement après la suppression préalable de leurs ressources connexes. Cette restriction protège vos CloudWatch ressources car vous ne pouvez pas supprimer par inadvertance les autorisations d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés aux services, consultez Services AWS fonctionnant avec IAM et recherchez les services où Oui figure dans la colonne Rôle lié à un service. Choisissez un Yes (oui) ayant un lien permettant de consulter les détails du rôle pour ce service.

Autorisations de rôle liées au service pour les actions d' CloudWatch alarme EC2

CloudWatch utilise le rôle lié au service nommé AWSServiceRoleForCloudWatchEvents— CloudWatch utilise ce rôle lié au service pour effectuer des actions d'alarme Amazon EC2.

Le rôle AWSServiceRoleForCloudWatchEvents lié au service fait confiance au service CloudWatch Events pour assumer ce rôle. CloudWatch Les événements invoquent les actions de fin, d'arrêt ou de redémarrage de l'instance lorsque l'alarme l'appelle.

La politique d'autorisation des rôles AWSServiceRoleForCloudWatchEvents liés au service permet à CloudWatch Events d'effectuer les actions suivantes sur les instances Amazon EC2 :

- `ec2:StopInstances`
- `ec2:TerminateInstances`
- `ec2:RecoverInstances`
- `ec2:DescribeInstanceRecoveryAttribute`
- `ec2:DescribeInstances`
- `ec2:DescribeInstanceStatus`

La politique d'autorisation des rôles AWSServiceRoleForCloudWatchCrossAccountliés au service permet d' CloudWatch effectuer les actions suivantes :

- `sts:AssumeRole`

Autorisations de rôle liées au service pour les signaux d'application CloudWatch

CloudWatch Application Signals utilise le rôle lié à un service nommé AWSServiceRoleForCloudWatchApplicationSignals: CloudWatch utilise ce rôle lié à un service pour collecter les données des journaux, les CloudWatch données de suivi X-Ray, les données CloudWatch métriques et les données de balisage à partir des applications que vous avez activées pour Application Signals. CloudWatch

Le rôle AWSServiceRoleForCloudWatchApplicationSignalslié au service fait confiance à CloudWatch Application Signals pour assumer le rôle. Application Signals collecte les données des journaux, des suivis, des métriques et des balises de votre compte.

Une politique IAM y est attachée, et cette politique est nommée CloudWatchApplicationSignalsServiceRolePolicy. AWSServiceRoleForCloudWatchApplicationSignals Cette politique autorise CloudWatch Application Signals à collecter des données de surveillance et de balisage auprès d'autres AWS services pertinents. Elle inclut les autorisations qui permettent à Application Signals d’effectuer les actions suivantes :

- xray:GetServiceGraph
- logs:StartQuery
- logs:GetQueryResults
- cloudwatch:GetMetricData
- cloudwatch:ListMetrics
- tag:GetResources

Le contenu complet de CloudWatchApplicationSignalsServiceRolePolicyest le suivant :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "XRayPermission",
            "Effect": "Allow",
            "Action": [
                "xray:GetServiceGraph"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWLogsPermission",
            "Effect": "Allow",
            "Action": [
                "logs:StartQuery",
                "logs:GetQueryResults"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/appsignals/*:*",
                "arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
                }
            }
        },
        {
            "Sid": "CWListMetricsPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "CWGetMetricDataPermission",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "TagsPermission",
            "Effect": "Allow",
            "Action": [
                "tag:GetResources"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
```

```
}
```

Autorisations de rôle liées au service pour les actions d' CloudWatch alarmes Systems Manager OpsCenter

CloudWatch utilise le rôle lié au service nommé AWSServiceRoleForCloudWatchAlarms_ActionSSM— CloudWatch utilise ce rôle lié au service pour exécuter les OpsCenter actions de Systems Manager lorsqu'une CloudWatch alarme passe à l'état ALARM.

Le rôle AWSServiceRoleForCloudWatchAlarms_ActionSSM lié au service fait confiance au CloudWatch service pour assumer le rôle. CloudWatch les alarmes invoquent les OpsCenter actions de Systems Manager lorsqu'elles sont déclenchées par l'alarme.

La politique d'autorisation des rôles AWSServiceRoleForCloudWatchAlarms_ActionSSMliés au service permet à Systems Manager d'effectuer les actions suivantes :

- `ssm:CreateOpsItem`

Autorisations de rôle liées au service pour les CloudWatch alarmes (actions de Systems Manager Incident Manager)

CloudWatch utilise le rôle lié au service nommé AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents: CloudWatch utilise ce rôle lié au service pour déclencher des incidents Incident Manager lorsqu'une CloudWatch alarme passe à l'état ALARM.

Le rôle AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidentslié au service fait confiance au CloudWatch service pour assumer le rôle. CloudWatch les alarmes appellent l'action Systems Manager Incident Manager lorsqu'elles sont déclenchées par l'alarme.

La politique d'autorisation des rôles AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidentsliés au service permet à Systems Manager d'effectuer les actions suivantes :

- `ssm-incidents:StartIncident`

Autorisations de rôle liées à un service pour CloudWatch plusieurs comptes et entre régions

CloudWatch utilise le rôle lié au service nommé AWSServiceRoleForCloudWatchCrossAccount: CloudWatch utilise ce rôle pour accéder aux CloudWatch données des autres AWS comptes que vous spécifiez. Le SLR fournit uniquement l'autorisation d'assumer le rôle pour permettre au CloudWatch service d'assumer le rôle dans le compte de partage. C'est le rôle de partage qui fournit l'accès aux données.

La politique d'autorisation des rôles AWSServiceRoleForCloudWatchCrossAccountliés au service permet d' CloudWatch effectuer les actions suivantes :

- `sts:AssumeRole`

Le rôle AWSServiceRoleForCloudWatchCrossAccountlié au service fait confiance au CloudWatch service pour assumer le rôle.

Autorisations de rôle liées à un service pour la base de CloudWatch données Performance Insights

CloudWatch utilise le rôle lié au service nommé. AWSServiceRoleForCloudWatchMetrics_DbPerfInsights — CloudWatch utilise ce rôle pour récupérer les métriques Performance Insights afin de créer des alarmes et de créer des instantanés.

La politique `AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicy` IAM est AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsattachée au rôle lié au service. Le contenu de cette politique est le suivant :

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "pi:GetResourceMetrics"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceAccount": "${aws:PrincipalAccount}"
```

```
      }
    }
   }
  ]
}
```

Le rôle AWSServiceRoleForCloudWatchMetrics_DbPerfInsightslié au service fait confiance au CloudWatch service pour assumer le rôle.

Création d'un rôle lié à un service pour CloudWatch

Vous n'avez pas besoin de créer manuellement l'un ou l'autre de ces rôles liés à un service. La première fois que vous créez une alarme dans le AWS Management Console, l'IAM CLI, ou l'API IAM, CloudWatch crée AWSServiceRoleForCloudWatchEvents et AWSServiceRoleForCloudWatchAlarms_ActionSSMpour vous.

La première fois que vous activez la découverte de services et de topologies, Application Signals crée AWSServiceRoleForCloudWatchApplicationSignalspour vous.

Lorsque vous activez pour la première fois un compte comme compte de surveillance pour la fonctionnalité inter-comptes interrégionaux, il CloudWatch crée AWSServiceRoleForCloudWatchCrossAccountpour vous.

Lorsque vous créez pour la première fois une alarme qui utilise la fonction mathématique `DB_PERF_INSIGHTS` métrique, CloudWatch elle le fait AWSServiceRoleForCloudWatchMetrics_DbPerfInsightspour vous.

Pour plus d'informations, consultez Création d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Modification d'un rôle lié à un service pour CloudWatch

CloudWatch ne vous permet pas de modifier les AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsrôles AWSServiceRoleForCloudWatchEventsAWSServiceRoleForCloudWatchAlarms_ActionSSMAWSServiceRol ou. Après avoir créé ces rôles, vous ne pouvez pas modifier leurs noms, car diverses entités peuvent y faire référence. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM.

Modification de la description d'un rôle lié à un service (console IAM)

Vous pouvez utiliser la console IAM, pour modifier la description d'un rôle lié à un service.

Pour modifier la description d'un rôle lié à un service (console)

1. Dans le panneau de navigation de la console IAM, sélectionnez Roles (Rôles).
2. Choisissez le nom du rôle à modifier.
3. A l'extrême droite de Description du rôle, choisissez Edit (Modifier).
4. Saisissez une nouvelle description dans la zone et choisissez Save (Enregistrer).

Modification de la description d'un rôle lié à un service (AWS CLI)

Vous pouvez utiliser les commandes IAM depuis le AWS Command Line Interface pour modifier la description d'un rôle lié à un service.

Pour changer la description d'un rôle lié à un service (AWS CLI)

1. (Facultatif) Pour afficher la description actuelle d'un rôle, utilisez les commandes suivantes :

   ```
   $ aws iam get-role --role-name role-name
   ```

   Utilisez le nom du rôle, pas l'ARN, pour faire référence aux rôles avec les commandes AWS CLI . Par exemple, si un rôle a l'ARN : `arn:aws:iam::123456789012:role/myrole`, vous faites référence au rôle en tant que **`myrole`**.
2. Pour mettre à jour la description d'un rôle lié à un service, utilisez la commande suivante :

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Modification de la description d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM pour modifier la description d'un rôle lié à un service.

Pour changer la description d'un rôle lié à un service (API)

1. (Facultatif) Pour afficher la description actuelle d'un rôle, utilisez la commande suivante :

   GetRole
2. Pour mettre à jour la description d'un rôle, utilisez la commande suivante :

   UpdateRoleDescription

Supprimer un rôle lié à un service pour CloudWatch

Si vous n'avez plus d'alarmes qui arrêtent, mettent fin ou redémarrent automatiquement les instances EC2, nous vous recommandons de supprimer le AWSServiceRoleForCloudWatchEvents rôle.

Si vous n'avez plus d'alarmes qui exécutent des OpsCenter actions de Systems Manager, nous vous recommandons de supprimer le AWSServiceRoleForCloudWatchAlarms_ActionSSM rôle.

Si vous supprimez toutes les alarmes qui utilisent la fonction mathématique `DB_PERF_INSIGHTS` métrique, nous vous recommandons de supprimer le rôle AWSServiceRoleForCloudWatchMetrics_DbPerfInsightslié au service.

De cette façon, vous n’avez aucune entité inutilisée qui n’est pas surveillée ou gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer.

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord vérifier qu'aucune session n'est active pour le rôle et supprimer toutes les ressources utilisées par le rôle.

Pour vérifier si une session est active pour le rôle lié à un service dans la console IAM

1. Ouvrez la console IAM à l’adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, sélectionnez Rôles. Choisissez le nom (et non la case à cocher) du AWSServiceRoleForCloudWatchEvents rôle.
3. Sur la page Summary (Résumé) du rôle sélectionné, choisissez Access Advisor et consultez l'activité récente du rôle lié au service.

> **Note**
>
> Si vous ne savez pas si le AWSServiceRoleForCloudWatchEvents rôle CloudWatch est utilisé, essayez de le supprimer. Si le service utilise le rôle, la suppression échoue et vous avez accès aux régions dans lesquelles le rôle est utilisé. Si le rôle est utilisé, vous devez attendre que la session se termine avant de pouvoir le supprimer. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Suppression d'un rôle lié à un service (console IAM)

Vous pouvez utiliser la console IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (console)

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, sélectionnez Rôles. Cochez la case en regard du nom du rôle que vous souhaitez supprimer, sans sélectionner le nom ou la ligne.
3. Pour Role actions (Actions du rôle), choisissez Delete role (Supprimer le rôle).
4. Dans la boîte de dialogue de confirmation, vérifiez les dernières données consultées dans le service. Elles indiquent quels rôles, parmi ceux sélectionnés, ont accédé en dernier à un service AWS . Cela vous permet de confirmer si le rôle est actif actuellement. Pour poursuivre, choisissez Oui, supprimer.
5. Consultez les notifications de la console IAM pour surveiller la progression de la suppression du rôle lié à un service. Dans la mesure où la suppression du rôle lié à un service IAM est asynchrone, la suppression peut réussir ou échouer après que vous soumettez le rôle afin qu'il soit supprimé. Si la tâche échoue, choisissez View details (Afficher les détails) ou View Resources (Afficher les ressources) à partir des notifications pour connaître le motif de l'échec de la suppression. Si la suppression échoue parce que certaines ressources du service sont actuellement utilisées par le rôle, la raison de l'échec comprend une liste de ressources.

Suppression d'un rôle lié à un service (AWS CLI)

Vous pouvez utiliser les commandes IAM depuis le AWS Command Line Interface pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (AWS CLI)

1. Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `deletion-task-id` de la réponse afin de vérifier l'état de la tâche de suppression. Tapez la commande suivante pour envoyer une demande de suppression d'un rôle lié à un service :

```
$ aws iam delete-service-linked-role --role-name service-linked-role-name
```

2. Tapez la commande suivante pour vérifier l'état de la tâche de suppression :

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

L'état de la tâche de suppression peut être `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

Suppression d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (API)

1. Pour soumettre une demande de suppression pour un rôle lié à un service, appelez. DeleteServiceLinkedRole Dans la demande, spécifiez le nom de rôle que vous souhaitez supprimer.

   Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `DeletionTaskId` de la réponse afin de vérifier l'état de la tâche de suppression.
2. Pour vérifier l'état de la suppression, appelez GetServiceLinkedRoleDeletionStatus. Dans la demande, spécifiez le `DeletionTaskId`.

   L'état de la tâche de suppression peut être `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

CloudWatch mises à jour des rôles AWS liés aux services

Consultez les détails des mises à jour des politiques AWS gérées CloudWatch 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 Historique du CloudWatch document.

Modification	Description	Date
		24 avril 2024

Modification	Description	Date
AWSServiceRoleForCloudWatchApplicationSignals — Mise à jour des autorisations de la politique des rôles liés aux services	CloudWatch ajoutez d'autres groupes de journaux à la portée `logs:StartQuery` des `logs:GetQueryResults` autorisations accordées par ce rôle.	
AWSServiceRoleForCloudWatchApplicationSignals — Nouveau rôle lié au service	CloudWatch a ajouté ce nouveau rôle lié à un service pour permettre à CloudWatch Application Signals de collecter les données des CloudWatch journaux, les données de suivi X-Ray, CloudWatch les données métriques et les données de balisage à partir des applications que vous avez activées pour CloudWatch Application Signals.	9 novembre 2023
AWSServiceRoleForCloudWatchMetrics_DbPerfInsights— Nouveau rôle lié au service	CloudWatch a ajouté ce nouveau rôle lié au service pour permettre de récupérer les métriques de Performance Insights CloudWatch à des fins d'alarme et de capture d'écran. Une politique IAM est associée à ce rôle, et cette politique autorise l'extraction des métriques CloudWatch de Performance Insights en votre nom.	13 septembre 2023

Modification	Description	Date
AWSServiceRoleForCloudWatchAlarms_ActionSSMIncidents— Nouveau rôle lié au service	CloudWatch a ajouté un nouveau rôle lié au service pour permettre de CloudWatch créer des incidents dans AWS Systems Manager Incident Manager.	26 avril 2021
CloudWatch a commencé à suivre les modifications	CloudWatch a commencé à suivre les modifications apportées à ses rôles liés aux services.	26 avril 2021

Utilisation de rôles liés à un service pour RUM CloudWatch

CloudWatch RUM utilise un rôle AWS Identity and Access Management lié à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM lié directement à RUM. Le rôle lié au service est prédéfini par RUM et inclut toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

RUM définit les autorisations du rôle lié à un service et, sauf définition contraire, seul RUM peut endosser ce rôle. Les autorisations définies comprennent la politique d’approbation et la politique d’autorisation. De plus, cette politique d’autorisation ne peut pas être attachée à une autre entité IAM.

Vous ne pouvez supprimer le rôle qu'après avoir d'abord supprimé ses ressources liées. Cette restriction protège vos ressources RUM, car vous ne pouvez pas involontairement supprimer d'autorisations pour accéder aux ressources.

Pour plus d’informations sur les autres services prenant en charge les rôles liés à un service, consultez les AWS services opérationnels avec IAM et recherchez les services présentant la mention Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Autorisations de rôles liés à des services pour RUM

RUM utilise le rôle lié à un service nommé AWSServiceRoleForCloudWatchRUM: ce rôle permet à RUM d'envoyer des données de AWS X-Ray suivi à votre compte, pour les moniteurs d'applications pour lesquels vous activez le suivi X-Ray.

Le rôle AWSServiceRoleForCloudWatchRUMlié au service fait confiance au service X-Ray pour assumer le rôle. X-Ray envoie les données de suivi à votre compte.

