Amazon CloudWatch Events

Guide de l’utilisateur

Amazon CloudWatch Events: Guide de l’utilisateur

Table of Contents

Qu'est-ce qu'Amazon CloudWatch Events ?

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Amazon CloudWatch Events fournit un flux d'événements système en quasi temps réel qui décrit les modifications apportées aux ressources Amazon Web Services (AWS). A l'aide de règles simples et rapidement configurées, vous pouvez faire correspondre des événements et les acheminer vers un ou plusieurs flux ou une ou plusieurs fonctions cibles. CloudWatch Events prend connaissance des changements opérationnels à mesure qu'ils se produisent. CloudWatch Events répond à ces changements opérationnels et, le cas échéant, prend des mesures correctives en envoyant des messages pour répondre à l'environnement, en activant des fonctions, en procédant à des modifications et en capturant des informations de statut.

Vous pouvez également utiliser CloudWatch Events pour planifier des actions automatisées qui se déclenchent automatiquement à certaines heures à l'aide d'expressions cron ou de fréquence. Pour de plus amples informations, veuillez consulter Expression de planification des règles (p. 34).

Vous pouvez configurer les différents services AWS en tant que cibles pour CloudWatch Events :

- Instances Amazon EC2
- AWS LambdaFonctions
- Flux dans Amazon Kinesis Data Streams
- Flux de diffusion dans Amazon Kinesis Data Firehose
- Groupes de journaux dans Amazon CloudWatch Logs
- Tâches Amazon ECS
- Run Command de Systems Manager
- Systems Manager Automation
- AWS Batch tâches
- Machines d'état Step Functions
- Pipeline dans CodePipeline
- Projets CodeBuild
- Modèles d'évaluation Amazon Inspector
- Rubriques Amazon SNS
- Files d'attente Amazon SQS
- Cibles intégrées : `EC2 CreateSnapshot API call`, `EC2 RebootInstances API call`, `EC2 StopInstances API call` et `EC2 TerminateInstances API call`.

- Le bus d'événement par défaut d'un autre compte AWS

Concepts

Avant de commencer à utiliser CloudWatch Events, vous devez bien comprendre les concepts suivants :

- Événements : les événements indiquent un changement dans votre environnement AWS. Les ressources AWS peuvent générer des événements lorsque leur état change. Par exemple, Amazon EC2 génère un événement lorsque l'état d'une instance EC2 passe de « pending » (en attente) à « running » (en cours d'exécution), ou quand Amazon EC2 Auto Scaling génère des événements au moment de lancer des instances ou d'y mettre fin. AWS CloudTrail publie les événements lorsque vous effectuez des appels d'API. Vous pouvez générer des événements de niveau application personnalisés et les publier dans CloudWatch Events. Vous pouvez également configurer des événements planifiés qui sont générés de façon périodique. Pour obtenir une liste des services qui génèrent des événements et des exemples d'événements de chaque service, consultez la rubrique Exemples d'événements CloudWatch Events provenant de services pris en charge (p. 43).
- Règles : les règles correspondent à des événements entrants et les acheminent vers des cibles pour être traités. Une seule règle peut aboutir à plusieurs cibles, qui sont toutes traitées en parallèle. Les règles ne sont pas traitées dans un ordre particulier. Les différentes parties d'une organisation peuvent ainsi rechercher et traiter les événements qui les intéressent. Une règle peut personnaliser le fichier JSON envoyé à la cible en transmettant uniquement certaines parties du fichier ou en les remplaçant par une constante.
- Cibles : les cibles sont chargées de traiter le événements. Les cibles peuvent englober les instances Amazon EC2, les fonctions AWS Lambda, les flux Kinesis, les tâches Amazon ECS, les machines d'état Step Functions, les rubriques Amazon SNS, les files d'attente Amazon SQS et les cibles intégrées. Les cibles reçoivent les événements au format JSON.

  Les cibles d'une règle doivent se trouver dans la même région que la règle.

Services AWS connexes

Les services suivants sont utilisés en association avec CloudWatch Events :

- AWS CloudTrail vous permet de surveiller les appels de l'API CloudWatch Events pour votre compte, y compris ceux effectués par la AWS Management Console, la AWS CLI et d'autres services. Lorsque la journalisation CloudTrail est activée, CloudWatch Events écrit les fichiers journaux dans un compartiment S3. Chaque fichier journal contient un ou plusieurs enregistrements, selon le nombre d'actions effectuées pour satisfaire une demande. Pour de plus amples informations, veuillez consulter Journalisation des appels d'API Amazon CloudWatch Events avec AWS CloudTrail (p. 113).
- AWS CloudFormation permet de modéliser et de configurer vos ressources AWS. Vous créez un modèle qui décrit les ressources AWS que vous voulez, et AWS CloudFormation s'occupe de la mise en service et de la configuration de ces ressources. Vous pouvez utiliser les règles CloudWatch Events figurant dans vos modèles AWS CloudFormation. Pour plus d'informations, veuillez consulter AWS::Events::Rule dans le Guide de l'utilisateur AWS CloudFormation.
- AWS Config vous permet d'enregistrer les modifications de configuration apportées à vos ressources AWS. Il indique comment les ressources sont liées entre elles et comment elles ont été configurées dans le passé, pour que vous puissiez observer comment les configurations et les relations changent au fil du temps. Vous pouvez également créer des règles AWS Config pour vérifier si vos ressources sont conformes ou non conformes aux stratégies de votre organisation. Pour plus d'informations, consultez le Guide du développeur AWS Config.
- AWS Identity and Access Management (IAM) vous permet de contrôler en toute sécurité l'accès aux ressources AWS pour vos utilisateurs. Utilisez IAM pour contrôler qui peut utiliser vos ressources AWS

(authentification), quelles ressources ils peuvent utiliser et comment ils peuvent les utiliser (autorisation). Pour plus d'informations, consultez le Guide de l'utilisateur IAM.