Le rôle AWSServiceRoleForCloudWatchRUMlié au service est associé à une politique IAM nommée RUM. AmazonCloudWatch ServiceRolePolicy Cette politique autorise CloudWatch RUM à publier des données de surveillance auprès d'autres AWS services concernés. Il inclut les autorisations qui permettent à RUM d'effectuer les actions suivantes :

- `xray:PutTraceSegments`
- `cloudwatch:PutMetricData`

Le contenu complet de AmazonCloudWatchRUM ServiceRolePolicy est le suivant.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "xray:PutTraceSegments"
   ],
   "Resource": [
    "*"
   ]
  },
  {
   "Effect": "Allow",
   "Action": "cloudwatch:PutMetricData",
   "Resource": "*",
   "Condition": {
    "StringLike": {
     "cloudwatch:namespace": [
      "RUM/CustomMetrics/*",
      "AWS/RUM"
     ]
```

```
      }
    }
   }
  ]
}
```

Création d'un rôle lié à un service pour

Il n'est pas nécessaire de créer manuellement le rôle lié à un service pour CloudWatch RUM. La première fois que vous créez un moniteur d'application avec le suivi X-Ray activé, ou que vous mettez à jour un moniteur d'application pour utiliser le suivi X-Ray, RUM le crée AWSServiceRoleForCloudWatchRUMpour vous.

Pour plus d'informations, consultez Création d'un rôle lié à un servie dans le Guide de l'utilisateur IAM.

Modification d'un rôle lié à un service pour RUM

CloudWatch RUM ne vous permet pas de modifier le AWSServiceRoleForCloudWatchRUMrôle. Après avoir créé ces rôles, vous ne pouvez pas modifier leurs noms, car diverses entités peuvent y faire référence. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM.

Modification de la description d'un rôle lié à un service (console IAM)

Vous pouvez utiliser la console IAM, pour modifier la description d'un rôle lié à un service.

Pour modifier la description d'un rôle lié à un service (console)

1. Dans le panneau de navigation de la console IAM, sélectionnez Roles (Rôles).
2. Choisissez le nom du rôle à modifier.
3. A l'extrême droite de Description du rôle, choisissez Edit (Modifier).
4. Saisissez une nouvelle description dans la zone et choisissez Save (Enregistrer).

Modification de la description d'un rôle lié à un service (AWS CLI)

Vous pouvez utiliser les commandes IAM depuis le AWS Command Line Interface pour modifier la description d'un rôle lié à un service.

Pour changer la description d'un rôle lié à un service (AWS CLI)

1. (Facultatif) Pour afficher la description actuelle d'un rôle, utilisez les commandes suivantes :

```
$ aws iam get-role --role-name role-name
```

Utilisez le nom du rôle, pas l'ARN, pour faire référence aux rôles avec les commandes AWS CLI . Par exemple, si un rôle a l'ARN : `arn:aws:iam::123456789012:role/myrole`, vous faites référence au rôle en tant que **`myrole`**.

2. Pour mettre à jour la description d'un rôle lié à un service, utilisez la commande suivante :

```
$ aws iam update-role-description --role-name role-name --description description
```

Modification de la description d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM pour modifier la description d'un rôle lié à un service.

Pour changer la description d'un rôle lié à un service (API)

1. (Facultatif) Pour afficher la description actuelle d'un rôle, utilisez la commande suivante :

   GetRole
2. Pour mettre à jour la description d'un rôle, utilisez la commande suivante :

   UpdateRoleDescription

Suppression d'un rôle lié à un service pour RUM

Si vous n'avez plus de moniteur d'applications sur lequel X-Ray est activé, nous vous recommandons de supprimer le AWSServiceRoleForCloudWatchRUMrôle.

De cette façon, vous n’avez aucune entité inutilisée qui n’est pas surveillée ou gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer.

Nettoyage d'un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez d'abord vérifier qu'aucune session n'est active pour le rôle et supprimer toutes les ressources utilisées par le rôle.

Pour vérifier si une session est active pour le rôle lié à un service dans la console IAM

1. Ouvrez la console IAM à l’adresse https://console.aws.amazon.com/iam/.

2. Dans le panneau de navigation, sélectionnez Rôles. Choisissez le nom (et non la case à cocher) du AWSServiceRoleForCloudWatchRUMrôle.
3. Sur la page Summary (Résumé) du rôle sélectionné, choisissez Access Advisor et consultez l'activité récente du rôle lié au service.

> **Note**
>
> Si vous ne savez pas si RUM utilise le AWSServiceRoleForCloudWatchRUMrôle, essayez de le supprimer. Si le service utilise le rôle, la suppression échoue et vous avez accès aux régions dans lesquelles le rôle est utilisé. Si le rôle est utilisé, vous devez attendre que la session se termine avant de pouvoir le supprimer. Vous ne pouvez pas révoquer la session d'un rôle lié à un service.

Suppression d'un rôle lié à un service (console IAM)

Vous pouvez utiliser la console IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (console)

1. Ouvrez la console IAM à l’adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, sélectionnez Rôles. Cochez la case en regard du nom du rôle que vous souhaitez supprimer, sans sélectionner le nom ou la ligne.
3. Pour Role actions (Actions du rôle), choisissez Delete role (Supprimer le rôle).
4. Dans la boîte de dialogue de confirmation, vérifiez les dernières données consultées dans le service. Elles indiquent quels rôles, parmi ceux sélectionnés, ont accédé en dernier à un service AWS . Cela vous permet de confirmer si le rôle est actif actuellement. Pour poursuivre, choisissez Oui, supprimer.
5. Consultez les notifications de la console IAM pour surveiller la progression de la suppression du rôle lié à un service. Dans la mesure où la suppression du rôle lié à un service IAM est asynchrone, la suppression peut réussir ou échouer après que vous soumettez le rôle afin qu'il soit supprimé. Si la tâche échoue, choisissez View details (Afficher les détails) ou View Resources (Afficher les ressources) à partir des notifications pour connaître le motif de l'échec de la suppression. Si la suppression échoue parce que certaines ressources du service sont actuellement utilisées par le rôle, la raison de l'échec comprend une liste de ressources.

Suppression d'un rôle lié à un service (AWS CLI)

Vous pouvez utiliser les commandes IAM depuis le AWS Command Line Interface pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (AWS CLI)

1. Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `deletion-task-id` de la réponse afin de vérifier l'état de la tâche de suppression. Tapez la commande suivante pour envoyer une demande de suppression d'un rôle lié à un service :

   ```
   $ aws iam delete-service-linked-role --role-name service-linked-role-name
   ```

2. Tapez la commande suivante pour vérifier l'état de la tâche de suppression :

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   L'état de la tâche de suppression peut être `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

Suppression d'un rôle lié à un service (API IAM)

Vous pouvez utiliser l'API IAM pour supprimer un rôle lié à un service.

Pour supprimer un rôle lié à un service (API)

1. Pour soumettre une demande de suppression pour un rôle lié à un service, appelez. DeleteServiceLinkedRole Dans la demande, spécifiez le nom de rôle que vous souhaitez supprimer.

   Dans la mesure où un rôle lié à un service ne peut pas être supprimé s'il est utilisé ou si des ressources lui sont associées, vous devez envoyer une demande de suppression. Cette demande peut être refusée si ces conditions ne sont pas satisfaites. Vous devez capturer le `DeletionTaskId` de la réponse afin de vérifier l'état de la tâche de suppression.

2. Pour vérifier l'état de la suppression, appelez GetServiceLinkedRoleDeletionStatus. Dans la demande, spécifiez le `DeletionTaskId`.

   L'état de la tâche de suppression peut être `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Si la suppression échoue, l'appel renvoie le motif de l'échec, afin que vous puissiez apporter une solution.

Utilisation de rôles liés à un service pour Application Insights CloudWatch

CloudWatch Application Insights utilise des AWS Identity and Access Management rôles liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à CloudWatch Application Insights. Les rôles liés à un service sont prédéfinis par CloudWatch Application Insights 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 facilite la configuration CloudWatch d'Application Insights, car il n'est pas nécessaire d'ajouter manuellement les autorisations nécessaires. CloudWatch Application Insights définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seule CloudWatch Application Insights peut assumer ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés aux services, consultez Services AWS qui fonctionnent avec IAM et recherchez les services pour lesquels la colonne Service-Linked Role (Rôle lié aux services) indique Yes (Oui). Choisissez un lien Yes (Oui) pour consulter la documentation du rôle lié à ce service.

Autorisations de rôle liées au service pour Application Insights CloudWatch

CloudWatch Application Insights utilise le rôle lié au service nommé. AWSServiceRoleForApplicationInsights Application Insights utilise ce rôle pour effectuer des opérations telles que l'analyse des groupes de ressources du client, la création de CloudFormation piles pour créer des alarmes sur les métriques et la configuration de l' CloudWatch agent sur les instances EC2. Une politique IAM, nommée `CloudwatchApplicationInsightsServiceLinkedRolePolicy`, est associée à ce rôle lié à un service. Pour connaître les mises à jour de cette politique, consultez Mises à jour d'Application Insights pour les politiques gérées par AWS.

La politique d'autorisation des rôles permet à CloudWatch Application Insights d'effectuer les actions suivantes sur les ressources.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:DescribeAlarmHistory",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListMetrics",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DeleteAlarms",
        "cloudwatch:PutAnomalyDetector",
        "cloudwatch:DeleteAnomalyDetector",
        "cloudwatch:DescribeAnomalyDetectors"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:FilterLogEvents",
        "logs:GetLogEvents",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:CreateStack",
        "cloudFormation:UpdateStack",
        "cloudFormation:DeleteStack",
        "cloudFormation:DescribeStackResources"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudFormation:DescribeStacks",
        "cloudFormation:ListStackResources",
        "cloudFormation:ListStacks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:ListGroupResources",
        "resource-groups:GetGroupQuery",
        "resource-groups:GetGroup"
      ],
      "Resource": [
        "*"
      ]
    },
```

```
    {
      "Effect": "Allow",
      "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:DeleteGroup"
      ],
      "Resource": [
        "arn:aws:resource-groups:*:*:group/ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:DescribeAutoScalingGroups"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:PutParameter",
        "ssm:DeleteParameter",
        "ssm:AddTagsToResource",
        "ssm:RemoveTagsFromResource",
        "ssm:GetParameters"
      ],
      "Resource": "arn:aws:ssm:*:*:parameter/AmazonCloudWatch-ApplicationInsights-*"
    },
    {
      "Effect": "Allow",
      "Action": [
```

```
        "ssm:CreateAssociation",
        "ssm:UpdateAssociation",
        "ssm:DeleteAssociation",
        "ssm:DescribeAssociation"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:association/*",
        "arn:aws:ssm:*:*:managed-instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-
ApplicationInsightsCloudwatchAgentInstallAndConfigure",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetOpsItem",
        "ssm:CreateOpsItem",
        "ssm:DescribeOpsItems",
        "ssm:UpdateOpsItem",
        "ssm:DescribeInstanceInformation"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:AddTagsToResource"
      ],
      "Resource": "arn:aws:ssm:*:*:opsitem/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ssm:ListCommandInvocations",
        "ssm:GetCommandInvocation"
      ],
      "Resource": [
        "*"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": "ssm:SendCommand",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ssm:*:*:document/AWSEC2-CheckPerformanceCounterSets",
        "arn:aws:ssm:*:*:document/AWS-ConfigureAWSPackage",
        "arn:aws:ssm:*:*:document/AWSEC2-DetectWorkload",
        "arn:aws:ssm:*:*:document/AmazonCloudWatch-ManageAgent"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeNatGateways"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:ListFunctions",
        "lambda:GetFunctionConfiguration",
        "lambda:ListEventSourceMappings"
      ],
      "Resource": [
```

```
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "events:DeleteRule"
      ],
      "Resource": [
        "arn:aws:events:*:*:rule/AmazonCloudWatch-ApplicationInsights-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "xray:GetServiceGraph",
        "xray:GetTraceSummaries",
        "xray:GetTimeSeriesServiceStatistics",
        "xray:GetTraceGraph"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:ListTables",
        "dynamodb:DescribeTable",
        "dynamodb:DescribeContributorInsights",
        "dynamodb:DescribeTimeToLive"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:DescribeScalableTargets"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:GetMetricsConfiguration",
        "s3:GetReplicationConfiguration"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "states:ListStateMachines",
        "states:DescribeExecution",
        "states:DescribeStateMachine",
        "states:GetExecutionHistory"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "apigateway:GET"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters",
        "ecs:DescribeContainerInstances",
        "ecs:DescribeServices",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeTasks",
```

```
        "ecs:DescribeTaskSets",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListServices",
        "ecs:ListTasks"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecs:UpdateClusterSettings"
      ],
      "Resource": [
        "arn:aws:ecs:*:*:cluster/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "eks:DescribeCluster",
        "eks:DescribeFargateProfile",
        "eks:DescribeNodegroup",
        "eks:ListClusters",
        "eks:ListFargateProfiles",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "fsx:DescribeVolumes"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:GetSubscriptionAttributes",
        "sns:GetTopicAttributes",
        "sns:GetSMSAttributes",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics"
      ],
```

```
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:ListQueues"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:DeleteSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutSubscriptionFilter"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:*",
        "arn:aws:logs:*:*:destination:AmazonCloudWatch-ApplicationInsights-
LogIngestionDestination*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53:GetHostedZone",
```

```
        "route53:GetHealthCheck",
        "route53:ListHostedZones",
        "route53:ListHealthChecks",
        "route53:ListQueryLoggingConfigs"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "route53resolver:ListFirewallRuleGroupAssociations",
        "route53resolver:GetFirewallRuleGroup",
        "route53resolver:ListFirewallRuleGroups",
        "route53resolver:ListResolverEndpoints",
        "route53resolver:GetResolverQueryLogConfig",
        "route53resolver:ListResolverQueryLogConfigs",
        "route53resolver:ListResolverQueryLogConfigAssociations",
        "route53resolver:GetResolverEndpoint",
        "route53resolver:GetFirewallRuleGroupAssociation"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Service-Linked Role Permissions (autorisations du rôle lié à un service) dans le IAM User Guide (guide de l'utilisateur IAM).

Création d'un rôle lié à un service pour Application Insights CloudWatch

Vous n’avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez une nouvelle application Application Insights dans le AWS Management Console, CloudWatch Application Insights crée pour vous le rôle lié au service.

Si vous supprimez ce rôle lié à un service et que vous souhaitez ensuite le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous créez une nouvelle

application Application Insights, CloudWatch Application Insights crée à nouveau le rôle lié au service pour vous.

Modification d'un rôle lié à un service pour Application Insights CloudWatch

CloudWatch Application Insights ne vous permet pas de modifier le rôle AWSServiceRoleForApplicationInsights lié au service. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez Modification d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Supprimer un rôle lié à un service pour Application Insights 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 évitez d'avoir une entité inutilisée non surveillée ou non gérée activement. Vous devez cependant supprimer toutes les applications dans Application Insights avant de pouvoir supprimer le rôle manuellement.

> **Note**
>
> Si le service CloudWatch Application Insights 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 Application Insights utilisées par AWSServiceRoleForApplicationInsights

- Supprimez toutes vos applications CloudWatch Application Insights. Pour plus d'informations, consultez la section « Supprimer vos applications » dans le guide de l'utilisateur d' CloudWatch Application Insights.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, le AWS CLI, ou l' AWS API pour supprimer le rôle lié au AWSServiceRoleForApplicationInsights service. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les CloudWatch rôles liés au service Application Insights

CloudWatch Application Insights 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 Régions et points de terminaison d'CloudWatch Application Insights.

AWS politiques gérées pour Amazon CloudWatch Application Insights

Une politique AWS gérée est une politique autonome créée et administrée par AWS. AWS les politiques gérées sont conçues pour fournir des autorisations pour de nombreux cas d'utilisation courants afin que vous puissiez commencer à attribuer des autorisations aux utilisateurs, aux groupes et aux rôles.

N'oubliez pas que les politiques AWS gérées peuvent ne pas accorder d'autorisations de moindre privilège pour vos cas d'utilisation spécifiques, car elles sont accessibles à tous les AWS clients. Nous vous recommandons de réduire encore les autorisations en définissant des politiques gérées par le client qui sont propres à vos cas d'utilisation.

Vous ne pouvez pas modifier les autorisations définies dans les politiques AWS gérées. Si les autorisations définies dans une politique AWS gérée sont AWS mises à jour, la mise à jour affecte toutes les identités principales (utilisateurs, groupes et rôles) auxquelles la politique est attachée. AWS est le plus susceptible de mettre à jour une politique AWS gérée lorsqu'une nouvelle politique Service AWS est lancée ou lorsque de nouvelles opérations d'API sont disponibles pour les services existants.

Pour plus d'informations, consultez la section Politiques gérées par AWS dans le Guide de l'utilisateur IAM.

AWS politique gérée : CloudWatchApplicationInsightsFullAccess

Vous pouvez associer la politique `CloudWatchApplicationInsightsFullAccess` à vos identités IAM.

Cette politique accorde des autorisations administratives qui permettent un accès complet à la fonctionnalité Application Insights.

Détails de l’autorisation

Cette politique inclut les autorisations suivantes.

- `applicationinsights` – Permet un accès complet à la fonctionnalité Application Insights.
- `iam`— Permet à Application Insights de créer le rôle lié au service,. AWSServiceRoleForApplicationInsights Cela est nécessaire pour qu'Application Insights puisse effectuer des opérations telles que l'analyse des groupes de ressources d'un client, la création de CloudFormation piles pour créer des alarmes sur les métriques et la configuration de l' CloudWatchagent sur les instances EC2. Pour plus d’informations, consultez Utilisation de rôles liés à un service pour Application Insights CloudWatch .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "applicationinsights:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeVolumes",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "sqs:ListQueues",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTargetGroups",
        "elasticloadbalancing:DescribeTargetHealth",
        "autoscaling:DescribeAutoScalingGroups",
        "lambda:ListFunctions",
        "dynamodb:ListTables",
```

```
        "s3:ListAllMyBuckets",
        "sns:ListTopics",
        "states:ListStateMachines",
        "apigateway:GET",
        "ecs:ListClusters",
        "ecs:DescribeTaskDefinition",
        "ecs:ListServices",
        "ecs:ListTasks",
        "eks:ListClusters",
        "eks:ListNodegroups",
        "fsx:DescribeFileSystems",
        "logs:DescribeLogGroups",
        "elasticfilesystem:DescribeFileSystems"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/aws-service-role/application-insights.amazonaws.com/
AWSServiceRoleForApplicationInsights"
      ],
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "application-insights.amazonaws.com"
        }
      }
    }
  ]
}
```

AWS politique gérée : CloudWatchApplicationInsightsReadOnlyAccess

Vous pouvez associer la politique `CloudWatchApplicationInsightsReadOnlyAccess` à vos identités IAM.

Cette politique accorde des autorisations administratives qui permettent un accès en lecture seule à toutes les fonctionnalités d'Application Insights.

Détails de l’autorisation

Cette politique inclut les autorisations suivantes.

- `applicationinsights` – Autorise l'accès en lecture seule à la fonctionnalité Application Insights.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "applicationinsights:Describe*",
                "applicationinsights:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS politique gérée : CloudwatchApplicationInsightsServiceLinkedRolePolicy

Vous ne pouvez pas vous associer CloudwatchApplicationInsightsServiceLinkedRolePolicy à vos entités IAM. Cette politique est associée à un rôle lié au service qui permet à Application Insights de surveiller les ressources du client. Pour plus d'informations, consultez Utilisation de rôles liés à un service pour Application Insights CloudWatch .

Mises à jour d'Application Insights pour les politiques gérées par AWS

Consultez les détails des mises à jour apportées aux politiques AWS gérées pour Application Insights depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS de la page Historique du document d'Application Insights.