- Amazon Kinesis Data Streams permet une extraction et un regroupement rapides et pratiquement continus des données. Le type de données utilisées inclut des données de journaux d'infrastructure informatique, des journaux d'applications, des réseaux sociaux, des flux de données du marché et des données de flux de clics Web. Comme le temps de réponse pour la récupération et le traitement des données est en temps réel, le traitement est généralement léger. Pour plus d'informations, consultez le Guide du développeur Amazon Kinesis Data Streams.
- AWS Lambda vous permet de créer des applications très réactives par rapport aux nouvelles informations. Chargez votre code d'application en tant que fonctions Lambda et Lambda exécutera votre code sur une infrastructure de calcul haute disponibilité. Lambda effectue toute l'administration des ressources de calcul, y compris la maintenance du serveur et du système d'exploitation, l'approvisionnement des capacités et leur mise à l'échelle automatique, le déploiement du code et des correctifs de sécurité, ainsi que la surveillance et la journalisation du code. Pour plus d'informations, consultez le Manuel du développeur AWS Lambda.

Configuration d'Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Pour utiliser Amazon CloudWatch Events, vous avez besoin d'un compte AWS. Votre compte AWS vous permet d'utiliser des services (par exemple, Amazon EC2) pour générer des événements susceptibles d'être affichés dans la console CloudWatch, une interface web. En outre, vous pouvez installer et configurer l'AWS Command Line Interface (AWS CLI) pour utiliser une interface de ligne de commande.

S'inscrire à Amazon Web Services (AWS)

Lorsque vous créez un compte AWS, nous l'inscrivons automatiquement à tous les services AWS. Vous payez des frais uniquement en fonction des services que vous utilisez réellement.

Si vous possédez déjà un compte AWS, passez à la prochaine étape. Si tel n'est pas le AWS cas, observez la procédure suivante pour en créer un.

Pour créer 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.

Se connecter à la console Amazon CloudWatch

Pour vous connecter à la console Amazon CloudWatch

1. Connectez-vous à AWS Management Console et ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Si nécessaire, changez la région. Dans la barre de navigation, choisissez la région où se trouvent vos ressources AWS.
3. Dans le volet de navigation, sélectionnez Events.

Informations d'identification de compte

Même si vous pouvez utiliser les informations d'identification de votre utilisateur racine pour accéder à CloudWatch Events, nous vous recommandons d'utiliser un compte IAM (AWS Identity and Access

Management). Si vous utilisez un compte IAM pour accéder à CloudWatch, vous devez avoir les autorisations suivantes :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "events:*",
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Configuration de l'interface de ligne de commande

Vous pouvez utiliser la AWS CLI pour effectuer des opérations CloudWatch Events.

Pour plus d'informations sur l'installation et la configuration de la AWS CLI, consultez Préparation de la configuration de AWS Command Line Interface dans le Guide de l'utilisateur AWS Command Line Interface.

Points de terminaison régionaux

Vous devez activer des points de terminaison régionaux (par défaut) pour utiliser CloudWatch Events. Pour de plus amples informations, veuillez consulter Activation et désactivation de AWS STS dans une région AWS dans le Guide de l'utilisateur IAM.

Mise en route avec Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Utilisez les procédures de cette section pour créer et supprimer des règles CloudWatch Events. Il s'agit de procédures générales utilisables pour n'importe quelle source ou cible d'événement. Pour accéder à des didacticiels écrits pour des scénarios et cibles spécifiques, consultez Didacticiels.

Chaque règle

Table des matières

Restrictions

- Les cibles que vous associez à une règle doivent se trouver dans la même région que la règle.
- Certains types de cibles peuvent ne pas être disponibles dans toutes les régions. Pour plus d'informations, consultez Régions et points de terminaison dans la Référence générale d'Amazon Web Services.
- La création de règles avec des cibles intégrées est uniquement prise en charge dans la AWS Management Console.
- Si vous créez une règle avec une file d'attente Amazon SQS chiffrée en tant que cible, la section suivante doit être incluses dans votre politique de clé KMS. Cela permet à l'événement d'être correctement remis dans la file d'attente chiffrée.

```
{
                "Sid": "Allow CWE to use the key",
                "Effect": "Allow",
                "Principal": {
                                "Service": "events.amazonaws.com"
                },
                "Action": [
                                "kms:Decrypt",
                                "kms:GenerateDataKey"
                ],
                "Resource": "*"
```

```
}
```

Création d'une règle CloudWatch Events qui se déclenche lors d'un événement

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Utilisez les étapes suivantes pour créer une règle CloudWatch Events qui se déclenche sur un événement émis par un service AWS.

Pour créer une règle qui se déclenche sur un événement :

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Choisissez Event Pattern, Build event pattern to match events by service.
   b. Pour Nom du service, choisissez le service qui émet l'événement devant déclencher la règle.
   c. Pour Type d'événement, choisissez l'événement spécifique qui doit déclencher la règle. Si la seule option est AWS API Call via CloudTrail (Appel d'API AWS via CloudTrail), le service sélectionné n'émet pas d'événements et vous pouvez uniquement baser les règles sur les appels d'API effectués pour ce service. Pour plus d'informations sur la création de ce type de règle, consultez Création d'une règle CloudWatch Events qui se déclenche lors d'un appel d'API AWS à l'aide de AWS CloudTrail (p. 8).
   d. Selon le service qui émet l'événement, vous pouvez voir des options pour Any... et Specific.... Choisissez Any... pour que l'événement se déclenche sur n'importe quel type de l'événement sélectionné, ou Specific... pour choisir un ou plusieurs types d'événement spécifiques.
4. Pour Cibles, sélectionnez Add Target (Ajouter une cible) et choisissez le service AWS qui doit agir lorsqu'un événement du type sélectionné est détecté.
5. Dans les autres champs de cette section, entrez des informations spécifiques à ce type de cible, le cas échéant.
6. Pour de nombreux types de cibles, CloudWatch Events a besoin d'autorisations pour envoyer des événements à la cible. Dans ce cas, CloudWatch Events peut créer le rôle IAM nécessaire à l'exécution de votre événement :
   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant.
7. Le cas échéant, répétez les étapes 4 à 6 afin d'ajouter une autre cible pour cette règle.
8. Sélectionnez Configure details. Dans Rule definition, saisissez un nom et une description pour la règle.

   Le nom de la règle doit être unique au sein de cette région.
9. Choisissez Create rule.

Création d'une règle CloudWatch Events qui se déclenche lors d'un appel d'API AWS à l'aide de AWS CloudTrail

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Pour créer une règle qui se déclenche sur une action exécutée par un service AWS qui n'émet pas d'événements, vous pouvez baser la règle sur les appels d'API effectués par ce service. Les appels d'API sont enregistrés par AWS CloudTrail. Pour plus d'informations sur les appels d'API que vous pouvez utiliser comme déclencheurs pour les règles, consultez Services pris en charge par l'historique des événements CloudTrail.

Les règles dans CloudWatch Events fonctionnent uniquement dans la région où elles ont été créées. Si vous configurez CloudTrail pour suivre les appels d'API dans plusieurs régions, et que vous souhaitez qu'une règle basée sur CloudTrail se déclenche dans chacune de ces régions, vous devez créer une règle distincte dans chaque région que vous souhaitez suivre.

Tous les événements fournis via CloudTrail ont `AWS API Call via CloudTrail` comme valeur pour `detail-type`.

> **Note**
>
> Dans CloudWatch Events, il est possible de créer des règles qui conduisent à des boucles infinies, dans lesquelles une règle est exécutée de manière répétée. Par exemple, une règle peut détecter que les listes de contrôle d'accès (ACL) ont été modifiées sur un compartiment S3 et lancer un logiciel pour les modifier afin qu'elles aient l'état souhaité. Si la règle n'est pas correctement écrite, la modification suivante des listes de contrôle d'accès (ACL) déclenche à nouveau la règle, créant ainsi une boucle infinie.
> Pour éviter ce problème, écrivez les règles de sorte que les actions déclenchées ne relancent pas la même règle. Par exemple, votre règle pourrait ne s'exécuter que si les listes ACL s'avèrent être dans un état incorrect plutôt qu'après une modification.
> Une boucle infinie peut rapidement entraîner des coûts plus importants que prévu. Nous vous recommandons d'utiliser les budgets, qui vous avertissent lorsque les frais dépassent votre limite spécifiée. Pour plus d'informations, consultez Gestion des coûts avec les budgets.

Pour créer une règle qui se déclenche sur un appel d'API via CloudTrail :

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :

   a. Choisissez Event Pattern, Build event pattern to match events by service.
   b. Pour Nom du service, choisissez le service qui utilise les opérations d'API à utiliser comme déclencheur.
   c. Pour Event Type (Type d'événement), sélectionnez AWS API Call via CloudTrail (Appel d'API AWS via CloudTrail).

   d. Pour déclencher votre règle lorsque toute opération d'API pour ce service est appelée, choisissez Any operation (Toute opération). Pour déclencher votre règle uniquement lorsque certaines opérations d'API sont appelées, choisissez Specific operation(s) (Opérations spécifiques), tapez le nom de l'opération dans la zone suivante, puis appuyez sur ENTRÉE. Pour ajouter d'autres opérations, choisissez +.
4. Pour Cibles, sélectionnez Add Target (Ajouter une cible) et choisissez le service AWS qui doit agir lorsqu'un événement du type sélectionné est détecté.
5. Dans les autres champs de cette section, entrez des informations spécifiques à ce type de cible, le cas échéant.
6. Pour de nombreux types de cibles, CloudWatch Events a besoin d'autorisations pour envoyer des événements à la cible. Dans ce cas, CloudWatch Events peut créer le rôle IAM nécessaire à l'exécution de votre événement :

   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant.
7. Le cas échéant, répétez les étapes 4 à 6 afin d'ajouter une autre cible pour cette règle.
8. Sélectionnez Configure details. Dans Rule definition, saisissez un nom et une description pour la règle.

   Le nom de la règle doit être unique au sein de cette région.
9. Choisissez Create rule.