Modification	Description	Date
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une stratégie existante	Application Insights a ajouté de nouvelles autorisations pour répertorier les CloudFormation piles. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse analyser et surveiller les AWS ressources imbriquées dans la CloudFormation pile.	24 avril 2023
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour obtenir la liste des ressources Amazon VPC et Route 53. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse configurer automatiquement la surveillance du réseau selon les meilleures pratiques Amazon CloudWatch.	23 janvier 2023
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour obtenir les résultats d'invocation de commandes SSM. Ces autorisations sont requises pour qu'Amazon	19 décembre 2022

Modification	Description	Date
	CloudWatch Application Insights détecte et surveille automatiquement les charges de travail exécutées sur les instances Amazon EC2.	
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire les ressources Amazon VPC et Route 53. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse lire les configurations de ressources Amazon VPC et Route 53 des clients, et pour aider les clients à configurer automatiquement les meilleures pratiques de surveillance du réseau avec. Amazon CloudWatch	19 décembre 2022

Modification	Description	Date
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire les ressources EFS. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse lire les configurations des ressources client Amazon EFS et pour aider les clients à définir automatiquement les meilleures pratiques en matière de surveillance EFS avec CloudWatch.	3 octobre 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire le système de fichiers EFS. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse créer des applications basées sur des comptes en interrogeant toutes les ressources prises en charge dans un compte.	3 octobre 2022

Modification	Description	Date
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d’une politique existante	Application Insights a ajouté de nouvelles autorisations pour récupérer des informations sur les ressources FSx. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse surveiller les charges de travail en récupérant suffisamment d'informations sur les volumes FSx sous-jacents.	12 septembre 2022
AWS politique gérée : CloudWatchApplicationInsightsFullAccess – Mise à jour d’une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire les groupes de journaux. Ces autorisations sont requises pour Amazon CloudWatch Application Insights afin de garantir que les autorisations appropriées pour surveiller les groupes de journaux figurent dans un compte lors de la création d'une nouvelle application.	24 janvier 2022

Modification	Description	Date
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d’une politique existante	Application Insights a ajouté de nouvelles autorisations pour créer et supprimer des filtres d'abonnement aux CloudWatch journaux. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse créer des filtres d'abonnement afin de faciliter la surveillance des journaux des ressources au sein des applications configurées.	24 janvier 2022
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d’une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire les groupes cibles et l'état de santé des cibles pour Elastic Load Balancers. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse créer des applications basées sur des comptes en interrogeant toutes les ressources prises en charge dans un compte.	4 novembre 2021

Modification	Description	Date
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d’une politique existante	Application Insights a ajouté de nouvelles autorisations pour exécuter le`AmazonCloudWatch-ManageAgent` Document SSM sur les instances Amazon EC2. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse nettoyer les fichiers de configuration des CloudWatch agents créés par Application Insights.	30 septembre 2021

Modification	Description	Date
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d’une politique existante	Application Insights a ajouté de nouvelles autorisations pour prendre en charge la surveillance des applications basée sur un compte à bord et surveiller toutes les ressources prises en charge dans votre compte. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse interroger, étiqueter des ressources et créer des groupes pour ces ressources. Application Insights a ajouté de nouvelles autorisations pour prendre en charge la surveillance des rubriques SNS. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights collecte des métadonnées à partir des ressources SNS afin de configurer la surveillance des sujets SNS.	15 septembre 2021

Modification	Description	Date
AWS politique gérée : CloudWatchApplicationInsightsFullAccess – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire et répertorier les ressources de service ECS et EKS. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse créer des applications basées sur des comptes en interrogeant toutes les ressources prises en charge dans un compte.	15 septembre 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire les ressources FSx. Ces autorisations sont requises pour qu'Amazon CloudWatch Application Insights puisse lire les configurations des ressources FSx des clients et pour aider les clients à configurer automatiquement les meilleures pratiques de surveillance de FSx avec. CloudWatch	31 août 2021

Modification	Description	Date
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations pour décrire et répertorier les ressources de service ECS et EKS. Cette autorisation est requise pour qu'Amazon CloudWatch Application Insights puisse lire la configuration des ressources des conteneurs des clients et pour aider les clients à configurer automatiquement les meilleures pratiques de surveillance des conteneurs avec CloudWatch.	18 mai 2021
CloudwatchApplicationInsightsServiceLinkedRolePolicy – Mise à jour d'une politique existante	Application Insights a ajouté de nouvelles autorisations permettant OpsCenter de baliser le type de ressource à l' OpsItemsaide de l'`ssm:AddTagsToResource` action sur les `opsitem` ressources. Cette autorisation est requise par OpsCenter. Amazon CloudWatch Application Insights crée OpsItems pour que le client puisse résoudre les problèmes à l'aide du AWS SSM OpsCenter.	13 avril 2021

Modification	Description	Date
Application Insights a commencé à suivre les modifications	Application Insights a commencé à suivre les modifications apportées AWS à ses politiques gérées.	13 avril 2021

Référence CloudWatch des autorisations Amazon

Le tableau suivant répertorie chaque opération CloudWatch d'API 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, et ajoutez un caractère générique (*) comme valeur de ressource dans le champ `Resource`.

Vous pouvez utiliser des AWS clés de condition larges dans vos CloudWatch polices 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 `cloudwatch:` suivi du nom de l'opération d'API. Par exemple : `cloudwatch:GetMetricData`, `cloudwatch:ListMetrics` ou `cloudwatch:*` (pour toutes les actions CloudWatch).

Rubriques

- CloudWatch Opérations d'API et autorisations requises pour les actions
- CloudWatch Opérations de l'API Contributor Insights et autorisations requises pour les actions
- CloudWatch Opérations de l'API d'événements et autorisations requises pour les actions
- CloudWatch Enregistre les opérations de l'API et les autorisations requises pour les actions
- Opérations d'API Amazon EC2 et autorisations requises pour les actions
- Opérations d'API Amazon EC2 Auto Scaling et autorisations requises pour les actions

CloudWatch Opérations d'API et autorisations requises pour les actions

CloudWatch Opérations d'API	Autorisations requises (actions d'API)
DeleteAlarms	`cloudwatch:DeleteAlarms` Exigé pour supprimer une alarme.
DeleteDashboards	`cloudwatch:DeleteDashboards` Exigé pour supprimer un tableau de bord.
DeleteMetricStream	`cloudwatch:DeleteMetricStream` Exigé pour supprimer un flux de métriques.
DescribeAlarmHistory	`cloudwatch:DescribeAlarmHistory` Exigé pour afficher l'historique de l'alarme. Pour récupérer des informations sur les alarmes composites, votre autorisation `cloudwatch:DescribeAlarmHistory` doit avoir une portée `*`. Vous ne pouvez pas renvoyer d'informations sur les alarmes composites si votre autorisation `cloudwatch:DescribeAlarmHistory` a une portée plus étroite.
DescribeAlarms	`cloudwatch:DescribeAlarms` Exigé pour récupérer des informations sur des alarmes. Pour récupérer des informations sur les alarmes composites, votre autorisation `cloudwatch:DescribeAlarms` doit avoir une portée `*`. Vous ne pouvez pas renvoyer

CloudWatch Opérations d'API	Autorisations requises (actions d'API)
	d'informations sur les alarmes composites si votre autorisation `cloudwatch:DescribeAlarms` a une portée plus étroite.
DescribeAlarmsForMetric	`cloudwatch:DescribeAlarmsForMetric` Exigé pour afficher les alarmes relatives à une métrique.
DisableAlarmActions	`cloudwatch:DisableAlarmActions` Exigé pour désactiver une action d'alarme.
EnableAlarmActions	`cloudwatch:EnableAlarmActions` Exigé pour activer une action d'alarme.
GetDashboard	`cloudwatch:GetDashboard` Exigé pour afficher les données sur les tableaux de bord existants.
GetMetricData	`cloudwatch:GetMetricData` Nécessaire pour représenter graphiquement les données métriques dans la CloudWatch console, pour récupérer de gros lots de données métriques et pour effectuer des calculs métriques sur ces données.

CloudWatch Opérations d'API	Autorisations requises (actions d'API)
GetMetricStatistics	`cloudwatch:GetMetricStatistics` Nécessaire pour afficher des graphiques dans d'autres parties de la CloudWatch console et dans les widgets du tableau de bord.
GetMetricStream	`cloudwatch:GetMetricStream` Requise pour afficher les informations sur un flux de métriques.
GetMetricWidgetImage	`cloudwatch:GetMetricWidgetImage` Nécessaire pour récupérer un graphique instantané d'une ou plusieurs CloudWatch métriques sous forme d'image bitmap.
ListDashboards	`cloudwatch:ListDashboards` Nécessaire pour consulter la liste des CloudWatch tableaux de bord de votre compte.
ListMetrics	`cloudwatch:ListMetrics` Nécessaire pour afficher ou rechercher des noms de métriques dans la CloudWatch console et dans la CLI. Exigé pour sélection ner les métriques sur les widgets de tableau de bord.
ListMetricStreams	`cloudwatch:ListMetricStreams` Exigé pour afficher ou rechercher la liste des flux de métriques dans le compte.

CloudWatch Opérations d'API	Autorisations requises (actions d'API)
PutCompositeAlarm	`cloudwatch:PutCompositeAlarm` Exigé pour créer une alarme composite. Pour créer une alarme composite, votre autorisation `cloudwatch:PutCompositeAlarm` doit avoir une portée `*`. Vous ne pouvez pas renvoyer d'informations sur les alarmes composites si votre autorisation `cloudwatch:PutCompositeAlarm` a une portée plus étroite.
PutDashboard	`cloudwatch:PutDashboard` Exigé pour créer un tableau de bord ou mettre à jour un tableau de bord existant.
PutMetricAlarm	`cloudwatch:PutMetricAlarm` Exigé pour créer ou mettre à jour une alarme.
PutMetricData	`cloudwatch:PutMetricData` Exigé pour créer des métriques.
PutMetricStream	`cloudwatch:PutMetricStream` Exigé pour créer un flux de métriques.
SetAlarmState	`cloudwatch:SetAlarmState` Exigé pour définir manuellement un état d'alarme.

CloudWatch Opérations d'API	Autorisations requises (actions d'API)
StartMetricStreams	`cloudwatch:StartMetricStreams` Exigé pour démarrer le flux de métriques dans un flux de métriques.
StopMetricStreams	`cloudwatch:StopMetricStreams` Exigé pour arrêter temporairement le flux de métriques dans un flux de métriques.
TagResource	`cloudwatch:TagResource` Nécessaire pour ajouter ou mettre à jour des balises sur CloudWatch des ressources telles que les alarmes et les règles Contributor Insights.
UntagResource	`cloudwatch:UntagResource` Nécessaire pour supprimer les balises des CloudWatch ressources.

CloudWatch Opérations de l'API Contributor Insights et autorisations requises pour les actions

> **Important**
>
> Lorsque vous accordez l'`cloudwatch:PutInsightRule`autorisation à un utilisateur, celui-ci peut par défaut créer une règle qui évalue n'importe quel groupe de CloudWatch journaux dans Logs. Vous pouvez ajouter des conditions de politique IAM qui limitent ces autorisations pour qu'un utilisateur inclue et exclue des groupes de journaux spécifiques. Pour plus d'informations, consultez Utilisation de clés de condition pour limiter l'accès des utilisateurs Contributor Insights aux groupes de journaux.

CloudWatch Opérations de l'API Contributor Insights	Autorisations requises (actions d'API)
DeleteInsightRules	`cloudwatch:DeleteInsightRules` Exigé pour supprimer les règles Contributor Insights.
DescribeInsightRules	`cloudwatch:DescribeInsightRules` Exigé pour afficher les règles de Contributor Insights dans votre compte.
EnableInsightRules	`cloudwatch:EnableInsightRules` Exigé pour activer les règles Contributor Insights.
GetInsightRuleReport	`cloudwatch:GetInsightRuleReport` Exigé pour récupérer des données de séries chronologiques et d'autres statistiques collectée s par les règles Contributor Insights.
PutInsightRule	`cloudwatch:PutInsightRule` Exigé pour créer les règles Contributor Insights. Consultez la remarque importante au début de ce tableau.

CloudWatch Opérations de l'API d'événements et autorisations requises pour les actions

CloudWatch Opérations de l'API Events	Autorisations requises (actions d'API)

CloudWatch Opérations de l'API Events	Autorisations requises (actions d'API)
DeleteRule	`events:DeleteRule` Requise pour supprimer une règle.
DescribeRule	`events:DescribeRule` Requise pour répertorier les détails relatifs à une règle.
DisableRule	`events:DisableRule` Requise pour désactiver une règle.
EnableRule	`events:EnableRule` Requise pour activer une règle.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget` Requise pour répertorier les règles associées à une cible.
ListRules	`events:ListRules` Requise pour répertorier toutes les règles de votre compte.
ListTargetsByRule	`events:ListTargetsByRule` Requise pour afficher toutes les cibles associées à une règle.

CloudWatch Opérations de l'API Events	Autorisations requises (actions d'API)
PutEvents	`events:PutEvents` Requise pour ajouter des événements personnalisés qui peuvent être associés à des règles.
PutRule	`events:PutRule` Requise pour créer ou mettre à jour une règle.
PutTargets	`events:PutTargets` Requise pour ajouter des cibles à une règle.
RemoveTargets	`events:RemoveTargets` Requise pour supprimer une cible d'une règle.
TestEventPattern	`events:TestEventPattern` Requise pour tester un modèle d'événement par rapport à un événement donné.

CloudWatch Enregistre les opérations de l'API et les autorisations requises pour les actions

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
CancelExportTask	`logs:CancelExportTask` Exigé pour annuler une tâche d'exportation en attente ou en cours d'exécution.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
CreateExportTask	`logs:CreateExportTask` Exigé pour exporter des données d'un groupe de journaux vers un compartiment Amazon S3.
CreateLogGroup	`logs:CreateLogGroup` Exigé pour créer un nouveau groupe de journaux.
CreateLogStream	`logs:CreateLogStream` Exigé pour créer un nouveau flux de journaux dans un groupe de journaux.
DeleteDestination	`logs:DeleteDestination` Exigé pour supprimer une destination de journal et désactive tous les filtres d'abonnem ent connexes.
DeleteLogGroup	`logs:DeleteLogGroup` Exigé pour supprimer un groupe de journaux et tous les événements du journal archivés associés.
DeleteLogStream	`logs:DeleteLogStream` Exigé pour supprimer un flux de journaux et tous les événements du journal archivés associés.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
DeleteMetricFilter	`logs:DeleteMetricFilter` Exigé pour supprimer un filtre de métrique associé à un groupe de journaux.
DeleteQueryDefinition	`logs:DeleteQueryDefinition` Nécessaire pour supprimer une définition de requête enregistrée dans CloudWatch Logs Insights.
DeleteResourcePolicy	`logs:DeleteResourcePolicy` Nécessaire pour supprimer une politique de ressources CloudWatch Logs.
DeleteRetentionPolicy	`logs:DeleteRetentionPolicy` Exigé pour supprimer la politique de rétention d'un groupe de journaux.
DeleteSubscriptionFilter	`logs:DeleteSubscriptionFilter` Exigé pour supprimer le filtre d'abonnement associé à un groupe de journaux.
DescribeDestinations	`logs:DescribeDestinations` Exigé pour afficher toutes les destinations associées au compte.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
DescribeExportTasks	`logs:DescribeExportTasks` Exigé pour afficher toutes les tâches d'exportation associées au compte.
DescribeLogGroups	`logs:DescribeLogGroups` Exigé pour afficher tous les groupes de journaux associés au compte.
DescribeLogStreams	`logs:DescribeLogStreams` Exigé pour afficher tous les flux de journaux associés à un groupe de journaux.
DescribeMetricFilters	`logs:DescribeMetricFilters` Exigé pour afficher toutes les métriques associées à un groupe de journaux.
DescribeQueryDefinitions	`logs:DescribeQueryDefinitions` Nécessaire pour voir la liste des définitions de requêtes enregistrées dans CloudWatch Logs Insights.
DescribeQueries	`logs:DescribeQueries` Obligatoire pour voir la liste des requêtes CloudWatch Logs Insights planifiées, en cours d'exécution ou récemment exécutées.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
DescribeResourcePolicies	`logs:DescribeResourcePolicies` Obligatoire pour consulter la liste des politiques relatives CloudWatch aux ressources des journaux.
DescribeSubscriptionFilters	`logs:DescribeSubscriptionFilters` Exigé pour afficher tous les filtres d'abonnement associés à un groupe de journaux.
FilterLogEvents	`logs:FilterLogEvents` Exigé pour trier les événements du journal par modèle de filtres de groupes de journaux.
GetLogEvents	`logs:GetLogEvents` Exigé pour récupérer les événements du journal à partir d'un flux de journaux.
GetLogGroupFields	`logs:GetLogGroupFields` Obligatoire pour récupérer la liste des champs qui sont inclus dans les événements du journal d'un groupe de journaux.
GetLogRecord	`logs:GetLogRecord` Obligatoire pour récupérer des informations à partir d'un seul événement du journal.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
GetQueryResults	`logs:GetQueryResults` Nécessaire pour récupérer les résultats des requêtes CloudWatch Logs Insights.
ListTagsLogGroup	`logs:ListTagsLogGroup` Exigé pour afficher toutes les étiquettes associées à un groupe de journaux.
PutDestination	`logs:PutDestination` Exigé pour créer ou mettre à jour un flux de journaux de destination (comme un flux Kinesis).
PutDestinationPolicy	`logs:PutDestinationPolicy` Exigé pour créer ou mettre à jour une politique d'accès associée à une destination de journal existante.
PutLogEvents	`logs:PutLogEvents` Exigé pour charger un lot d'événements du journal dans un flux de journaux.
PutMetricFilter	`logs:PutMetricFilter` Exigé pour créer ou mettre à jour un filtre de métrique et l'associer à un groupe de journaux.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
PutQueryDefinition	`logs:PutQueryDefinition` Nécessaire pour enregistrer une requête dans CloudWatch Logs Insights.
PutResourcePolicy	`logs:PutResourcePolicy` Nécessaire pour créer une politique de ressources CloudWatch Logs.
PutRetentionPolicy	`logs:PutRetentionPolicy` Exigé pour définir le nombre de jours de conservation des événements du journal (rétention) dans un groupe de journaux.
PutSubscriptionFilter	`logs:PutSubscriptionFilter` Exigé pour créer ou mettre à jour un filtre d'abonnement et l'associer à un groupe de journaux.
StartQuery	`logs:StartQuery` Nécessaire pour démarrer CloudWatch les requêtes Logs Insights.
StopQuery	`logs:StopQuery` Nécessaire pour arrêter une requête CloudWatch Logs Insights en cours.

CloudWatch Journalise les opérations de l'API	Autorisations requises (actions d'API)
TagLogGroup	`logs:TagLogGroup` Exigé pour ajouter ou mettre à jour des étiquettes de groupe de journaux.
TestMetricFilter	`logs:TestMetricFilter` Exigé pour tester un modèle de filtre par rapport à un échantillonnage de messages d'événements du journal.

Opérations d'API Amazon EC2 et autorisations requises pour les actions

Opérations d'API Amazon EC2	Autorisations requises (actions d'API)
DescribeInstanceStatus	`ec2:DescribeInstanceStatus` Exigé pour afficher les détails sur l'état d'une instance EC2.
DescribeInstances	`ec2:DescribeInstances` Exigé pour afficher les détails sur une instance EC2.
RebootInstances	`ec2:RebootInstances` Exigé pour redémarrer une instance EC2.
StopInstances	`ec2:StopInstances` Exigé pour arrêter une instance EC2.

Opérations d'API Amazon EC2	Autorisations requises (actions d'API)
TerminateInstances	`ec2:TerminateInstances` Exigé pour mettre fin à une instance EC2.

Opérations d'API Amazon EC2 Auto Scaling et autorisations requises pour les actions

Opérations d'API Amazon EC2 Auto Scaling	Autorisations requises (actions d'API)
Mise à l'échelle	`autoscaling:Scaling` Exigé pour dimensionner un groupe Auto Scaling.
Déclencheur	`autoscaling:Trigger` Exigé pour déclencher une action Auto Scaling.

Validation de conformité pour Amazon CloudWatch

Des auditeurs tiers évaluent la sécurité et la conformité d'Amazon dans CloudWatch le cadre de plusieurs programmes de AWS conformité. Il s'agit notamment des certifications SOC, PCI, FedRAMP, HIPAA et d'autres.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir AWS Services concernés par programme de conformitéAWS . Pour obtenir des renseignements généraux, consultez Programmes de conformitéAWS .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Lorsque vous utilisez Amazon CloudWatch , votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides démarrage rapide de la sécurité et de la conformité. Ces guides de déploiement traitent des considérations architecturales et fournissent des étapes pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA — Ce livre blanc décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- Évaluation des ressources à l'aide des règles énoncées dans le guide du AWS Config développeur : AWS Configévalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Résilience chez Amazon CloudWatch

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 sur Amazon CloudWatch

En tant que service géré, Amazon CloudWatch 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 CloudWatch via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Isolement de réseau

Un cloud privé virtuel (VPC) est un réseau virtuel situé dans votre propre zone logiquement isolée du cloud Amazon Web Services. Un sous-réseau est une plage d'adresses IP dans un VPC. Vous pouvez déployer diverses ressources AWS dans les sous-réseaux de vos VPC. Par exemple, vous pouvez déployer des instances Amazon EC2, des clusters EMR et des tables de DynamoDB dans des sous-réseaux. Pour de plus amples informations, consultez le Guide de l'utilisateur Amazon VPC.