Création d'une règle CloudWatch Events qui se déclenche selon une planification

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Utilisez les étapes suivantes pour créer une règle CloudWatch Events qui se déclenche selon une planification régulière.

Pour créer une règle qui se déclenche selon une planification régulière

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Pour Event source, choisissez Schedule.
4. Choisissez Fixed rate of et spécifiez la fréquence à laquelle la tâche doit s'exécuter, ou choisissez Cron expression et indiquez une expression cron qui définit quand la tâche doit être déclenchée. Pour en savoir plus sur la syntaxe des expressions cron, consultez Expression de planification des règles (p. 34).
5. Pour Cibles, sélectionnez Add Target (Ajouter une cible) et choisissez le service AWS qui doit agir lorsqu'un événement du type sélectionné est détecté.
6. Dans les autres champs de cette section, entrez des informations spécifiques à ce type de cible, le cas échéant.

7. Pour de nombreux types de cibles, CloudWatch Events a besoin d'autorisations pour envoyer des événements à la cible. Dans ce cas, CloudWatch Events peut créer le rôle IAM nécessaire à l'exécution de votre événement :

   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant.
8. Le cas échéant, répétez les étapes 5 à 7 afin d'ajouter une autre cible pour cette règle.
9. Sélectionnez Configure details. Dans Rule definition, saisissez un nom et une description pour la règle.

   Le nom de la règle doit être unique au sein de cette région.
10. Choisissez Create rule.

Suppression ou désactivation d'une règle CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Procédez comme suit pour supprimer ou désactiver une règle CloudWatch Events.

Pour supprimer ou désactiver une règle

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Rules.

   Les règles gérées sont dotées d'une icône représentant une boîte en regard de leur nom. Pour de plus amples informations, veuillez consulter Règles gérées par Amazon CloudWatch Events (p. 100).
3. Effectuez l'une des actions suivantes :

   a. Pour supprimer une règle, sélectionnez le bouton en regard de la règle et choisissez Actions, Delete, Delete.

      Si la règle est une règle gérée, vous devrez saisir le nom de la règle pour confirmer qu'il s'agit d'une règle gérée, et que sa suppression peut arrêter la fonctionnalité dans le service l'ayant créée. Pour continuer, tapez le nom de règle et choisissez Forcer la suppression.

   b. Pour désactiver temporairement une règle, sélectionnez le bouton en regard de la règle et choisissez Actions, Disable, Disable.

      Vous ne pouvez pas désactiver une règle gérée.

Didacticiels CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Les didacticiels suivants vous montrent comment créer des règles CloudWatch Events pour certaines tâches et cibles.

Didacticiels:

Didacticiel : Utiliser CloudWatch Events pour relayer des événements vers la fonctionnalité Run Command de AWS Systems Manager

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez utiliser Amazon CloudWatch Events pour appeler la fonctionnalité Run Command de AWS Systems Manager et effectuer des actions sur les instances Amazon EC2 lorsque certains événements se produisent. Dans ce didacticiel, vous allez configurer la fonctionnalité Run Command pour exécuter les commandes du shell et configurer chaque nouvelle instance lancée dans un groupe Amazon EC2 Auto Scaling. Ce didacticiel suppose que vous avez déjà attribué une balise au groupe Amazon EC2 Auto Scaling avec `environment` comme clé et `production` comme valeur.

Pour créer la règle CloudWatch Events

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Choisissez Event Pattern, Build event pattern to match events by service.
   b. Dans Nom du service, choisissez Auto Scaling. Dans Type d'événement, choisissez Instance Launch and Terminate.
   c. Choisissez Specific instance event(s), puis EC2 Instance-launch Lifecycle Action.
   d. Par défaut, la règle correspond à tout groupe Amazon EC2 Auto Scaling dans la région. Pour que la règle corresponde à un groupe spécifique, choisissez Specific group name(s), puis sélectionnez un ou plusieurs groupes.
4. Dans Cibles, choisissez Add Target, SSM Run Command.
5. Pour Document, choisissez AWS-RunShellScript (Linux). Notez que de nombreuses autres options Document couvrent les instances Linux et Windows. Pour Target key, tapez **`tag:environment`**. Dans Target value(s), tapez **`production`** puis choisissez Add.
6. Sous Configure parameter(s), choisissez Constant.
7. Dans Commands, tapez une commande de shell puis choisissez Add. Répétez cette étape pour toutes les commandes à exécuter au lancement d'une instance.
8. Le cas échéant, saisissez les informations pertinentes dans WorkingDirectory et ExecutionTimeout.
9. CloudWatch Events peut créer le rôle IAM nécessaire à l'exécution de votre événement :
   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant.
10. Sélectionnez Configure details. Dans Rule definition, saisissez un nom et une description pour la règle.
11. Choisissez Create rule.

Didacticiel : Consigner l'état d'une instance Amazon Amazon EC2 à l'aide de CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez créer une fonction AWS Lambda qui enregistre les changements d'état d'une instance Amazon EC2. Vous pouvez décider de créer une règle qui exécute la fonction à chaque changement d'état

ou lorsqu'un ou plusieurs statuts spécifiques sont activés. Dans ce didacticiel, vous consignez le lancement d'une nouvelle instance.

Étape 1 : Création d'une fonction AWS Lambda

Créez une fonction Lambda pour enregistrer les événements de changement d'état. Vous spécifiez cette fonction lors de la création de votre règle.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Si vous utilisez Lambda pour la première fois, une page de bienvenue s'affiche. Sélectionnez Pour commencer. Sinon, choisissez Create a Lambda Function.
3. Sur la page Select blueprint, tapez `hello` comme filtre, puis choisissez le plan hello-world.
4. Sur la page Configure triggers, choisissez Next.
5. Sur la page Configure function, procédez comme suit :
   a. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction « LogEC2InstanceStateChange ».
   b. Modifiez l'exemple de code pour la fonction Lambda. Exemples :

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2InstanceStateChange');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. Pour Role (Rôle), choisissez Choose an existing role (Sélectionner un rôle existant). Pour Existing role (Rôle existant), sélectionnez votre rôle d'exécution de base. Sinon, créez un rôle d'exécution de base.
   d. Choisissez Suivant.
6. Sur la page Review, choisissez Create function.

Étape 2 : Création d'une règle

Créez une règle pour exécuter votre fonction Lambda chaque fois que vous lancez une instance Amazon EC2.

Pour créer une règle CloudWatch Events

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Choisissez Modèle d'événement.
   b. Choisissez Build event pattern to match events by service.
   c. Choisissez EC2, Notification de changement d'état d'instance.
   d. Choisissez Specific state(s), puis Running.
   e. Par défaut, la règle correspond à toute instance dans la région. Pour que la règle corresponde à une instance spécifique, choisissez Specific instance(s), puis choisissez une ou plusieurs instances.

4. Pour Targets (Cibles), sélectionnez Add target (Ajouter une cible), Function Lambda (Fonction Lambda).
5. DansFunction, sélectionnez la fonction Lambda que vous avez créée.
6. Sélectionnez Configure details.
7. Dans Rule definition, saisissez un nom et une description pour la règle.
8. Choisissez Create rule.

Étape 3 : Test de la règle

Pour tester la règle, lancez une instance Amazon EC2. Après avoir attendu quelques minutes que l'instance soit lancée et initialisée, vous pouvez vérifier que votre fonction Lambda a été appelée.

Pour tester votre règle en lançant une instance

1. Ouvrez la console Amazon EC2 sur https://console.aws.amazon.com/ec2/.
2. Lancez une instance. Pour de plus amples informations, veuillez consulter Lancer votre instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
3. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
4. Dans le volet de navigation, choisissez Événements, Règles, sélectionnez le nom de la règle que vous avez créée, puis choisissez Show metrics for the rule.
5. Pour afficher la sortie de la fonction Lambda, procédez de la manière suivante :
   a. Dans le volet de navigation, sélectionnez Logs.
   b. Cliquez sur le nom du groupe de journaux pour votre fonction Lambda (/aws/lambda/function-name).
   c. Choisissez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez lancée.
6. (Facultatif) Lorsque vous avez terminé, vous pouvez ouvrir la console Amazon EC2 et arrêter ou mettre hors service l'instance que vous avez lancée. Pour de plus amples informations, veuillez consulter Terminer votre instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Didacticiel : Enregistrer l'état d'un groupe Auto Scaling à l'aide de CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez exécuter une fonction AWS Lambda qui enregistre un événement chaque fois qu'un groupe Auto Scaling lance ou met hors service une instance Amazon EC2, en fonction de la réussite de cet événement.

Pour plus d'informations sur d'autres scénarios CloudWatch Events utilisant les événements Amazon EC2 Auto Scaling, consultez Utiliser CloudWatch Events lorsque votre groupe Auto Scaling évolue dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Étape 1 : Création d'une fonction AWS Lambda

Créez une fonction Lambda pour enregistrer les événements de montée et de diminution en charge de votre groupe Auto Scaling. Vous spécifiez cette fonction lors de la création de votre règle.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Si vous utilisez Lambda pour la première fois, une page de bienvenue s'affiche. Sélectionnez Pour commencer. Sinon, choisissez Create a Lambda Function.
3. Sur la page Select blueprint, tapez `hello` comme filtre, puis choisissez le plan hello-world.
4. Sur la page Configure triggers, choisissez Next.
5. Sur la page Configure function, procédez comme suit :
   a. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction « LogAutoScalingEvent ».
   b. Modifiez l'exemple de code pour la fonction Lambda. Exemples :

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogAutoScalingEvent');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. Pour Role (Rôle), choisissez Choose an existing role (Sélectionner un rôle existant). Pour Existing role (Rôle existant), sélectionnez votre rôle d'exécution de base. Sinon, créez un rôle d'exécution de base.
   d. Choisissez Suivant.
6. Sélectionnez Créer une fonction.

Étape 2 : Création d'une règle

Créez une règle pour exécuter votre fonction Lambda chaque fois que votre groupe Auto Scaling lance ou met hors service une instance.

Pour créer une règle

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Choisissez Modèle d'événement.
   b. Choisissez Build event pattern to match events by service.
   c. Sélectionnez Auto Scaling, puis Launch and Terminate (Lancement et résiliation d'instances).
   d. Pour capturer tous les événements de lancement et de mise hors service d'instances réussis ou non, choisissez Any instance event.
4. Par défaut, la règle correspond à tout groupe Auto Scaling de la région. Pour que la règle corresponde à un groupe spécifique, sélectionnez Specific group name(s) (Nom[s] de groupe[s] spécifique[s]), puis sélectionnez un ou plusieurs groupes Auto Scaling.
5. Pour Targets (Cibles), sélectionnez Add target (Ajouter une cible), Function Lambda (Fonction Lambda).