Pour CloudWatch permettre de communiquer avec les ressources d'un VPC sans passer par l'Internet public, utilisez. AWS PrivateLink Pour plus d'informations, consultez Utilisation CloudWatch et CloudWatch synthèse des points de terminaison VPC d'interface.

Un sous-réseau privé est un sous-réseau sans routage par défaut vers le réseau Internet public. Le déploiement d'une AWS ressource dans un sous-réseau privé n'empêche pas Amazon CloudWatch de collecter des métriques intégrées à partir de la ressource.

Si vous devez publier des métriques personnalisées à partir d'une AWS ressource d'un sous-réseau privé, vous pouvez le faire à l'aide d'un serveur proxy. Le serveur proxy transmet ces demandes HTTPS aux points de terminaison de l'API publics pour CloudWatch.

AWS Security Hub

Surveillez votre utilisation CloudWatch en ce qui concerne les meilleures pratiques de sécurité à l'aide AWS de Security Hub. Security Hub utilise des contrôles de sécurité pour évaluer les

configurations des ressources et les normes de sécurité afin de vous aider à respecter divers cadres de conformité. Pour plus d'informations sur l'utilisation de Security Hub pour évaluer les CloudWatch ressources, consultez Amazon CloudWatch Controls dans le Guide de l'utilisateur du AWS Security Hub.

Utilisation CloudWatch et CloudWatch synthèse des points de terminaison VPC d'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 Synthetics. CloudWatch CloudWatch Vous pouvez utiliser ces connexions pour permettre CloudWatch à CloudWatch Synthetics de communiquer avec vos ressources sur votre VPC sans passer par l'Internet public.

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 à CloudWatch Synthetics CloudWatch , vous devez définir un point de terminaison VPC d'interface pour connecter votre VPC aux services. AWS Le point de terminaison fournit une connectivité fiable et évolutive à CloudWatch Synthetics sans nécessiter de passerelle Internet, d'instance de traduction d'adresses réseau (NAT) CloudWatch ou de connexion VPN. Pour de plus amples informations, veuillez consulter 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, consultez le billet de blog New — AWS PrivateLink for AWS Services.

Les étapes suivantes s'adressent aux utilisateurs d'Amazon VPC. Pour plus d'informations, consultez Démarrez dans le Amazon VPC Guide de l'utilisateur.

CloudWatch Point de terminaison VPC

CloudWatch prend actuellement en charge les points de terminaison VPC dans les régions suivantes : AWS

- USA Est (Ohio)

- USA Est (Virginie du Nord)
- USA Ouest (Californie du Nord)
- US West (Oregon)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Paris)
- Moyen-Orient (EAU)
- Amérique du Sud (São Paulo)
- AWS GovCloud (USA Est)
- AWS GovCloud (US-Ouest)

Création d'un point de terminaison VPC pour CloudWatch

Pour commencer à utiliser CloudWatch avec votre VPC, créez un point de terminaison VPC d'interface pour. CloudWatch Le nom du service à choisir est `com.amazonaws.`*`region`*`.monitoring`. Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Amazon VPC Guide de l'utilisateur.

Il n'est pas nécessaire de modifier les paramètres de CloudWatch. CloudWatch appelle d'autres AWS services en utilisant des points de terminaison publics ou des points de terminaison VPC d'interface privée, selon ceux utilisés. Par exemple, si vous créez un point de terminaison VPC d'interface pour CloudWatch et que vous disposez déjà de métriques CloudWatch provenant de ressources situées sur votre VPC, ces métriques commencent à passer par le point de terminaison VPC d'interface par défaut.

Contrôle de l'accès à votre point de CloudWatch terminaison 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 n'attachez pas de stratégie quand vous créez un point de terminaison, Amazon VPC attache une stratégie par défaut pour vous qui autorise un accès total au service. Une stratégie de point de terminaison n'annule pas et ne remplace pas les stratégies utilisateur ou les stratégies propres au service. 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. Cette politique permet aux utilisateurs qui se connectent CloudWatch via le VPC d'envoyer des données métriques CloudWatch et les empêche d'effectuer d'autres CloudWatch actions.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
      "Principal": "*",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Pour modifier la politique de point de terminaison VPC pour 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, choisissez Create Endpoint. Sélectionnez com.amazonaws.***region***.monitoring et choisissez Create endpoint (Créer un point de terminaison).

4. Sélectionnez le point de terminaison com.amazonaws.*region*.monitoring, puis choisissez l'onglet Policy (Politique).
5. Choisissez Edit Policy (Modifier la politique), puis apportez vos modifications.

CloudWatch Point de terminaison VPC Synthetics

CloudWatch Synthetics prend actuellement en charge les points de terminaison VPC dans les régions suivantes : AWS

- USA Est (Ohio)
- USA Est (Virginie du Nord)
- USA Ouest (Californie du Nord)
- US West (Oregon)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Paris)
- Amérique du Sud (São Paulo)

Création d'un point de terminaison VPC pour Synthetics CloudWatch

Pour commencer à utiliser CloudWatch Synthetics avec votre VPC, créez un point de terminaison VPC d'interface pour Synthetics. CloudWatch Le nom du service à choisir est `com.amazonaws.region.synthetics`. Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Amazon VPC Guide de l'utilisateur.

Il n'est pas nécessaire de modifier les paramètres des CloudWatch Synthetics. CloudWatch Synthetics communique avec AWS d'autres services à l'aide de points de terminaison publics ou de points de terminaison VPC d'interface privée, selon ceux utilisés. Par exemple, si vous créez un point de terminaison VPC d'interface pour Synthetics CloudWatch et que vous possédez déjà un point de terminaison d'interface pour Amazon S3, Synthetics CloudWatch commence à communiquer avec Amazon S3 via le point de terminaison VPC d'interface par défaut.

Contrôle de l'accès à votre point de CloudWatch terminaison Synthetics 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 stratégie de point de terminaison n'annule pas et ne remplace pas les stratégies utilisateur ou les stratégies propres au service. Il s'agit d'une politique distincte qui contrôle l'accès depuis le point de terminaison jusqu'au service spécifié.

Les stratégies de point de terminaison affectent les Canary gérés en privé par VPC. Ils ne sont pas nécessaires pour les Canary qui fonctionnent sur des sous-réseaux privés.

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 Synthetics. Cette politique permet aux utilisateurs qui se connectent à CloudWatch Synthetics via le VPC de consulter des informations sur les canaris et leurs courses, mais pas de créer, de modifier ou de supprimer des canaris.

```
{
    "Statement": [
        {
            "Action": [
                "synthetics:DescribeCanaries",
                "synthetics:GetCanaryRuns"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        }
```

```
    ]
}
```

Pour modifier la politique de point de terminaison VPC pour Synthetics 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 Synthetics, choisissez Create Endpoint. Sélectionnez com.amazonaws.***region***.synthetics, puis choisissez Create endpoint (Créer un point de terminaison).
4. Sélectionnez le point de terminaison com.amazonaws.***region***.synthetics, puis choisissez l'onglet Policy (Politique).
5. Choisissez Edit Policy (Modifier la politique), puis apportez vos modifications.

Considérations de sécurité pour les scripts Canary Synthetics

Les sections suivantes expliquent les problèmes de sécurité que vous devez prendre en compte lors de la création et de l'exécution de scripts Canary dans Synthetics.

Utiliser des connexions sécurisées

Étant donné que le code du script Canary et les résultats des tests peuvent contenir des informations sensibles, ne vous connectez pas aux points de terminaison via des connexions non chiffrées. Utilisez toujours des connexions chiffrées, telles que celles qui commencent par `https://`.

Considérations relatives à la dénomination des scripts Canary

Le nom de ressource Amazon (ARN) d'un canari est inclus dans l'en-tête de l'agent utilisateur dans le cadre des appels sortants effectués depuis le navigateur Chromium piloté par Puppeteer et inclus dans la bibliothèque de wrappers Synthetics. CloudWatch Cela permet d'identifier le trafic CloudWatch des canaris de Synthetics et de le relier aux canaris qui passent des appels.

L'ARN du script Canary inclut le nom du script. Choisissez des noms de scripts Canary qui ne révèlent pas d'informations exclusives.

En outre, assurez-vous de pointer vos scripts Canary uniquement vers les sites web et les points de terminaison que vous contrôlez.

Secrets et informations sensibles dans le code canary

Si vous transmettez votre code Canary directement au Canary à l'aide d'un fichier zip, le contenu du script est visible dans AWS CloudTrail les journaux.

Si vous avez des informations sensibles ou des secrets (tels que des clés d'accès ou des informations d'identification de base de données) dans un script canary, nous vous recommandons fortement de stocker le script comme un objet versionné dans Amazon S3 et de transmettre l'emplacement Amazon S3 dans le canary, au lieu de transmettre le code canary par un fichier zip.

Si vous utilisez un fichier zip pour transmettre le script canary, nous vous recommandons fortement de ne pas inclure de secrets ou d'informations sensibles dans votre code source canary. Pour plus d'informations sur la façon AWS Secrets Manager de protéger vos secrets, consultez Qu'est-ce que c'est AWS Secrets Manager ? .

Considérations relatives aux autorisations

Nous vous recommandons de restreindre l'accès aux ressources créées ou utilisées par CloudWatch Synthetics. Utilisez des autorisations strictes sur les compartiments Amazon S3 où les scripts Canary stockent les résultats des tests et d'autres artefacts, tels que les journaux et les captures d'écran.

De même, utilisez des autorisations strictes pour les emplacements où votre code source de script Canary est stocké, de sorte qu'aucun utilisateur ne supprime accidentellement ou de façon malveillante les couches Lambda ou les fonctions Lambda utilisées pour le script Canary.

Pour vous assurer que vous exécutez le code de script Canary voulu, vous pouvez utiliser le contrôle de version d'objet sur le compartiment Amazon S3 où votre code de script Canary est stocké. Ensuite, lorsque vous spécifiez ce code à exécuter en tant que script Canary, vous pouvez inclure l'objet `versionId` dans le chemin d'accès, comme dans les exemples suivants.

```
https://bucket.s3.amazonaws.com/path/object.zip?versionId=version-id
https://s3.amazonaws.com/bucket/path/object.zip?versionId=version-id
https://bucket.s3-region.amazonaws.com/path/object.zip?versionId=version-id
```

Traces de pile et messages d'exception

Par défaut, CloudWatch les canaris Synthetics capturent toutes les exceptions générées par votre script Canary, que le script soit personnalisé ou qu'il provienne d'un plan. CloudWatch Synthetics enregistre à la fois le message d'exception et le suivi de la pile à trois emplacements :

- Retournez dans le service CloudWatch Synthetics pour accélérer le débogage lorsque vous décrivez les essais
- Dans CloudWatch les journaux selon la configuration avec laquelle vos fonctions Lambda sont créées
- Dans le fichier journal Synthetics, qui est un fichier en texte brut téléchargé dans l'emplacement Amazon S3 spécifié par la valeur que vous définissez pour le `resultsLocation` du script Canary

Si vous souhaitez envoyer et stocker moins d'informations, vous pouvez capturer les exceptions avant qu'elles ne retournent dans la bibliothèque de CloudWatch wrappers Synthetics.

Vous pouvez également inclure des URL de demande dans vos erreurs. CloudWatch Synthetics analyse toutes les URL contenues dans l'erreur générée par votre script et en supprime les paramètres d'URL restreints en fonction de la configuration. restrictedUrlParameters Si vous journalisez des messages d'erreur dans votre script, vous pouvez utiliser getSanitizedErrorMessage pour effacer les URL avant la journalisation.

Définir une portée limitée pour les rôles IAM

Nous vous recommandons de ne pas configurer votre script Canary de façon à ce qu'il visite des URL ou des points de terminaison potentiellement malveillants. Le fait de pointer votre script Canary vers des sites web ou des points de terminaison non fiables ou inconnus pourrait exposer votre code de fonction Lambda à des scripts d'utilisateurs malveillants. En supposant qu'un site web malveillant puisse sortir de Chromium, il pourrait avoir accès à votre code Lambda comme si vous vous y étiez connecté à l'aide d'un navigateur Internet.

Exécutez votre fonction Lambda avec un rôle d'exécution IAM disposant d'autorisations délimitées. Ainsi, si votre fonction Lambda est compromise par un script malveillant, les actions qu'elle peut effectuer lorsqu'elle est exécutée en tant que compte Canary sont limitées. AWS

Lorsque vous utilisez la CloudWatch console pour créer un Canary, celui-ci est créé avec un rôle d'exécution IAM limité.

Expurgation des données sensibles

CloudWatch Synthetics capture les URL, le code d'état, le motif de l'échec (le cas échéant), ainsi que les en-têtes et le corps des demandes et des réponses. Cela permet à un utilisateur d'un script Canary de comprendre, de surveiller et de déboguer les Canary.

Les configurations décrites dans les sections suivantes peuvent être définies à tout moment de l'exécution des scripts Canary. Vous pouvez également choisir d'appliquer différentes configurations à différentes étapes de synthèse.

URL de demande

Par défaut, les journaux CloudWatch Synthetics demandent des URL, des codes d'état et le motif du statut de chaque URL dans Canary Logs. Les URL de demande peuvent également apparaître dans les rapports d'exécution des scripts Canary, les fichiers HAR, etc. L'URL de votre demande peut contenir des paramètres de requête sensibles, tels que des jetons d'accès ou des mots de passe. Vous pouvez supprimer les informations sensibles afin qu'elles ne soient pas enregistrées par CloudWatch Synthetics.

Pour supprimer des informations sensibles, définissez la propriété restrictedUrlParametersde configuration. Pour plus d'informations, consultez SyntheticsConfiguration classe. Cela oblige CloudWatch Synthetics à supprimer les paramètres d'URL, y compris les valeurs des paramètres de chemin et de requête, sur la base de ce qui précède la journalisation. restrictedUrlParameters Si vous journalisez des URL dans votre script, vous pouvez utiliser getSanitizedUrl(url, StepConfig = nul) pour effacer les URL avant la journalisation. Pour plus d'informations, consultez SyntheticsLogHelper classe.

En-têtes

Par défaut, CloudWatch Synthetics n'enregistre pas les en-têtes de demande/réponse. Pour les scripts Canary de l'interface utilisateur, il s'agit du comportement par défaut pour les scripts Canary utilisant la version d'exécution `syn-nodejs-puppeteer-3.2` et version ultérieure.

Si vos en-têtes ne contiennent pas d'informations sensibles, vous pouvez activer les en-têtes dans les fichiers HAR et les rapports HTTP en définissant les includeResponseHeaderspropriétés includeRequestHeaderset sur. `true` Vous pouvez activer tous les en-têtes, mais choisir de restreindre les valeurs des clés d'en-tête sensibles. Par exemple, vous pouvez choisir de ne masquer que les en-têtes `Authorization` des artefacts produits par des scripts Canary.

Corps de la demande et de la réponse

Par défaut, CloudWatch Synthetics n'enregistre pas le corps de la demande/réponse dans les journaux ou les rapports Canary. Cette information est particulièrement utile pour les scripts Canary d'API. Synthetics capture toutes les requêtes HTTP et peut afficher les en-têtes, les corps de demandes et de réponse. Pour plus d'informations, consultez executeHttpStep(StepName,

RequestOptions, [rappel], [StepConfig]). Vous pouvez choisir d'activer le corps de la demande/réponse en définissant les propriétés includeRequestBodyet includeResponseBodysur. `true`

Journalisation des appels CloudWatch d'API Amazon avec AWS CloudTrail

Amazon CloudWatch et CloudWatch Synthetics sont intégrés AWS CloudTrailà un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un service. AWS CloudTrail capture les appels d'API effectués par ou pour le compte de votre AWS compte. Les appels capturés incluent des appels depuis la console et des appels de code vers des opérations d'API.

Si vous créez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un compartiment S3, y compris les événements pour CloudWatch. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite CloudWatch, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et d'autres détails.

Pour en savoir plus CloudTrail, notamment comment le configurer et l'activer, consultez le guide de AWS CloudTrail l'utilisateur.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l’identité permettent de déterminer les éléments suivants :

- Si la demande a été faite avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d’identification de sécurité temporaires d’un rôle ou d’un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez l'élément CloudTrail UserIdentity.

Pour un enregistrement continu des événements enregistrés sur votre AWS compte, y compris les événements pour Synthetics CloudWatch et ceux de CloudWatch Synthetics, créez un parcours. Un journal permet CloudTrail de fournir des fichiers journaux à un compartiment S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment S3 que vous spécifiez. Vous pouvez configurer d'autres AWS services pour analyser et agir de manière plus approfondie sur les données d'événements collectées dans CloudTrail les journaux. Pour plus d’informations, consultez les ressources suivantes :

- Vue d’ensemble de la création d’un journal d’activité
- CloudTrail Services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux de plusieurs régions et réception de fichiers CloudTrail journaux de plusieurs comptes

> **Note**
>
> Pour plus d'informations sur CloudWatch les appels d'API Logs connectés CloudTrail, consultez la section Informations de CloudWatch journalisation CloudTrail.