6. DansFunction, sélectionnez la fonction Lambda que vous avez créée.
7. Sélectionnez Configure details.
8. Dans Rule definition, saisissez un nom et une description pour la règle. Par exemple, décrivez la règle de la manière suivante : « Log whenever an Auto Scaling group scales out or in » (Enregistrer chaque fois que la charge d'un groupe Auto Scaling est montée en puissance ou mise à l'échelle horizontalement).
9. Choisissez Create rule.

Étape 3 : Test de la règle

Vous pouvez tester votre règle en dimensionnant manuellement un groupe Auto Scaling de sorte qu'il lance une instance. Après avoir attendu quelques minutes que l'événement de montée en charge se produise, vous pouvez vérifier que votre fonction Lambda a été appelée.

Pour tester votre règle avec un groupe Auto Scaling

1. Pour augmenter la taille du groupe Auto Scaling, procédez de la manière suivante :
   a. Ouvrez la console Amazon EC2 sur https://console.aws.amazon.com/ec2/.
   b. Dans le volet de navigation, choisissez Auto Scaling, puis Groupes Auto Scaling.
   c. Cochez la case correspondant à votre groupe Auto Scaling.
   d. Dans l'onglet Details (Détails), choisissez Edit (Modifier). Pour Desired, augmentez la capacité souhaitez d'un. Par exemple, si la valeur actuelle est 2, saisissez 3. La capacité souhaitée doit être inférieure ou égale à la taille maximum du groupe. Si la nouvelle valeur pour Desired est supérieure à Max, vous devez mettre à jour Max. Lorsque vous avez terminé, choisissez Save.
2. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
3. Dans le volet de navigation, choisissez Événements, Règles, sélectionnez le nom de la règle que vous avez créée, puis choisissez Show metrics for the rule.
4. Pour afficher la sortie de la fonction Lambda, procédez de la manière suivante :
   a. Dans le volet de navigation, sélectionnez Logs.
   b. Sélectionnez le nom du groupe de journaux pour la fonction Lambda (/aws/lambda/function-name).
   c. Sélectionnez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez lancée.
5. (Facultatif) Lorsque vous avez terminé, vous pouvez réduire la capacité souhaitée d'une unité, de sorte que le groupe Auto Scaling reprenne sa taille antérieure.

Didacticiel : Consigner des opérations au niveau des objets Amazon S3 avec CloudWatch

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez enregistrer les opérations d'API au niveau de l'objet dans vos compartiments S3. Pour qu'Amazon CloudWatch Events puisse apparier ces événements, vous devez utiliser AWS CloudTrail pour configurer un journal d'activité destiné à recevoir ces événements.

Étape 1 : Configuration de votre journal de suivi AWS CloudTrail

Pour enregistrer des événements de données correspondant à un compartiment S3 dans AWS CloudTrail et CloudWatch Events, créez un journal de suivi. Un journal de suivi capture les appels d'API et les événements liés dans votre compte et transfère les fichiers journaux dans un compartiment S3 que vous spécifiez. Vous pouvez mettre à jour un journal de suivi existant ou en créer une nouveau.

Pour créer un journal d'activité

1. Ouvrez la console CloudTrail à l'adresse https://console.aws.amazon.com/cloudtrail/.
2. Dans le volet de navigation, choisissez Trails, Create trail.
3. Dans Trail name, tapez un nom pour le journal.
4. Pour Data events, tapez le nom du compartiment et le préfixe (facultatif). Pour chaque journal de suivi, vous pouvez ajouter jusqu'à 250 objets Amazon S3.
   - Pour enregistrer des événements de données pour tous les objets Amazon S3 d'un compartiment, précisez un compartiment S3 et un préfixe vide. Lorsqu'un événement se produit sur un objet de ce compartiment , celui-ci est traité et enregistré par le suivi.
   - Pour consigner des événements de données pour des objets Amazon S3 spécifiques, sélectionnez Add S3 bucket (Ajouter un compartiment S3), puis spécifiez un compartiment S3 et, le cas échéant, le préfixe de l'objet. Lorsqu'un événement se produit sur un objet de ce compartiment et que l'objet commence par le préfixe spécifié, le suivi traite et consigne l'événement.
5. Pour chaque ressource, spécifiez si vous souhaitez consigner les événements Read, Write ou les deux types d'événements.
6. Pour Storage location, créez ou choisissez un compartiment S3 existant à désigner pour le stockage de fichiers journaux.
7. Sélectionnez Créer.

Pour plus d'informations, consultez Événements de données dans le Guide de l'utilisation AWS CloudTrail.

Étape 2 : Création d'une fonction AWS Lambda

Créez une fonction Lambda pour enregistrer les événements de données de vos compartiments S3. Vous spécifiez cette fonction lors de la création de votre règle.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Si vous utilisez Lambda pour la première fois, une page de bienvenue s'affiche. Choisissez Créer une fonction. Sinon, choisissez Créer la fonction.
3. Choisissez Créer à partir de zéro.
4. Sous Author from scratch, effectuez les étapes suivantes :
   a. Attribuez un nom à la fonction Lambda. Par exemple, nommez la fonction « LogS3DataEvents ».
   b. Dans le champ Rôle, choisissez Créer un rôle personnalisé.