Rubriques

- CloudWatch informations dans CloudTrail
- CloudWatch Moniteur Internet en CloudTrail
- CloudWatch Informations sur les synthetics dans CloudTrail

CloudWatch informations dans CloudTrail

CloudWatch prend en charge la journalisation des actions suivantes sous forme d'événements dans les fichiers CloudTrail journaux :

- DeleteAlarms
- DeleteAnomalyDetector
- DeleteDashboards
- DescribeAlarmHistory
- DescribeAlarms
- DescribeAlarmsForMetric
- DescribeAnomalyDetectors
- DisableAlarmActions
- EnableAlarmActions
- GetDashboard
- ListDashboards

- PutAnomalyDetector
- PutDashboard
- PutMetricAlarm
- SetAlarmState

Exemple : entrées de fichier CloudWatch journal

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`PutMetricAlarm`action.

```
{
    "Records": [{
        "eventVersion": "1.01",
        "userIdentity": {
            "type": "Root",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID"
        },
        "eventTime": "2014-03-23T21:50:34Z",
        "eventSource": "monitoring.amazonaws.com",
        "eventName": "PutMetricAlarm",
        "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": {
            "threshold": 50.0,
            "period": 60,
            "metricName": "CloudTrail Test",
            "evaluationPeriods": 3,
            "comparisonOperator": "GreaterThanThreshold",
            "namespace": "AWS/CloudWatch",
            "alarmName": "CloudTrail Test Alarm",
            "statistic": "Sum"
        },
        "responseElements": null,
        "requestID": "29184022-b2d5-11e3-a63d-9b463e6d0ff0",
        "eventID": "b096d5b7-dcf2-4399-998b-5a53eca76a27"
    },
    ..additional entries
  ]
```

```
  }
```

L'entrée de fichier journal suivante indique qu'un utilisateur a appelé l'`PutRule`action CloudWatch Evénements.

```
{
        "eventVersion":"1.03",
        "userIdentity":{
           "type":"Root",
           "principalId":"123456789012",
           "arn":"arn:aws:iam::123456789012:root",
           "accountId":"123456789012",
           "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
           "sessionContext":{
              "attributes":{
                 "mfaAuthenticated":"false",
                 "creationDate":"2015-11-17T23:56:15Z"
              }
           }
        },
        "eventTime":"2015-11-18T00:11:28Z",
        "eventSource":"events.amazonaws.com",
        "eventName":"PutRule",
        "awsRegion":"us-east-1",
        "sourceIPAddress":"AWS Internal",
        "userAgent":"AWS CloudWatch Console",
        "requestParameters":{
           "description":"",
           "name":"cttest2",
           "state":"ENABLED",
           "eventPattern":"{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance
 State-change Notification\"]}",
           "scheduleExpression":""
        },
        "responseElements":{
           "ruleArn":"arn:aws:events:us-east-1:123456789012:rule/cttest2"
        },
        "requestID":"e9caf887-8d88-11e5-a331-3332aa445952",
        "eventID":"49d14f36-6450-44a5-a501-b0fdcdfaeb98",
        "eventType":"AwsApiCall",
        "apiVersion":"2015-10-07",
        "recipientAccountId":"123456789012"
}
```

L'entrée de fichier journal suivante indique qu'un utilisateur a appelé l'`CreateExportTask`action CloudWatch Logs.


```
{
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/someuser",
            "accountId": "123456789012",
            "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
            "userName": "someuser"
        },
        "eventTime": "2016-02-08T06:35:14Z",
        "eventSource": "logs.amazonaws.com",
        "eventName": "CreateExportTask",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "127.0.0.1",
        "userAgent": "aws-sdk-ruby2/2.0.0.rc4 ruby/1.9.3 x86_64-linux Seahorse/0.1.0",
        "requestParameters": {
            "destination": "yourdestination",
            "logGroupName": "yourloggroup",
            "to": 123456789012,
            "from": 0,
            "taskName": "yourtask"
        },
        "responseElements": {
            "taskId": "15e5e534-9548-44ab-a221-64d9d2b27b9b"
        },
        "requestID": "1cd74c1c-ce2e-12e6-99a9-8dbb26bd06c9",
        "eventID": "fd072859-bd7c-4865-9e76-8e364e89307c",
        "eventType": "AwsApiCall",
        "apiVersion": "20140328",
        "recipientAccountId": "123456789012"
}
```


CloudWatch Moniteur Internet en CloudTrail

CloudWatch Internet Monitor prend en charge l'enregistrement des actions suivantes sous forme d'événements dans des fichiers CloudTrail journaux.

- CreateMonitor

- DeleteMonitor
- GetHealthEvent
- GetMonitor
- GetQueryResults
- GetQueryStatus
- ListHealthEvents
- ListMonitors
- ListTagsForResource
- StartQuery
- StopQuery
- UpdateMonitor

Exemple : entrées du fichier journal d' CloudWatch Internet Monitor

L'exemple suivant montre une entrée du journal CloudTrail Internet Monitor qui illustre l'`ListMonitors`action.


```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
```

```
            }
        },
        "eventTime": "2022-10-11T17:30:18Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "ListMonitors",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": null,
        "responseElements": null,
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": true,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```

L'exemple suivant montre une entrée du journal CloudTrail Internet Monitor qui illustre l'`CreateMonitor`action.

```
{
        "eventVersion": "1.08",
        "userIdentity": {
            "type": "AssumedRole",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::000000000000:assumed-role/role_name",
            "accountId":"123456789012",
            "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
            "sessionContext": {
                "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::000000000000:role/Admin",
                "accountId":"123456789012",
                "userName": "SAMPLE_NAME"
                },
                "webIdFederationData": {},
                "attributes": {
                    "creationDate": "2022-10-11T17:25:41Z",
                    "mfaAuthenticated": "false"
                }
```

```
            }
        },
        "eventTime": "2022-10-11T17:30:08Z",
        "eventSource": "internetmonitor.amazonaws.com",
        "eventName": "CreateMonitor",
        "awsRegion": "us-east-2",
        "sourceIPAddress": "192.0.2.0",
        "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)",
        "requestParameters": {
            "MonitorName": "TestMonitor",
            "Resources": ["arn:aws:ec2:us-east-2:444455556666:vpc/vpc-febc0b95"],
            "ClientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
        },
        "responseElements": {
            "Arn": "arn:aws:internetmonitor:us-east-2:444455556666:monitor/ct-
onboarding-test",
            "Status": "PENDING"
        },
        "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "eventCategory": "Management"
    }
```


CloudWatch Informations sur les synthetics dans CloudTrail

CloudWatch Synthetics prend en charge l'enregistrement des actions suivantes sous forme d'événements CloudTrail dans des fichiers journaux :

- CreateCanary
- DeleteCanary
- DescribeCanaries
- DescribeCanariesLastRun
- DescribeRuntimeVersions
- GetCanary
- GetCanaryRuns

- ListTagsForResource
- StartCanary
- StopCanary
- TagResource
- UntagResource
- UpdateCanary

Exemple : entrées du CloudWatch fichier journal Synthetics

L'exemple suivant montre une CloudTrail entrée du journal Synthetics qui illustre l'action. `DescribeCanaries`


```
{
    "eventVersion": "1.05",
    "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"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "DescribeCanaries",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
```

```
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "201ed5f3-15db-4f87-94a4-123456789",
    "eventID": "73ddbd81-3dd0-4ada-b246-123456789",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

L'exemple suivant montre une CloudTrail entrée du journal Synthetics qui illustre l'action. `UpdateCanary`

```
{
    "eventVersion": "1.05",
    "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"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:47Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "UpdateCanary",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Schedule": {
            "Expression": "rate(1 minute)"
        },
        "name": "sample_canary_name",
        "Code": {
            "Handler": "myOwnScript.handler",
            "ZipFile": "SAMPLE_ZIP_FILE"
        }
    },
    "responseElements": null,
    "requestID": "fe4759b0-0849-4e0e-be71-1234567890",
    "eventID": "9dc60c83-c3c8-4fa5-bd02-1234567890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

L'exemple suivant montre une CloudTrail entrée du journal Synthetics qui illustre l'action. `GetCanaryRuns`

```
{
    "eventVersion": "1.05",
    "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"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
```

```
                "creationDate": "2020-04-08T21:43:24Z"
            }
        }
    },
    "eventTime": "2020-04-08T23:06:30Z",
    "eventSource": "synthetics.amazonaws.com",
    "eventName": "GetCanaryRuns",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "aws-internal/3 aws-sdk-java/1.11.590
 Linux/4.9.184-0.1.ac.235.83.329.metal1.x86_64 OpenJDK_64-Bit_Server_VM/25.212-b03
 java/1.8.0_212 vendor/Oracle_Corporation",
    "requestParameters": {
        "Filter": "TIME_RANGE",
        "name": "sample_canary_name",
        "FilterValues": [
            "2020-04-08T23:00:00.000Z",
            "2020-04-08T23:10:00.000Z"
        ]
    },
    "responseElements": null,
    "requestID": "2f56318c-cfbd-4b60-9d93-1234567890",
    "eventID": "52723fd9-4a54-478c-ac55-1234567890",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Marquer vos ressources Amazon CloudWatch

Une balise est une étiquette d'attribut personnalisée que vous attribuez ou AWS assignez à une AWS ressource. Chaque balise se compose de deux parties :

- Une clé de balise (par exemple, `CostCenter`, `Environment` ou `Project`). Les clés de balises sont sensibles à la casse.
- Un champ facultatif appelé valeur de balise (par exemple, `111122223333` ou `Production`). Si la valeur de balise est identique à l'utilisation d'une chaîne vide. Les valeurs de balise sont sensibles à la casse, tout comme les clés de balise.

Les balises vous permettent d'effectuer les actions suivantes :

- Identifiez et organisez vos AWS ressources. De nombreux AWS services prennent en charge le balisage. Vous pouvez donc attribuer le même tag aux ressources de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à une CloudWatch règle que celle que vous attribuez à une instance EC2.

Les sections suivantes fournissent plus d'informations sur les balises pour CloudWatch.

Ressources prises en charge dans CloudWatch

Les ressources suivantes relatives au balisage CloudWatch sont les suivantes :

- Alarmes — Vous pouvez étiqueter les alarmes à l'aide de la AWS CLI commande tag-resource et de l'TagResourceAPI. Vous pouvez également consulter et gérer vos balises d'alarme à l'aide de la page de détails des alarmes de la CloudWatch console.
- Canaris — Vous pouvez étiqueter les canaris à l'aide de la CloudWatch console. Pour plus d'informations, consultez Création d'un Canary.
- Règles de Contributor Insights : vous pouvez baliser les règles de Contributor Insights lorsque vous les créez à l'aide de la put-insight-rule AWS CLI commande et de l'PutInsightRuleAPI. Vous pouvez ajouter des balises aux règles existantes à l'aide de la AWS CLI commande tag-resource et de l'TagResourceAPI.
- Flux métriques : vous pouvez baliser les flux métriques lorsque vous les créez à l'aide de la put-metric-stream AWS CLI commande et de l'PutMetricStreamAPI. Vous pouvez ajouter des

balises aux flux métriques existants à l'aide de la AWS CLI commande tag-resource et de l'TagResourceAPI.

Pour obtenir des informations sur l'ajout et la gestion de balises, veuillez consulter Gestion des balises.

Gestion des balises

Les balises comprennent les propriétés `Value` et `Key` d'une ressource. Vous pouvez utiliser la CloudWatch console, le AWS CLI, ou l' CloudWatch API pour ajouter, modifier ou supprimer les valeurs de ces propriétés. Pour obtenir plus d'informations sur l'utilisation des balises, consultez ce qui suit :

- TagResourceUntagResource, et ListTagsForResourcedans le Amazon CloudWatch API Reference
- tag-resource, untag-resource et dans le list-tags-for-resourceAmazon CLI Reference CloudWatch
- Utilisation de l'éditeur de balises dans le Guide de l'utilisateur Resource Groups

Conventions de dénomination et d'utilisation des balises

Les conventions de dénomination et d'utilisation de base suivantes s'appliquent à l'utilisation de balises avec CloudWatch des ressources :

- Chaque ressource peut avoir un maximum de 50 balises.
- Pour chaque ressource, chaque clé de balise doit être unique, et chaque clé de balise peut avoir une seule valeur.
- La longueur maximale des clés de balise est de 128 caractères Unicode en UTF-8.
- La longueur maximale des valeurs de balise est de 256 caractères Unicode en UTF-8.
- Les caractères autorisés sont les lettres, les espaces et les chiffres représentables en UTF-8, ainsi que les caractères spéciaux suivants : . : + = @ _ / - (tiret).
- Les clés et valeurs de balise sont sensibles à la casse. La bonne pratique consiste à choisir une stratégie pour mettre des balises en majuscule et mettre en œuvre cette stratégie de manière cohérente sur tous les types de ressources. Par exemple, décidez si vous souhaitez utiliser `Costcenter`, `costcenter` ou `CostCenter`, et utilisez la même convention pour toutes les balises. Évitez d'utiliser des balises avec une incohérence de traitement de cas similaires.

- Le `aws:` préfixe est interdit pour les balises car il est réservé à l' AWS usage. Vous ne pouvez pas modifier ni supprimer des clés ou valeurs d’étiquette ayant ce préfixe. Les balises avec ce préfixe ne sont pas comptabilisées comme vos balises pour la limite de ressources.

Intégration de Grafana

Vous pouvez utiliser Grafana version 6.5.0 et versions ultérieures pour avancer de manière contextuelle dans la CloudWatch console et interroger une liste dynamique de métriques à l'aide de caractères génériques. Cela peut vous aider à surveiller les métriques pour les ressources AWS , telles que les instances ou les conteneurs Amazon Elastic Compute Cloud. Lorsque de nouvelles instances sont créées dans le cadre d'un événement Auto Scaling, elles apparaissent automatiquement dans le graphique. Vous n'avez pas besoin de suivre les nouveaux ID d'instance. Les tableaux de bord prédéfinis permettent de simplifier l'expérience de démarrage en matière de surveillance d'Amazon EC2, d'Amazon Elastic Block Store AWS Lambda et des ressources.

Vous pouvez utiliser Grafana version 7.0 et versions ultérieures pour effectuer des requêtes CloudWatch Logs Insights sur des groupes de journaux dans CloudWatch Logs. Vous pouvez visualiser les résultats de vos requêtes sous forme de graphiques à barres, de graphiques linéaires et d'histogrammes empilés, ainsi que sous forme de tableaux. Pour plus d'informations sur CloudWatch Logs Insights, consultez Analyser les données des CloudWatch journaux avec Logs Insights.

Pour plus d'informations sur la façon de démarrer, consultez la section Utilisation AWS CloudWatch dans Grafana dans la documentation de Grafana Labs.

Console multicompte et multirégion CloudWatch

Pour tirer le meilleur parti de l'observabilité et de la découverte entre comptes pour vos métriques, journaux et traces, nous vous recommandons d'utiliser l'observabilité CloudWatch entre comptes. Pour plus d'informations, consultez CloudWatch observabilité entre comptes.

CloudWatch propose également un tableau de bord multicompte et interrégional. CloudWatch Cette fonctionnalité vous offre une visibilité entre comptes de vos tableaux de bord, alarmes, métriques et tableaux de bord automatiques. Elle ne fournit pas de visibilité inter-comptes pour les journaux ou pour les traces.

Si vous utilisez également l'observabilité CloudWatch entre comptes, ce tableau de CloudWatch bord multicompte peut notamment permettre à l'un de vos comptes sources d'observabilité CloudWatch entre comptes de consulter les statistiques d'un autre compte source.

Le reste de cette section décrit le tableau de bord inter-comptes et inter-régions. Vous pouvez l'utiliser pour créer des tableaux de bord qui résument les CloudWatch données de plusieurs AWS comptes et de plusieurs AWS régions en un seul tableau de bord. Vous pouvez également créer une alarme dans un compte qui surveille une métrique située dans un autre compte.

De nombreuses entreprises déploient leurs AWS ressources sur plusieurs comptes, afin de définir des limites de facturation et de sécurité. Dans ce cas, nous vous recommandons de désigner un ou plusieurs de vos comptes comme comptes de surveillance et de créer vos tableaux de bord entre comptes dans ces comptes.

La fonctionnalité multi-comptes est intégrée pour vous aider à créer efficacement vos tableaux de bord inter-comptes. AWS Organizations

Fonctionnalité entre régions

La fonctionnalité entre régions est désormais intégrée automatiquement. Vous n'avez pas besoin d'effectuer d'étapes supplémentaires pour pouvoir afficher les métriques de différentes régions dans un seul compte sur le même graphique ou le même tableau de bord. La fonctionnalité inter-régions n'est pas prise en charge pour les alarmes. Vous ne pouvez donc pas créer d'alarme dans une région qui surveille une métrique dans une région différente.

Rubriques

- Résolution des problèmes liés à la CloudWatch configuration de plusieurs comptes
- Désactivation et nettoyage après l'utilisation de la fonctionnalité entre comptes

Activation de la fonctionnalité multi-comptes dans CloudWatch

Pour configurer la fonctionnalité multi-comptes dans votre CloudWatch console, utilisez la CloudWatch console pour configurer vos comptes de partage et vos comptes de surveillance.

Configuration d'un compte de partage

Vous devez activer le partage dans chaque compte qui mettra les données à la disposition du compte de surveillance.

Ainsi, vous accordez les autorisations en lecture seule que vous choisissez à l'étape 5 à tous les utilisateurs capables d'afficher un tableau de bord entre comptes dans le compte que vous partagez avec lui, si l'utilisateur dispose des autorisations correspondantes dans le compte que vous partagez avec lui.

Pour permettre à votre compte de partager CloudWatch des données avec d'autres comptes

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres).
3. Pour Partager vos CloudWatch données, choisissez Configurer.
4. Dans Sharing (Partage), choisissez Specific accounts (Comptes spécifiques) et saisissez les ID des comptes avec lesquels vous souhaitez partager des données.

   Tous les comptes que vous spécifiez ici peuvent consulter les CloudWatch données de votre compte. Spécifiez uniquement les ID des comptes que vous connaissez et dans lesquels vous avez confiance.
5. Dans Permissions (Autorisations), spécifiez comment partager vos données avec l'une des options suivantes :

   - Fournissez un accès en lecture seule à vos CloudWatch indicateurs, tableaux de bord et alarmes. Cette option permet aux comptes de surveillance de créer des tableaux de bord inter-comptes qui incluent des widgets contenant les CloudWatch données de votre compte.
   - Incluez des tableaux de bord CloudWatch automatiques. Si vous sélectionnez cette option, les utilisateurs du compte de surveillance peuvent également voir les informations dans les

tableaux de bord automatiques de ce compte. Pour plus d'informations, consultez Commencer à utiliser Amazon CloudWatch.

- Intégrez un accès X-Ray en lecture seule pour la carte de suivi X-Ray. Si vous sélectionnez cette option, les utilisateurs du compte de surveillance peuvent également consulter la carte de suivi X-Ray et les informations de suivi X-Ray dans ce compte. Pour plus d'informations, veuillez consulter la rubrique Using the X-Ray Trace Map.
- Full read-only access to everything in your account (Accès complet en lecture seule à tout ce qui se trouve dans votre compte). Cette option permet aux comptes que vous utilisez pour le partage de créer des tableaux de bord inter-comptes qui incluent des widgets contenant les CloudWatch données de votre compte. Elle permet également à ces comptes d'examiner plus en profondeur votre compte et de voir les données de votre compte dans les consoles d'autres services AWS .

6. Choisissez le CloudFormation modèle de lancement.

   Dans l'écran de confirmation, saisissez **`Confirm`** et choisissez Launch template (Lancer le modèle).

7. Cochez la case Je sais... , puis choisissez Créer une pile.

Partage avec une organisation entière

L'exécution de la procédure précédente crée un rôle IAM qui permet à votre compte de partager des données avec un seul compte. Vous pouvez créer ou modifier un rôle IAM qui partage vos données avec tous les comptes d'une organisation. Faites-le uniquement si vous connaissez et faites confiance à tous les comptes de l'organisation.

Ainsi, vous accordez les autorisations en lecture seule répertoriées dans les stratégies reprises à l'étape 5 de la procédure précédente à tous les utilisateurs capables d'afficher un tableau de bord entre comptes dans le compte que vous partagez avec lui, si l'utilisateur dispose des autorisations correspondantes dans le compte que vous partagez avec lui.

Pour partager les données de votre CloudWatch compte avec tous les comptes d'une organisation

1. Si ce n'est pas déjà fait, suivez la procédure précédente pour partager vos données avec un seul AWS compte.
2. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
3. Dans le panneau de navigation, choisissez Roles (Rôles).

4. Dans la liste des rôles, choisissez CloudWatch- CrossAccountSharingRole.
5. Choisissez Relations d'approbation, Modifier la relation d'approbation.

   Vous voyez une stratégie comme la suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. Remplacez la stratégie par la suivante, en remplaçant *org-id* par l'ID de votre organisation.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:PrincipalOrgID": "org-id"
        }
      }

    }
  ]
}
```

7. Choisissez Mettre à jour la politique d'approbation.

Configuration d'un compte de surveillance

Activez chaque compte de surveillance si vous souhaitez consulter les CloudWatch données entre comptes.

Lorsque vous avez terminé la procédure suivante, CloudWatch crée un rôle lié à un service qui est CloudWatch utilisé dans le compte de surveillance pour accéder aux données partagées depuis vos autres comptes. Ce rôle lié au service est appelé. AWSServiceRoleForCloudWatchCrossAccount Pour de plus amples informations, veuillez consulter Utilisation des rôles liés aux services pour CloudWatch.

Pour permettre à votre compte de consulter les données entre comptes CloudWatch

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Settings (Paramètres). Puis, dans la section Cross-account cross-region (Entre régions et comptes), sélectionnez Configure (Configurer).
3. Dans la section Afficher plusieurs comptes entre régions, choisissez Activer, puis cochez la case Afficher le sélecteur dans la console pour permettre à un sélecteur de compte d'apparaître dans la CloudWatch console lorsque vous tracez un graphique sur une métrique ou que vous créez une alarme.
4. Sous View cross-account cross-region (Afficher entre régions et comptes), choisissez l'une des options suivantes :

   - Account Id Input (Saisie d'ID de compte). Cette option vous invite à saisir manuellement un ID de compte chaque fois que vous souhaitez changer de compte lorsque vous affichez des données entre comptes.
   - AWS Sélecteur de compte d'organisation. Cette option entraîne l'affichage des comptes que vous avez spécifiés après avoir terminé l'intégration entre comptes avec Organizations. La prochaine fois que vous utiliserez la console, elle CloudWatch affichera une liste déroulante de ces comptes parmi laquelle vous pourrez faire votre choix lorsque vous consulterez les données entre comptes.

     Pour ce faire, vous devez d'abord avoir utilisé le compte de gestion de votre organisation CloudWatch pour permettre de voir la liste des comptes de votre organisation. Pour plus d’informations, consultez (Facultatif) Intégrer avec AWS Organizations.
   - Custom account selector (Sélecteur de compte personnalisé). Cette option vous invite à saisir une liste des ID de compte. La prochaine fois que vous utiliserez la console, elle CloudWatch affichera une liste déroulante de ces comptes parmi laquelle vous pourrez faire votre choix lorsque vous consulterez les données entre comptes.

Vous pouvez également saisir une étiquette pour chacun de ces comptes, qui vous permettra de les identifier lorsqu'il faudra choisir les comptes à afficher.

Les paramètres de sélecteur de compte qu'un utilisateur définit ici ne sont conservés que pour cet utilisateur, et non pour tous les autres utilisateurs du compte de surveillance.

5. Sélectionnez Activer.

Une fois cette configuration terminée, vous pouvez créer des tableaux de bord entre comptes. Pour plus d'informations, consultez Tableaux de bord entre régions et comptes.

(Facultatif) Intégrer avec AWS Organizations

Si vous souhaitez intégrer la fonctionnalité multi-comptes AWS Organizations, vous devez créer une liste de tous les comptes de l'organisation accessibles aux comptes de surveillance.

Pour activer la CloudWatch fonctionnalité multi-comptes afin d'accéder à la liste de tous les comptes de votre organisation

1. Connectez-vous au compte de gestion de votre organisation.
2. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le volet de navigation, choisissez Settings (Paramètres), puis Configurer (Configurer).
4. Dans Grant permission to view the list of accounts in the organization (Accorder l'autorisation d'afficher la liste des comptes de l'organisation), choisissez Specific accounts (Comptes spécifiques) pour être invité à saisir une liste des ID de compte. La liste des comptes de votre organisation est uniquement partagée avec les comptes que vous spécifiez ici.
5. Choisissez Share organization account list (Partager la liste des comptes de l'organisation).
6. Choisissez le CloudFormation modèle de lancement.

   Dans l'écran de confirmation, saisissez **`Confirm`** et choisissez Launch template (Lancer le modèle).

Résolution des problèmes liés à la CloudWatch configuration de plusieurs comptes

Cette section contient des conseils de dépannage pour le déploiement de consoles entre comptes dans CloudWatch.

J'obtiens des erreurs d'accès refusés affichant des données entre comptes

Vérifiez les éléments suivants :

- Votre compte de surveillance doit avoir un rôle nommé AWSServiceRoleForCloudWatchCrossAccount. Si ce n'est pas le cas, vous devez créer ce rôle. Pour plus d'informations, consultez Set Up a Monitoring Account.
- Chaque compte de partage doit avoir un rôle nommé CloudWatch- CrossAccountSharingRole. Si ce n'est pas le cas, vous devez créer ce rôle. Pour de plus amples informations, consultez Set Up A Sharing Account.
- Le rôle de partage doit faire confiance au compte de surveillance.

Pour vérifier que vos rôles sont correctement configurés pour la console CloudWatch multi-comptes

1. Connectez-vous à la console IAM AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Dans la liste des rôles, assurez-vous que le rôle nécessaire existe. Dans un compte de partage, recherchez CloudWatch- CrossAccountSharingRole. Dans un compte de surveillance, recherchez AWSServiceRoleForCloudWatchCrossAccount.
4. Si vous utilisez un compte de partage et CloudWatch qu'il existe CrossAccountSharingRole déjà, choisissez CloudWatch- CrossAccountSharingRole.
5. Choisissez Relations d'approbation, Modifier la relation d'approbation.
6. Vérifiez que la stratégie répertorie soit l'ID de compte du compte de surveillance, soit l'ID d'organisation d'une organisation qui contient le compte de surveillance.

Je ne vois aucune liste déroulante de compte dans la console

Commencez par vérifier que vous avez créé les rôles IAM appropriés, selon les instructions de la section de résolution des problèmes précédente. Si ceux-ci sont correctement configurés,

assurez-vous que vous avez activé ce compte pour afficher les données entre comptes, selon la description de la section Enable Your Account to View Cross-Account Data.

Désactivation et nettoyage après l'utilisation de la fonctionnalité entre comptes

Pour désactiver la fonctionnalité multi-comptes pour CloudWatch, procédez comme suit.

Étape 1 : supprimer les piles ou les rôles entre comptes

La meilleure méthode consiste à supprimer les AWS CloudFormation piles utilisées pour activer la fonctionnalité multi-comptes.

- Dans chacun des comptes de partage, supprimez la CrossAccountSharingRole pile CloudWatch-.
- Si vous aviez AWS Organizations l'habitude d'activer la fonctionnalité multi-comptes avec tous les comptes d'une organisation, supprimez la CrossAccountListAccountsRole pile CloudWatch- dans le compte de gestion de l'organisation.

Si vous n'avez pas utilisé les AWS CloudFormation piles pour activer la fonctionnalité multi-comptes, procédez comme suit :

- Dans chacun des comptes de partage, supprimez le rôle CloudWatch- CrossAccountSharingRole IAM.
- Si vous aviez AWS Organizations l'habitude d'activer la fonctionnalité multi-comptes avec tous les comptes d'une organisation, supprimez le rôle CloudWatch- CrossAccountSharing - ListAccountsRole IAM dans le compte de gestion de l'organisation.

Étape 2 : supprimer le rôle lié à un service

Dans le compte de surveillance, supprimez le rôle AWSServiceRoleForCloudWatchCrossAccountIAM lié au service.

CloudWatch quotas de service

CloudWatch possède les quotas suivants pour les métriques, les alarmes, les demandes d'API et les notifications par e-mail d'alarme.

> **Note**
>
> Pour certains AWS services CloudWatch, notamment, vous pouvez utiliser les indicateurs CloudWatch d'utilisation pour visualiser votre utilisation actuelle des services sur CloudWatch des graphiques et des tableaux de bord. Vous pouvez utiliser une fonction mathématique CloudWatch métrique pour afficher les quotas de service pour ces ressources sur vos graphiques. Vous pouvez également configurer des alarmes qui vous alertent lorsque votre utilisation approche d'un quota de service. Pour plus d'informations, consultez Visualisation de vos quotas de service et définition d'alertes.

Ressource	Quota par défaut
Actions d'alarme	5/alarme. Ce quota ne peut pas être modifié.
Période d'évaluation de l'alarme	La valeur maximale, calculée en multipliant la période d'alarme par le nombre de périodes d'évaluation utilisées , est d'un jour (86 400 secondes). Ce quota ne peut pas être modifié.
Alertes	10/mois/client gratuites. Des alarmes supplémentaires entraînent des frais. Aucune limite sur le nombre total d'alarmes par compte. Les alarmes basées sur des expressions mathématiques appliquées aux métriques peuvent comporter jusqu'à 10 métriques. 200 alarmes Metrics Insights par région. Vous pouvez demander une augmentation de quota.
Modèles de détection d'anomalies	500 par région et par compte.

Ressource	Quota par défaut
Demandes d'API	1 000 000/mois/client gratuites.
Scripts Canary	200 par région et par compte. Vous pouvez demander une augmentation de quota.
Demandes d'API de Contributor Insights	Les API suivantes ont un quota de 20 transactions par seconde (TPS) et par région. • DescribeInsightRules Ce quota ne peut pas être modifié. • GetInsightRuleReport Vous pouvez demander une augmentation de quota. Les API suivantes ont un quota de 5 TPS par région. Ce quota ne peut pas être modifié. • DeleteInsightRules • PutInsightRule Les API suivantes ont un quota de 1 TPS par région. Ce quota ne peut pas être modifié. • DisableInsightRules • EnableInsightRules
Règles de Contributor Insights	100 règles par région et par compte. Vous pouvez demander une augmentation de quota.
Métriques personnalisées	Pas de quota.

Ressource	Quota par défaut
Tableaux de bord	Jusqu'à 500 widgets par tableau de bord. Jusqu'à 500 métriques par widget de tableau de bord. Jusqu'à 2 500 métriques par tableau de bord, pour tous les widgets. Ces quotas incluent toutes les métriques récupérées pour être utilisées dans les fonctions mathématiques appliquées aux métriques, même si ces métriques ne sont pas affichées sur le graphique. Ces quotas ne peuvent pas être modifiés.
DescribeAlarms	9 transactions par seconde (TPS) par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Vous pouvez demander une augmentation de quota.
DeleteAlarmsdemande DescribeAlarmHistorydemande DisableAlarmActionsdemande EnableAlarmActionsdemande SetAlarmStatedemande	3 TPS par région pour chacune de ces opérations. Le nombre maximum de demandes d'opérations par seconde sans être limité. Ces quotas ne peuvent pas être modifiés.
DescribeAlarmsForMetricdemande	9 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Ces quotas ne peuvent pas être modifiés.

Ressource	Quota par défaut
DeleteDashboardsdemande GetDashboarddemande ListDashboardsdemande PutDashboarddemande	10 TPS par région pour chacune de ces opérations. Le nombre maximum de demandes d'opérations par seconde sans être limité. Ces quotas ne peuvent pas être modifiés.
PutAnomalyDetector DescribeAnomalyDetectors	10 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité.
DeleteAnomalyDetector	5 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité.
Dimensions	30/métrique. Ce quota ne peut pas être modifié.

Ressource	Quota par défaut
GetMetricData	10 TPS par région pour les opérations qui comprennent des requêtes Metrics Insights. Pour les opérations qui ne comprennent pas de requêtes Metrics Insights, le quota est de 50 TPS par région. Il s'agit du nombre maximal de demandes d'opérations que vous pouvez effectuer par seconde sans être limité. Vous pouvez demander une augmentation de quota. Pour`GetMetricData` les opérations qui comprennent une requête Metrics Insights, le quota est de 4 300 000 points de données par seconde (DPS) pendant les 3 dernières heures. Ce quota est calculé par rapport au nombre total de points de données analysés par la requête (qui ne peut comprendre plus de 10 000 métriques). 180 000 points de données par seconde (DPS) si la valeur `StartTime` utilisée dans la demande d'API est inférieure ou égale à trois heures par rapport à l'heure actuelle. 396 000 DPS si la valeur `StartTime` est supérieure à trois heures par rapport à l'heure actuelle. Il s'agit du nombre maximal de points de données que vous pouvez demander par seconde à l'aide d'un ou de plusieurs appels d'API sans être limité. Ce quota ne peut pas être modifié. Le nombre de points de données par seconde est calculé en fonction des points de données estimés et non des points de données réels. L'estimation des points de données est calculée à l'aide de la plage de temps, de la période et de la période de conservation demandées. Cela signifie que si les points de données réels dans les métriques demandées sont rares ou vides, la limitation se produit toujours si les points de données estimés dépassent le quota. Le quota DPS est par région.

Ressource	Quota par défaut
GetMetricData	Un seul appel `GetMetricData` peut inclure les éléments suivants : • Jusqu'à 500 structures `MetricDataQuery` . • Jusqu'à 100 fonctions `SERVICE_QUOTA()` . • Jusqu'à 100 fonctions `SEARCH()`. • Jusqu'à 5 fonctions `LAMBDA()`. Ces quotas ne peuvent pas être modifiés.
GetMetricStatistics	400 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Vous pouvez demander une augmentation de quota.
GetMetricWidgetImage	Jusqu'à 500 métriques par image. Ce quota ne peut pas être modifié. 20 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Vous pouvez demander une augmentation de quota.
ListMetrics	25 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Vous pouvez demander une augmentation de quota.
Valeurs de données métriques	La valeur d'un point de données métriques doit être comprise entre -2^360 et 2^360. Les valeurs spéciales (par exemple, NaN, +Infini, -Infini) ne sont pas prises en charge. Ce quota ne peut pas être modifié.

Ressource	Quota par défaut
MetricDatumarticles	1000/ PutMetricDatademande. Un MetricDatumobjet peut contenir une seule valeur ou un StatisticSetobjet représentant plusieurs valeurs. Ce quota ne peut pas être modifié.
Métriques	10/mois/client gratuites.
Requêtes Métriques Insights	Une seule requête ne peut traiter plus de 10 000 métriques. Cela signifie que si SÉLECTIONNER,À PARTIR DE, etOÙles clauses correspondent à plus de 10 000 métriques, seules les 10 000 premières métriques trouvées seront traitées par la requête. Une seule requête ne peut pas renvoyer plus de 500 séries chronologiques. Vous ne pouvez interroger que les trois heures de données les plus récentes
Taux de demandes d'API Observabi lity Access Manager (OAM).	1 TPS par région pour PutSinkPolicy. 10 TPS par région pour chaque autre API CloudWatch OAM. Ces quotas reflètent le nombre maximum de requêtes d'opérations par seconde sans être limité. Ces quotas ne peuvent pas être modifiés.
Liens vers les comptes sources OAM	Chaque compte source peut être lié à 5 comptes de surveillance au maximum Ce quota ne peut pas être modifié.
Récepteurs OAM	1 évier par région et par compte Ce quota ne peut pas être modifié.

Ressource	Quota par défaut
PutCompositeAlarmdemande	3 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Vous pouvez demander une augmentation de quota.
PutMetricAlarmdemande	3 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Vous pouvez demander une augmentation de quota.
PutMetricDatademande	1 Mo pour les requêtes HTTP POST. PutMetricDatapeut gérer 500 transactions par seconde (TPS), soit le nombre maximum de demandes d'opérations que vous pouvez effectuer par seconde sans être limité. `PutMetricData` peut gérer 1 000 métriques par demande. Vous pouvez demander une augmentation de quota.
Envoi de notifications Amazon SNS par e-mail	1 000/mois/client gratuites.
Groupes Synthetics	20 par compte. Ce quota ne peut pas être modifié.
TagResource	20 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Ce quota ne peut pas être modifié.
UntagResource	20 TPS par région. Le nombre maximum de demandes d'opérations par seconde sans être limité. Ce quota ne peut pas être modifié.

Historique du document

Le tableau suivant décrit les modifications importantes apportées à chaque version du guide de l' CloudWatch utilisateur Amazon, à compter du mois de juin 2018. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date
CloudWatch La carte de service Application Signals prend en charge les clients Canary, RUM et les groupements AWS de dépendances de service.	La version préliminaire d'Application Signals a ajouté des groupements par défaut dans la carte des services pour les canaris, les clients RUM et les dépendances de AWS service du même type. Cette modification réduit le nombre d'icônes dans la vue par défaut de la carte de service afin de faciliter la visualisation et la navigation.	21 mai 2024
CloudWatchReadOnlyAccess Politique IAM mise à jour	CloudWatch a modifié le champ d'application d'une autorisation dans CloudWatchReadOnlyAccess. La portée de la politique a ajouté les `application-signals:List*` actions `application-signals:BatchGet*` `application-signals:Get*` , et afin que les utilisateurs puissent utiliser les signaux CloudWatch d'application pour visualiser, étudier et diagnostiquer les problèmes liés à l'état de santé	17 mai 2024

	de leurs services. CloudWatch a également ajouté une `iam:GetRole` action afin que les utilisateurs puissent vérifier si Application Signals est configuré.	
CloudWatchFullAccessPolitique IAM V2 mise à jour	CloudWatch a modifié la portée d'une autorisation dans la CloudWatchFullAccessversion 2. La portée de la politique a été ajoutée `application-signals:*` afin que les utilisateurs puissent utiliser les signaux CloudWatch d'application pour visualiser, étudier et diagnostiquer les problèmes liés à l'état de santé de leurs services.	17 mai 2024
Lambda Insights prend en charge AWS GovCloud (USA Est) et AWS GovCloud (USA Ouest)	CloudWatch Lambda Insights a ajouté le support pour les régions AWS GovCloud (USA Est) et AWS GovCloud (USA Ouest).	29 avril 2024
CloudWatch l'observabilité entre comptes prend en charge les filtres de ressources	Vous pouvez désormais créer des filtres pour spécifier les espaces de noms métriques et les groupes de journaux qui sont partagés entre le compte source et le compte de surveillance, lorsque vous créez le lien entre les comptes.	26 avril 2024

CloudWatch Mise à jour des signaux d'application	La version préliminaire d'Application Signals a ajouté trois fonctionnalités. Application Signals prend désormais en charge les applications Python. Il propose un processus d'activation plus simple pour les applications sur les architectures Amazon EKS. Il inclut également de nouvelles configurations que vous pouvez utiliser pour gérer la cardinalité des métriques collectées.	26 avril 2024