Une nouvelle fenêtre s'ouvre. Changez le contenu de Nom du rôle si nécessaire et choisissez Allow (Autoriser).

c. Dans la console Lambda, sélectionnez Create function (Créer une fonction).

5. Remplacez le code pour la fonction Lambda par ce qui suit, puis choisissez Enregistrer.

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogS3DataEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

Étape 3 : Création d'une règle

Créez une règle pour exécuter votre fonction Lambda en réponse à un événement de données Amazon S3.

Pour créer une règle

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Rules (Règles), Create rule (Créer une règle).
3. Dans Event source, effectuez l'une des opérations suivantes :

   a. Choisissez Modèle d'événement.
   b. Choisissez Build event pattern to match events by service.
   c. Choisissez Simple Storage Service (S3), puis Object Level Operations.
   d. Choisissez Specific operation(s) (Opération(s) spécifique(s)), puis PutObject.
   e. Par défaut, la règle correspond aux événements de données pour tous les compartiments dans la région. Pour faire correspondre des événements de données pour des compartiments spécifiques, choisissez Specify bucket(s) by name, puis précisez un ou plusieurs compartiments.
4. Pour Targets (Cibles), sélectionnez Add target (Ajouter une cible), Function Lambda (Fonction Lambda).
5. DansFunction, sélectionnez la fonction Lambda que vous avez créée.
6. Sélectionnez Configure details.
7. Dans Rule definition, saisissez un nom et une description pour la règle.
8. Choisissez Create rule.

Étape 4 : Test de la règle

Pour tester la règle, placez un objet dans votre compartiment S3. Vous pouvez vérifier que votre fonction Lambda a été appelée.

Pour afficher les journaux de votre fonction Lambda

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Logs.
3. Sélectionnez le nom du groupe de journaux pour la fonction Lambda (/aws/lambda/function-name).
4. Sélectionnez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez lancée.

Vous pouvez également vérifier le contenu de vos fichiers journaux CloudTrail dans le compartiment S3 que vous avez précisé pour votre journal d'activité. Pour plus d'informations, consultez Obtention et consultation des fichiers journaux CloudTrail dans le AWS CloudTrail Guide de l'utilisateur.

Didacticiel : Utilisation d'un transformateur d'entrée pour personnaliser les éléments transmis à la cible d'événement

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez utiliser la fonction de transformation d'entrée de CloudWatch Events pour personnaliser le texte d'un événement avant qu'il ne soit entré dans la cible d'une règle.

Vous pouvez définir plusieurs chemins JSON à partir de l'événement et affecter leurs sorties à différentes variables. Vous pouvez ensuite utiliser ces variables dans le modèle d'entrée au format <*variable-name*>. Les caractères < et > ne peuvent pas être placés dans une séquence d'échappement.

Si vous spécifiez une variable à mettre en relation avec un chemin JSON qui n'existe pas dans l'événement, la variable n'est pas créée et n'apparaît pas dans la sortie.

Dans ce didacticiel, nous extrayons l'ID d'instance et l'état d'une instance Amazon EC2 de l'événement de modification de l'état d'instance. Nous utilisons le transformateur d'entrée pour placer ces données dans un message facile à lire envoyé à une rubrique Amazon SNS. La règle est déclenchée lorsqu'une instance change d'état. Par exemple, avec cette règle, l'événement de notification du changement d'état de l'instance Amazon EC2 suivant produit le message Amazon SNS The EC2 instance i-1234567890abcdef0 has changed state to stopped (L'instance EC2 i-1234567890abcdef0 est passée à l'état Arrêtée).