CloudWatch Container Insights, avec une observabilité améliorée pour Amazon EKS, permet de collecter les métriques d' AWS Elastic Fabric Adapter (EFA)	Vous pouvez désormais utiliser CloudWatch Container Insights avec une observabilité améliorée pour qu'Amazon EKS collecte les métriques AWS Elastic Fabric Adapter (EFA) à partir de clusters Amazon EKS.	23 avril 2024

Politique IAM mise à jour	CloudWatch a mis à jour la CloudWatchApplicationSignalsServiceRolePolicypolitique. Le champ d'application `logs:StartQuery` et les `logs:GetQueryResults` autorisations de cette politique ont été modifiés pour ajouter `arn:aws:logs:*:*:log-group:/aws/appsignals/**` et activer les signaux `"arn:aws:logs:*:*:log-group:/aws/application-signals/data:*"` d'application sur un plus grand nombre d'architectures. Cette politique est associée au rôle AWSServiceRoleForCloudWatchApplicationSignalslié au service.	18 avril 2024
Internet Monitor fournit une carte météo mondiale sur Internet aux clients authentifiés AWS	Amazon CloudWatch Internet Monitor affiche désormais une carte météo Internet mondiale qui est disponible dans la console pour tous les AWS clients authentifiés. Pour afficher la carte, dans la CloudWatch console Amazon, accédez à Internet Monitor.	16 avril 2024

CloudWatch Container Insights avec une observabilité améliorée pour Amazon EKS peut collecter des métriques AWS Neuron	Vous pouvez désormais utiliser CloudWatch Container Insights avec une observabilité améliorée pour qu'Amazon EKS collecte des métriques AWS Neuron à partir de clusters Amazon EKS.	16 avril 2024
CloudWatch Application Signals ajoute un onglet de présentation des services et d'autres indicateurs pour faciliter les diagnostics	Un nouvel onglet Aperçu du service affiche une vue d'ensemble de votre service, y compris le nombre d'opérations, de dépendances, de synthèses et de pages client. L'onglet affiche les indicateurs clés pour l'ensemble de votre service, ainsi que les principales opérations et dépendances. Vous pouvez également désormais consulter les traces X-Ray corrélées à des problèmes tels que des pannes, des erreurs et des problèmes de latence.	16 avril 2024
CloudWatch Container Insights, avec une observabilité améliorée pour Amazon EKS, ajoute la prise en charge de Windows	Vous pouvez désormais utiliser CloudWatch Container Insights avec une observabilité améliorée pour qu'Amazon EKS collecte des métriques à partir de nœuds de travail Windows sur des clusters Amazon EKS.	10 avril 2024

CloudWatchApplicationSignalsServiceRolePolicyPolitique IAM mise à jour	CloudWatch a modifié le champ d'application d'une autorisation dans CloudWatchApplicationSignalsServiceRolePolicy. L'étendue de l'`cloudwatch:GetMetricData` autorisation a été modifiée pour `*` permettre à Application Signals de récupérer des métriques à partir de sources situées dans des comptes liés.	8 avril 2024
Amazon CloudWatch Internet Monitor prend désormais en charge l'observabilité entre comptes	Vous pouvez désormais utiliser l'observabilité entre comptes d'Internet Monitor pour surveiller vos applications qui s'étendent sur plusieurs comptes Comptes AWS au sein d'un même. Région AWS	29 mars 2024

CloudWatchAgentServerPolicy et CloudWatchAgentAdminPolicypolitiques mises à jour	CloudWatch a ajouté des autorisations à la fois aux `CloudWatchAgentAdminPolicy` politiques `CloudWatchAgentServerPolicy` et pour permettre à l' CloudWatch agent de publier les traces de X-Ray et de modifier les périodes de conservation des groupes de journaux. Dans les deux politique `sxray:PutTraceSegments` , les `logs:PutRetentionPolicy` autorisations`xray:PutTelemetryRecords` ,`xray:GetSamplingRules` ,`xray:GetSamplingTargets` , `xray:GetSamplingStatisticSummaries` et ont été ajoutées	12 février 2024

Nouveau rôle lié au service et nouvelle politique IAM pour CloudWatch Network Monitor	CloudWatch a ajouté un nouveau rôle lié à un service, appelé. AWSServiceRoleForNetworkMonitor CloudWatch a ajouté ce nouveau rôle lié au service pour vous permettre de créer des moniteurs pour récupérer les métriques réseau entre les sous-réseaux sources et les adresses IP de destination. La nouvelle stratégie CloudWatchNetworkMonitorServiceRolePolicyIAM est attachée à ce rôle, et elle autorise l'extraction CloudWatch des métriques réseau en votre nom.	22 décembre 2023
CloudWatch lance Amazon CloudWatch Network Monitor	CloudWatch a publié une nouvelle fonctionnalité, Amazon CloudWatch Network Monitor. Il s'agit d'un nouveau service de surveillance active du réseau qui identifie si un problème de réseau existe au sein du AWS réseau ou de votre propre réseau d'entreprise.	22 décembre 2023

CloudWatchReadOnly Accesspolitique mise à jour	CloudWatch a ajouté des autorisations de lecture seule existantes pour Synthetics CloudWatch , X-Ray et CloudWatch RUM et de nouvelles autorisations de lecture seule pour CloudWatch Application Signals afin que les utilisateurs soumis à cette politique CloudWatchReadOnlyAccesspuissent trier et diagnostiquer les problèmes de santé du service signalés par Application Signals. CloudWatch L'`cloudwatch:GenerateQuery` autorisation a été ajoutée afin que les utilisateurs dotés de cette politique puissent générer une chaîne de requête CloudWatch Metrics Insights à partir d'une invite en langage naturel.	5 décembre 2023

CloudWatchFullAccessPolitique V2 mise à jour	CloudWatch a ajouté des autorisations existantes à la CloudWatchFullAccessV2 pour CloudWatch Synthetics, X-Ray et RUM CloudWatch , et ajouté de nouvelles CloudWatch autorisations pour les signaux d'application afin que les utilisateurs soumis à cette politique puissent gérer pleinement les signaux d'application afin de trier et de diagnostiquer les problèmes liés à l'état du service.	5 décembre 2023

Nouveau rôle lié à un service et nouvelle politique IAM	CloudWatch a ajouté un nouveau rôle lié à un service, appelé. AWSServiceRoleForCloudWatchApplicationSignals CloudWatch a ajouté ce nouveau rôle lié à un service pour permettre à CloudWatch Application Signals de collecter les données des CloudWatch journaux, les données de suivi X-Ray, CloudWatch les données métriques et les données de balisage à partir des applications que vous avez activées pour CloudWatch Application Signals. La nouvelle politique CloudWatchApplicationSignalsServiceRolePolicyIAM est attachée à ce rôle, et elle autorise CloudWatch Application Signals à collecter des données de surveillance et de balisage auprès d'autres services pertinents AWS .	30 novembre 2023

CloudWatch lance la version préliminaire de Application Signals	CloudWatch Application Signals est en cours de prévisualisation. Utilisez les signaux d'application pour piloter vos applications AWS afin de surveiller l'état actuel des applications, de créer des objectifs de niveau de service (SLO) et de suivre les performances des applications à long terme par rapport à vos objectifs commerciaux. Pour plus d'informations, veuillez consulter la rubrique Applications Signals.	30 novembre 2023
CloudWatch ajoute la prise en charge de l'interrogation d'autres sources de données	Vous pouvez l'utiliser CloudWatch pour interroger, visualiser et créer des alarmes pour des métriques provenant d'autres sources de données. Pour plus d'informations, voir Interroger des métriques provenant d'autres sources de données.	26 novembre 2023
CloudWatch Metrics Insights prend en charge la génération de requêtes en langage naturel	CloudWatch Metrics Insights prend en charge les requêtes en 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 les requêtes CloudWatch Metric Insights.	26 novembre 2023

CloudWatch publie Container Insights avec une observabilité améliorée pour Amazon EKS	CloudWatch a publié une nouvelle version de Container Insights. Cette version prend en charge l'observa bilité améliorée des clusters Amazon EKS et permet de collecter des métriques plus détaillées à partir de clusters exécutant Amazon EKS. Après l'installation, il collecte automatiquement des données télémétriques détaillées sur l'infrastructure et des journaux de conteneurs pour vos clusters Amazon EKS. Vous pouvez ensuite utiliser des tableaux de bord élaborés et immédiatement exploitables pour approfondir la télémétri e des applications et des infrastructures.	6 novembre 2023
CloudWatch Metric Streams permet une configuration rapide des partenaires	CloudWatch metric streams propose désormais une option de configuration rapide des partenaires, que vous pouvez utiliser pour configure r rapidement un flux métrique destiné à certains fournisseurs tiers.	17 octobre 2023

CloudWatch publie des recommandations d'alarme	CloudWatch Synthetics fournit désormais des recommandations d'alarme pour les métriques d'autres services. AWS Ces recommandations peuvent vous aider à identifier les métriques pour lesquelles vous devez définir des alarmes afin de suivre les bonnes pratiques en matière de surveillance de ces services.	16 octobre 2023
CloudWatch Synthetics lance Runtime -6.0 syn-nodejs-puppeteer	CloudWatch Synthetics a publié Runtime. `syn-nodejs-puppeteer-6.0`	26 septembre 2023
Ajoute la prise en charge CloudWatch d'Amazon Application Insights pour les applications multi-comptes	Vous pouvez désormais partager les applications CloudWatch Application Insights au-delà des limites de compte.	26 septembre 2023

Nouveau rôle lié à un service et nouvelle politique IAM	CloudWatch a ajouté un nouveau rôle lié à un service, appelé. AWSServiceRoleForCloudWatchMetrics_DbPerfInsights CloudWatch a ajouté ce nouveau rôle lié au service pour permettre de récupérer les métriques de Performance Insights CloudWatch à des fins d'alarme, de détection d'anomalies et de capture instantanée. La nouvelle politique AWSServiceRoleForCloudWatchMetrics_DbPerfInsightsServiceRolePolicyIAM est attachée à ce rôle, et elle autorise l'extraction des CloudWatch métriques Performance Insights en votre nom.	20 septembre 2023
Nouvelle fonction mathématique de métrique	CloudWatch a ajouté une nouvelle fonction mathématique des métriques`DB_PERF_INSIGHTS` , que vous pouvez utiliser pour récupérer les métriques Performance Insights à partir des services de AWS base de données à des fins d'alarme, de détection d'anomalies et de capture instantanée.	20 septembre 2023

CloudWatchReadOnly Accesspolitique mise à jour	CloudWatch a ajouté l'`application-autoscaling:DescribeScalingPolicies` autorisation CloudWatchReadOnly Accessafin que les utilisateurs dotés de cette politique puissent accéder aux informations relatives aux politiques Application Auto Scaling.	14 septembre 2023
CloudWatch support ajouté par l'agent pour AL2023	L' CloudWatch agent soutient AL2023.	08 août 2023
Nouvelle politique IAM gérée, V2 CloudWatchFullAccess	CloudWatch a ajouté une nouvelle politique CloudWatchFullAccessV2. Cette politique accorde un accès complet aux CloudWatch actions et aux ressources tout en définissant mieux les autorisations accordées à d'autres services tels qu'Amazon Amazon EC2 Auto Scaling SNS et.	1er août 2023
Rôle lié à un service mis à jour pour Amazon CloudWatch Internet Monitor — mise à jour d'une politique existante	Nouvelles autorisations, `elasticloadbalancing:DescribeLoadBalancers` et `ec2:DescribeNetworkInterfaces` , ajoutées au rôle lié à un service pour le Moniteur Internet afin de surveiller le trafic de ressources Network Load Balancer spécifiques.	25 juillet 2023

Ajout de la prise en charge des ressources Network Load Balancer dans Amazon Internet Monitor CloudWatch	Permet de créer un moniteur dans le Moniteur Internet avec des ressources Network Load Balancer spécifiques, afin de fournir des niveaux d'observabilité plus précis à votre application.	25 juillet 2023
Fonctionnalité de variables de tableau de bord	CloudWatch variables de tableau de bord publiées, que vous pouvez utiliser pour créer des tableaux de bord flexibles qui peuvent afficher rapidement différents contenus en fonction de la façon dont vous définissez un champ de saisie dans le tableau de bord. Par exemple, vous pouvez créer un tableau de bord capable de basculer rapidement entre différentes fonctions Lambda ou différent s ID d'instance Amazon EC2, ou un tableau de bord qui peut passer d'une région à l'autre. AWS Pour plus d'informations, voir Create flexible dashboard s with dashboard variables.	28 juin 2023

Le Moniteur Internet prend en charge la personnalisation du seuil des événements de santé	Le Moniteur Internet permet désormais de personnaliser le seuil lorsqu'un score de performance globale ou un score de disponibilité déclenche un événement de santé. Pour plus d'informations, consultez la section Suivi des performances et de la disponibilité en temps réel dans Amazon CloudWatch Internet Monitor.	26 juin 2023
Le Moniteur Internet prend en charge toutes les régions commerciales	Internet Monitor a ajouté sept nouvelles régions Régions AWS et prend désormais en charge toutes les régions commerciales.	19 juin 2023
Nouvelles versions d'extension Lambda Insights	CloudWatch a ajouté la version 1.0.229.0 de l'extension Lambda Insights pour les plateformes x86-64 et ARM64. Pour plus d'informations, consultez Available versions of the Lambda Insights extension .	12 juin 2023

CloudWatchReadOnly Accesspolitique mise à jour	CloudWatch a ajouté des autorisations à CloudWatchReadOnlyAccess. Les `logs:StopLiveTail` autorisations `logs:StartLiveTail` et ont été ajoutées afin que les utilisateurs soumis à cette politique puissent utiliser la console pour démarrer et arrêter CloudWatch les sessions Logs Live Tail. Pour plus d'informations, veuillez consulter Utilisation de Live Tail pour visualiser les journaux en temps quasi réel.	6 juin 2023
CloudWatch RUM ajoute le support pour les métriques personnalisées	Vous pouvez utiliser les moniteurs de l'application CloudWatch RUM pour créer des métriques personnalisées et les envoyer à CloudWatch et CloudWatch Evidently. Cette fonctionnalité inclut une mise à jour de la politique IAM ServiceRolePolicy gérée par AmazonCloudWatchRUM. Dans cette politique, une clé de condition a été modifiée afin que CloudWatch RUM puisse envoyer des métriques personnalisées à des espaces de noms de métriques personnalisés.	9 février 2023

Politiques gérées nouvelles et mises à jour pour CloudWatch	Pour favoriser CloudWatch l'observabilité entre comptes, les `CloudWatchReadOnlyAccess` politiques `CloudWatchFullAccess` et ont été mises à jour, et les nouvelles politiques gérées suivantes ont été ajoutées : `CloudWatchCrossAccountSharingConfiguration` `OAMFullAccess` , et. `OAMReadOnlyAccess` Pour plus d'informations, voir les CloudWatch mises à jour des politiques AWS gérées.	7 février 2023
CloudWatch Mises à jour de la politique de rôle liée au service Application Insights : mise à jour d'une politique existante.	CloudWatch Application Insights a mis à jour une politique de rôle AWS liée à un service existante.	19 décembre 2022
Amazon CloudWatch Application Insights prend en charge les applications conteneurisées et les microservices depuis la console Container Insights.	Vous pouvez afficher les problèmes détectés par CloudWatch Application Insights pour Amazon ECS et Amazon EKS sur votre tableau de bord Container Insights.	17 novembre 2021
Surveillance des bases de données SAP HANA par Amazon CloudWatch Application Insights.	Vous pouvez surveiller les bases de données SAP HANA à l'aide d'Application Insights.	15 novembre 2021

Amazon CloudWatch Application Insights prend en charge la surveillance de toutes les ressources d'un compte.	Vous pouvez intégrer et surveiller toutes les ressources d'un compte.	15 septembre 2021
Support CloudWatch d'Amazon Application Insights pour Amazon FSx.	Vous pouvez surveiller les métriques récupérées à partir d'Amazon FSx.	31 août 2021
SDK Metrics n'est plus pris en charge.	CloudWatch Les métriques du SDK ne sont plus prises en charge.	25 août 2021
Amazon CloudWatch Application Insights prend en charge la configuration de la surveillance des conteneurs.	Vous pouvez surveiller les conteneurs en utilisant les meilleures pratiques avec Amazon CloudWatch Application Insights.	18 mai 2021
Les flux de métriques sont disponibles pour tous	Vous pouvez utiliser les flux métriques pour diffuser en continu CloudWatch les métriques vers la destination de votre choix. Pour plus d'informations, consultez la section Streams métriques dans le guide de CloudWatch l'utilisateur Amazon.	31 mars 2021
Surveillance par Amazon CloudWatch Application Insights des bases de données Oracle sur Amazon RDS et Amazon EC2.	Vous pouvez surveiller les métriques et les journaux extraits d'Oracle avec Amazon CloudWatch Application Insights.	16 janvier 2021

Lambda Insights est disponible pour tous	CloudWatch Lambda Insights est une solution de surveillance et de dépannage pour les applications sans serveur exécutées sur. AWS Lambda Pour plus d'informations, consultez la section Utilisation de Lambda Insights dans le guide de CloudWatch l'utilisateur Amazon.	3 décembre 2020
Surveillance par Amazon CloudWatch Application Insights des métriques relatives aux exportateurs JMX de Prometheus.	Vous pouvez surveiller les métriques extraites de l'exportateur JMX Prometheus avec CloudWatch Amazon Application Insights.	20 novembre 2020
CloudWatch Synthetics publie une nouvelle version d'exécution	CloudWatch Synthetics a publié une nouvelle version d'exécution. Pour plus d'informations, consultez la section Versions de Canary Runtime dans le guide de CloudWatch l'utilisateur Amazon.	11 septembre 2020
Surveillance par Amazon CloudWatch Application Insights pour Postgre SQL sur Amazon RDS et Amazon EC2.	Vous pouvez surveiller les applications créées avec PostgreSQL exécutées sur Amazon RDS ou Amazon EC2.	11 septembre 2020

CloudWatch permet le partage de tableaux de bord	Vous pouvez désormais partager CloudWatch des tableaux de bord avec des personnes extérieures à votre organisation et à votre AWS compte. Pour plus d'informations, consultez la section Partage de CloudWatch tableaux de bord dans le guide de CloudWatch l'utilisateur Amazon.	10 septembre 2020
Configurez des moniteurs pour les applications .NET à l'aide de SQL Server sur le backend avec CloudWatch Application Insights	Vous pouvez utiliser le didacticiel de documentation pour vous aider à configurer des moniteurs pour les applications .NET à l'aide de SQL Server sur le backend avec CloudWatch Application Insights.	19 août 2020
AWS CloudFormation support pour les applications Amazon CloudWatch Application Insights.	Vous pouvez ajouter la surveillance CloudWatch Application Insights, y compris les indicateurs clés et la télémétrie, à votre application, à votre base de données et à votre serveur Web, directement à partir de AWS CloudFormation modèles.	30 juillet 2020
Surveillance par Amazon CloudWatch Application Insights pour les clusters de bases de données Aurora for MySQL.	Vous pouvez surveiller les clusters de bases de données Aurora for MySQL (RDS Aurora) avec Amazon CloudWatch Application Insights.	2 juillet 2020

CloudWatch Disponibilité générale de Contributor Insights	CloudWatch Contributor Insights est désormais disponible pour tous. Il vous permet d'analyser les données des journaux et de créer des séries chronologiques qui affichent les données du contributeur. Vous pouvez voir les mesures concernant les premiers contributeurs, le nombre total de contributeurs uniques et leur utilisation. Pour plus d'informations, consultez la section Utilisation des informations sur les contributeurs pour analyser les données à haute cardinalité dans le guide de CloudWatch l'utilisateur Amazon.	2 avril 2020
CloudWatch Aperçu public de Synthetics	CloudWatch Synthetics est désormais disponible en avant-première publique. Il vous permet de créer des scripts Canary pour surveiller vos points de terminaison et vos API. Pour plus d'informations, consultez la section Utilisation des canaries dans le guide de CloudWatch l'utilisateur Amazon.	25 novembre 2019

CloudWatch Aperçu public de Contributor Insights	CloudWatch Contributor Insights est désormais disponible en version préliminaire publique. Il vous permet d'analyser les données des journaux et de créer des séries chronologiques qui affichent les données du contributeur. Vous pouvez voir les mesures concernant les premiers contributeurs, le nombre total de contributeurs uniques et leur utilisation. Pour plus d'informations, consultez la section Utilisation des informations sur les contributeurs pour analyser les données à haute cardinalité dans le guide de CloudWatch l'utilisateur Amazon.	25 novembre 2019
CloudWatch ServiceLens fonction de lancement	ServiceLens Améliore l'observabilité de vos services et applications en vous permettant d'intégrer les traces, les métriques, les journaux et les alarmes en un seul endroit. ServiceLens s' CloudWatch intègre AWS X-Ray pour fournir une end-to-end vue de votre application.	21 novembre 2019

CloudWatch À utiliser pour gérer de manière proactive vos quotas AWS de service	Vous pouvez l'utiliser CloudWatch pour gérer vos quotas de AWS service de manière proactive. CloudWatch les statistiques d'utilisation fournissent une visibilité sur l'utilisation des ressources et les opérations d'API par votre compte. Pour plus d'informations, consultez la section Intégration des Quotas de Service et Mesures d'utilisation dans le Guide de CloudWatch l'utilisateur Amazon.	19 novembre 2019
CloudWatch envoie des événements lorsque l'état des alarmes change	CloudWatch envoie désormais un événement à Amazon EventBridge lorsqu'une CloudWatch alarme change d'état. Pour plus d'informations, consultez Alarm Events et EventBridge le guide de CloudWatch l'utilisateur Amazon.	8 octobre 2019

Container Insights	CloudWatch Container Insights est désormais disponible pour tous. Il vous permet de collecter, regrouper et récapituler les métriques et les journaux de vos applications et microservices conteneurisés. Pour plus d'informations, consultez la section Utilisation de Container Insights dans le guide de CloudWatch l'utilisateur Amazon.	30 août 2019
Mises à jour des métriques dans la version préliminaire de Container Insights sur Amazon EKS et Kubernetes	La version préliminaire publique de Container Insights sur Amazon EKS et Kubernetes a été mise à jour. InstanceId est désormais incluse en tant que dimension dans les instances EC2 du cluster. Cela permet aux alarmes qui ont été créées sur ces métriques de déclencher les actions EC2 suivantes : Stop (Arrêt), Terminate (Résiliation), Reboot (Redémarrage) ou Recover (Récupération). De plus, les métriques pod et service sont désormais signalées par l'espace de noms Kubernetes pour simplifier la surveillance et les alarmes sur les métriques par espace de noms.	19 août 2019

Mises à jour pour AWS Systems Manager OpsCenter l'intégration	Mises à jour sur la façon dont CloudWatch Application Insights s'intègre à Systems Manager OpsCenter.	7 août 2019
CloudWatch métriques d'utilisation	CloudWatch les statistiques d'utilisation vous aident à suivre l'utilisation de vos CloudWatch ressources et à respecter les limites de votre service. Pour plus d'informations, consultez https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch-Usage-Metrics.html.	6 août 2019
CloudWatch Aperçu public de Container Insights	CloudWatch Container Insights est désormais disponible en version préliminaire publique. Il vous permet de collecter, regrouper et récapituler les métriques et les journaux de vos applications et microservices conteneurisés. Pour plus d'informations, consultez la section Utilisation de Container Insights dans le guide de CloudWatch l'utilisateur Amazon.	9 juillet 2019

CloudWatch Aperçu public de Anomaly Detection	CloudWatch la détection des anomalies est désormais disponible en avant-pre mière publique. CloudWatc h applique des algorithmes d'apprentissage automatique aux données passées d'une métrique afin de créer un modèle des valeurs attendues de la métrique. Vous pouvez utiliser ce modèle pour la visualisation et pour définir des alarmes. Pour plus d'informa tions, consultez la section Utilisation de la détection des CloudWatch anomalies dans le guide de CloudWatch l'utilisateur Amazon.	9 juillet 2019
CloudWatch Informations sur les applications pour .NET et SQL Server	CloudWatch Application Insights for .NET et SQL Server facilite l'observabilité des applications .NET et SQL Server. Cette solution peut vous aider à configurer les meilleurs surveillances pour vos ressources d'application, afin d'analyser en continu les données à la recherche de problèmes liés à vos applicati ons.	21 juin 2019

CloudWatch section des agents réorganisée	La documentation de l' CloudWatch agent a été réécrite pour être plus claire, en particulier pour les clients utilisant la ligne de commande pour installer et configurer l'agent. Pour plus d'informations, 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 .	28 mars 2019
Fonction SEARCH (RECHERCHE) ajoutée aux expressions mathématiques de métrique	Vous pouvez désormais utiliser une fonction de RECHERCHE dans des expressions mathématiques de métrique. Cela vous permet de créer des tableaux de bord qui se mettent à jour automatiquement lorsque de nouvelles ressources sont créées qui correspondent à la requête de recherche. Pour plus d'informations, consultez la section Utilisation des expressions de recherche dans les graphiques du guide de CloudWatch l'utilisateur Amazon.	21 mars 2019

AWS Métriques du SDK pour le support aux entreprises	SDK Metrics vous aide à évaluer l'état de vos AWS services et à diagnostiquer le temps de latence causé par l'atteinte des limites d'utilisation de votre compte ou par une panne de service. Pour plus d'informations, consultez la section Surveiller les applications à l'aide des métriques du AWS SDK dans le guide de CloudWatch l'utilisateur Amazon.	11 décembre 2018
Alarmes sur les expressions mathématiques	CloudWatch prend en charge la création d'alarmes basées sur des expressions mathématiques métriques . Pour plus d'informations, consultez la section Alarmes sur les expressions mathématiques dans le guide de CloudWatch l'utilisateur Amazon.	20 novembre 2018

Nouvelle page d'accueil de la CloudWatch console	Amazon a créé une nouvelle page d'accueil dans la CloudWatch console, qui affiche automatiquement les indicateurs clés et les alarmes pour tous les AWS services que vous utilisez. Pour plus d'informations, consultez Getting Started with Amazon CloudWatch dans le guide de CloudWatch l'utilisateur Amazon.	19 novembre 2018
AWS CloudFormation modèles pour l' CloudWatch agent	Amazon a chargé AWS CloudFormation des modèles que vous pouvez utiliser pour installer et mettre à jour l' CloudWatchagent. Pour plus d'informations, consultez Installer l' CloudWatch agent sur les nouvelles instances AWS CloudFormationà l'aide du guide de CloudWatch l'utilisateur Amazon.	9 novembre 2018

Améliorations apportées à l' CloudWatch agent	L' CloudWatch agent a été mis à jour pour fonctionner à la fois avec les protocole s StatsD et collectd. Il offre également une meilleure prise en charge entre comptes. Pour plus d'informations, consultez les sections Récupérer des métriques personnalisées avec StatsD, Récupérer des métriques personnalisées avec collectd et Envoyer des métriques et des journaux à un autre AWS compte dans le guide de l'utilisateur Amazon CloudWatch .	28 septembre 2018
Prise en charge des points de terminaison Amazon VPC	Vous pouvez désormais établir une connexion privée entre votre VPC et. CloudWatc h Pour plus d'informations, consultez la section Utilisation CloudWatch avec les points de terminaison VPC d'interfa ce dans le guide de l'utilisateur Amazon CloudWatch .	28 juin 2018

Le tableau suivant décrit les modifications importantes apportées au guide de CloudWatch l'utilisateur Amazon avant juin 2018.

Modification	Description	Date de publication
Mathématiques appliquées aux métriques	Vous pouvez désormais exécuter des expressio ns mathématiques sur CloudWatch les métriques , afin de produire de nouvelles séries chronolog iques que vous pouvez ajouter aux graphiques de	4 avril 2018

Modification	Description	Date de publication
	votre tableau de bord. Pour plus d'informations, consultez Utilisation des mathématiques appliquées aux métriques.	
Alarmes « M sur N »	Vous pouvez désormais configurer une alarme afin qu'elle se déclenche en fonction de points de données « M sur N » dans n'importe quel intervalle d'évaluation d'alarme. Pour de plus amples informations, consultez Évaluation d'une alerte.	8 décembre 2017
CloudWatch agent	Un nouvel CloudWatch agent unifié a été publié. Vous pouvez utiliser l'agent multiplateforme unifié pour collecter des métriques système et des fichiers journaux personnalisés à partir d'instances Amazon EC2 et de serveurs sur site. Le nouvel agent prend en charge Windows et Linux, et vous permet de personnaliser les métriques collectées, notamment des métriques de sous-ressource, par exemple, cœur par CPU. Pour de plus amples informations, consultez Collectez des métriques, des journaux et des traces avec l' CloudWatch agent.	7 septembre 2017
Métriques de passerelle NAT	Ajout de métriques pour la passerelle NAT Amazon VPC.	7 septembre 2017
Métriques haute résolution	Vous pouvez désormais, si vous le souhaitez, configurer les métriques personnalisées en tant que métriques haute résolution, avec une granularité de seulement 1 seconde. Pour de plus amples informations, consultez Métriques haute résolution.	26 juillet 2017
API de tableau de bord	Vous pouvez désormais créer, modifier et supprimer des tableaux de bord à l'aide d'API et de l' AWS CLI. Pour plus d'informations, consultez Création d'un CloudWatch tableau de bord.	6 juillet 2017

Modification	Description	Date de publication
AWS Direct Connect métriques	Ajout de métriques pour AWS Direct Connect.	29 juin 2017
Métriques VPN Amazon VPC	Ajout de métriques pour VPN Amazon VPC.	15 mai 2017
AppStream Métriques 2.0	Ajout de métriques pour la AppStream version 2.0.	8 mars 2017
CloudWatch sélecteur de couleurs de console	Vous pouvez désormais choisir la couleur de chaque métrique dans vos widgets de tableau de bord. Pour de plus amples informations, consultez Modifier un graphique sur un CloudWatch tableau de bord .	27 février 2017
Alarmes dans les tableaux de bord	Vous pouvez désormais ajouter des alarmes aux tableaux de bord. Pour de plus amples informati ons, consultez Ajouter ou supprimer un widget d'alarme dans un CloudWatch tableau de bord .	15 février 2017
Ajout de métriques pour Amazon Polly	Ajout de métriques pour Amazon Polly.	1er décembre 2016
Nouvelles métriques pour le service géré Amazon pour Apache Flink	Nouvelles métriques pour le service géré Amazon pour Apache Flink.	1er décembre 2016
Ajout de la prise en charge des statistiques sur les centiles	Vous pouvez spécifier un centile en utilisant jusqu'à deux décimales (par exemple, p95.45). Pour de plus amples informations, consultez Centiles.	17 novembre 2016

Modification	Description	Date de publication
Ajout de métriques pour Amazon Simple En	Ajout de métriques pour Amazon Simple Email Service.	2 novembre 2016
Conservation de métriques mises à jour	Amazon conserve CloudWatch désormais les données métriques pendant 15 mois au lieu de 14 jours.	1 novembre 2016
Interface de la console des métriques mises à jour	La CloudWatch console est mise à jour avec des améliorations apportées aux fonctionnalités existantes et de nouvelles fonctionnalités.	1 novembre 2016
Ajout de métriques pour Amazon Elastic Tra r	Ajout de métriques pour Amazon Elastic Transcoder.	20 septembre 2016
Ajout de métriques pour Amazon API Gatev	Ajout de métriques pour Amazon API Gateway.	9 septembre 2016
Indicateurs ajoutés pour AWS Key Management Service	Ajout de métriques pour AWS Key Management Service.	9 septembre 2016
Ajout de métriques pour les nouveaux Applicati on Load Balancers pris en charge par Elastic Load Balan	Ajout de métriques pour les Application Load Balancers.	11 août 2016

Modification	Description	Date de publication
Ajout de nouvelles NetworkPacketsOut statistiques NetworkPacketsIn et de nouvelles mesures pour Amazon EC2	Ajout de nouvelles NetworkPacketsOut métriques NetworkPacketsIn et de nouvelles mesures pour Amazon EC2.	23 mars 2016
Ajout de nouvelles métriques pour les parcs d'instances Spot Amazon EC2	Ajout de nouvelles métriques pour les parcs d'instances Spot Amazon EC2.	21 mars 2016
Ajout de nouvelles métriques CloudWatch Logs	Ajout de nouvelles métriques CloudWatch Logs.	10 mars 2016
Ajout d'Amazon OpenSearch Service, de AWS WAF métriques et de dimensions	Ajout d'Amazon OpenSearch Service, de AWS WAF métriques et de dimensions.	14 octobre 2015

Modification	Description	Date de publication
Support supplémentaire pour les CloudWatch tableaux de bord	Les tableaux de bord sont des pages d'accueil personnalisables dans la CloudWatch console que vous pouvez utiliser pour surveiller vos ressources dans une seule vue, même celles qui sont réparties dans différentes régions. Pour plus d'informations, consultez Utilisation des tableaux de CloudWatch bord Amazon.	8 octobre 2015
AWS Lambda Mesures et dimensions ajoutées	AWS Lambda Indicateurs et dimensions ajoutés.	4 septembre 2015
Ajout des métriques et dimensions Amazon Elastic Co	Ajout des métriques et dimensions Amazon Elastic Container Service.	17 août 2015
Ajout des métriques et dimensions Amazon Simple St	Ajout des métriques et dimensions Amazon Simple Storage Service.	26 juillet 2015
Nouvelle fonctionnalité : redémarrage de l'action d'alarme	Ajout de la nouvelle action d'alarme de redémarrage et du nouveau rôle IAM à utiliser avec les actions d'alarme. Pour de plus amples informations, consultez Création d'alarmes qui arrêtent, mettent hors service, redémarrent ou récupèrent une instance EC2.	23 juillet 2015
WorkSpaces Mesures et dimensions Amazon ajoutées	Ajout de WorkSpaces statistiques et de dimensions Amazon.	30 avril 2015

Modification	Description	Date de publication
Ajout des métriques et dimensions Amazon Machine L	Ajout des métriques et dimensions Amazon Machine Learning.	9 avril 2015
Nouvelle fonction : actions de l'alarme de récupération de l'instance Amazon EC2	Mise à jour des actions d'alarme pour inclure la nouvelle action de récupération d'instance EC2. Pour de plus amples informations, consultez Création d'alarmes qui arrêtent, mettent hors service, redémarrent ou récupèrent une instance EC2.	12 mars 2015
CloudSear ch Mesures CloudFront et dimension s Amazon et Amazon ajoutées	Ajout de CloudSearch métriques CloudFront et de dimensions Amazon et Amazon.	6 mars 2015
Ajout des métriques et dimensions Amazon Simple W	Ajout des métriques et dimensions Amazon Simple Workflow Service.	9 mai 2014
Guide mis à jour pour ajouter la prise en charge de AWS CloudTrail	Ajout d'une nouvelle rubrique expliquant comment AWS CloudTrail enregistrer l'activité sur Amazon CloudWatch. Pour plus d’informations, consultez Journalisation des appels CloudWatch d'API Amazon avec AWS CloudTrail.	30 avril 2014

Modification	Description	Date de publication
Guide mis à jour pour utiliser le nouveau AWS Command Line Interface (AWS CLI)	La AWS CLI est une CLI multiservice dotée d'une installation simplifiée, d'une configuration unifiée et d'une syntaxe de ligne de commande cohérente. La AWS CLI est prise en charge sous Linux/Unix, Windows et Mac. Les exemples de CLI présentés dans ce guide ont été mis à jour pour utiliser la nouvelle AWS CLI. Pour plus d'informations sur l'installation et la configuration de la nouvelle AWS CLI, consultez la section Getting Set Up with the AWS CLI Interface dans le guide de AWS Command Line Interface l'utilisateur.	21 février 2014
Ajout d'Amazon Redshift, de AWS OpsWorks métriques et de dimensions	Ajout d'Amazon Redshift, de AWS OpsWorks métriques et de dimensions.	16 juillet 2013
Ajout des métriques et dimensions Amazon Route 53	Ajout des métriques et dimensions Amazon Route 53.	26 juin 2013
Nouvelle fonctionnalité : Amazon CloudWatch Alarm Actions	Ajout d'une nouvelle section pour documenter les actions CloudWatch d'alarme Amazon, que vous pouvez utiliser pour arrêter ou mettre fin à une instance Amazon Elastic Compute Cloud. Pour plus d'informations, consultez Création d'alarmes qui arrêtent, mettent hors service, redémarrent ou récupèrent une instance EC2.	8 janvier 2013
Mise à jour des métriques EBS	Mise à jour des métriques EBS pour inclure deux nouvelles métriques pour les volumes IOPS provisionnées.	20 novembre 2012

Modification	Description	Date de publication
Nouvelles alertes de facturation	Vous pouvez désormais surveiller vos AWS frais à l'aide CloudWatch des statistiques Amazon et créer des alarmes pour vous avertir lorsque vous avez dépassé le seuil spécifié. Pour plus d'informations, consultez Créez une alarme de facturation pour surveiller vos AWS frais estimés.	10 mai 2012
Nouvelles métriques	Vous pouvez désormais accéder à six nouvelles métriques Elastic Load Balancing qui prennent en considération différents codes de réponse HTTP.	19 octobre 2011
Nouvelle fonctionnalité	Vous pouvez désormais accéder aux métriques d'Amazon EMR.	30 juin 2011
Nouvelle fonctionnalité	Vous pouvez maintenant accéder aux métriques d'Amazon Simple Notification Service et d'Amazon Simple Queue Service.	14 juillet 2011
Nouvelle fonction	Ajout d'informations sur l'utilisation de l'API `PutMetricData` pour publier des métriques personnalisées. Pour de plus amples informations, consultez Publier des métriques personnalisées .	10 mai 2011
Conservation de métriques mises à jour	Amazon conserve CloudWatch désormais l'historique d'une alarme pendant deux semaines au lieu de six semaines. La période de conservation des alarmes correspond ainsi à la période de conservation des données des métriques.	7 avril 2011
Nouvelle fonctionnalité	Ajout de la possibilité d'envoyer des notifications Amazon Simple Notification Service ou Auto Scaling lorsqu'une métrique a dépassé un seuil. Pour de plus amples informations, consultez alertes.	2 décembre 2010

Modification	Description	Date de publication
Nouvelle fonctionnalité	Un certain nombre d' CloudWatch actions incluent désormais les NextToken paramètres MaxRecord s et, qui vous permettent de contrôler les pages de résultats à afficher.	2 décembre 2010
Nouvelle fonctionnalité	Ce service s'intègre désormais à AWS Identity and Access Management (IAM).	2 décembre 2010

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.