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2015-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/ i-1234567890abcdef0"
   ],
   "detail":{
      "instance-id":" i-1234567890abcdef0",
      "state":"stopped"
   }
}
```

Pour ce faire, nous relions la variable *instance* au chemin JSON `$.detail.instance-id` de l'événement, et la variable *state* au chemin JSON `$.detail.state`. Nous définissons ensuite le modèle d'entrée sur « L'instance EC2 <instance> est passée à l'état state>. »

Créer une règle

Pour utiliser le transformateur d'entrée et personnaliser les informations de changement d'état de l'instance envoyées à une cible

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Choisissez Modèle d'événement.
   b. Choisissez Build event pattern to match events by service.
   c. Choisissez EC2, Notification de changement d'état d'instance.
   d. Sélectionnez Tout état et Toute instance.
4. Dans Cibles, choisissez Ajouter une cible, Rubrique SNS.
5. Dans Topic (Rubrique), sélectionnez la rubrique Amazon SNS pour laquelle vous souhaitez recevoir une notification en cas de changement d'état des instances Amazon EC2.
6. Choisissez Configurer l'entrée, puis Transformateur d'entrée.
7. Dans la zone suivante, saisissez {« state » : « $.detail.state », « instance » : « $.detail.instance-id »}
8. Dans la zone suivante, saisissez « L'instance EC2 <instance> est passée à l'état state <state>."
9. Sélectionnez Configure details.
10. Tapez un nom et une description pour la règle, puis choisissez Create rule.

Didacticiel : Enregistrer des appels d'API AWS avec CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez utiliser une fonction AWS Lambda qui enregistre chaque appel d'API AWS. Par exemple, vous pouvez créer une règle pour consigner chaque opération dans Amazon EC2, ou vous pouvez limiter cette règle pour consigner uniquement un appel d'API spécifique. Dans ce didacticiel, une consignation a lieu chaque fois qu'une instance Amazon EC2 est arrêtée.

Prerequisite

Pour pouvoir apparier ces événements, vous devez utiliser AWS CloudTrailpour configurer un journal d'activité. Si vous ne possédez pas de journal d'activité, utilisez la procédure suivante.

Pour créer un journal d'activité

1. Ouvrez la console CloudTrail à l'adresse https://console.aws.amazon.com/cloudtrail/.
2. Choisissez Trails (Suivis), Create trail (Créer un suivi).
3. Dans Trail name, tapez un nom pour le journal.

4. Pour Storage location (Emplacement de stockage), dans Create a new S3 bucket (Créer un compartiment S3), saisissez le nom du nouveau compartiment auquel CloudTrail doit remettre les journaux.
5. Sélectionnez Créer.

Étape 1 : Création d'une fonction AWS Lambda

Créez une fonction Lambda pour enregistrer les événements d'appels d'API. Vous spécifiez cette fonction lors de la création de votre règle.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Si vous utilisez Lambda pour la première fois, une page de bienvenue s'affiche. Sélectionnez Pour commencer. Sinon, choisissez Create a Lambda Function.
3. Sur la page Select blueprint, tapez `hello` comme filtre, puis choisissez le plan hello-world.
4. Sur la page Configure triggers, choisissez Next.
5. Sur la page Configure function, procédez comme suit :

   a. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction « LogEC2StopInstance ».

   b. Modifiez l'exemple de code pour la fonction Lambda. Exemples :

```
'use strict';

exports.handler = (event, context, callback) => {
    console.log('LogEC2StopInstance');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

   c. Pour Role (Rôle), choisissez Choose an existing role (Sélectionner un rôle existant). Pour Existing role (Rôle existant), sélectionnez votre rôle d'exécution de base. Sinon, créez un rôle d'exécution de base.

   d. Choisissez Suivant.
6. Sur la page Review, choisissez Create function.

Étape 2 : Création d'une règle

Créez une règle pour exécuter votre fonction Lambda chaque fois que vous arrêtez une instance Amazon EC2.

Pour créer une règle

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :

   a. Choisissez Modèle d'événement.

   b. Choisissez Build event pattern to match events by service.

   c. Sélectionnez EC2, Appel d'API AWS via CloudTrail.

   d. Choisissez Specific operation(s), puis tapez `StopInstances` dans la zone ci-dessous.

4. Pour Targets (Cibles), sélectionnez Add target (Ajouter une cible), Function Lambda (Fonction Lambda).
5. DansFunction, sélectionnez la fonction Lambda que vous avez créée.
6. Sélectionnez Configure details.
7. Dans Rule definition, saisissez un nom et une description pour la règle.
8. Choisissez Create rule.

Étape 3 : Test de la règle

Vous pouvez tester votre règle en arrêtant une instance Amazon EC2 à l'aide de la console Amazon EC2. Après avoir attendu quelques minutes que l'instance soit arrêtée, vérifiez vos métriques AWS Lambda dans la console CloudWatch afin de vous assurer que votre fonction a été appelée.

Pour tester votre règle en arrêtant une instance

1. Ouvrez la console Amazon EC2 sur https://console.aws.amazon.com/ec2/.
2. Lancez une instance. Pour de plus amples informations, veuillez consulter Lancer votre instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
3. Arrêtez l'instance. Pour plus d'informations, veuillez consulter Arrêter et démarrage de votre instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
4. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
5. Dans le volet de navigation, choisissez Événements, sélectionnez le nom de la règle que vous avez créée, puis choisissez Show metrics for the rule.
6. Pour afficher la sortie de la fonction Lambda, procédez de la manière suivante :
   a. Dans le volet de navigation, sélectionnez Logs.
   b. Sélectionnez le nom du groupe de journaux pour la fonction Lambda (/aws/lambda/function-name).
   c. Sélectionnez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez arrêtée.
7. (Facultatif) Lorsque vous avez terminé, vous pouvez mettre hors service l'instance arrêtée. Pour de plus amples informations, veuillez consulter Terminer votre instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Didacticiel : Planifier des instantanés Amazon EBS automatisés avec CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez exécuter des règles CloudWatch Events selon une planification. Dans ce didacticiel, vous allez créer un instantané automatisé d'un volume Amazon Elastic Block Store (Amazon EBS) existant selon

une planification. Vous pouvez choisir une fréquence fixe pour créer un instantané toutes les quelques minutes, ou utiliser une expression cron pour spécifier la prise de l'instantané à un moment précis de la journée.

> Important
>
> La création de règles avec des cibles intégrées est uniquement prise en charge dans la AWS Management Console.

Étape 1 : Création d'une règle

Créez une règle qui prend des instantanés sur un calendrier. Vous pouvez utiliser une expression de fréquence ou une expression cron pour préciser le calendrier. Pour de plus amples informations, veuillez consulter Expression de planification des règles (p. 34).

Pour créer une règle

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Sélectionnez Programme.
   b. Choisissez Fixed rate of et précisez l'intervalle de calendrier (par exemple, 5 minutes). Vous pouvez également choisir Expression CRON et préciser une expression cron (par exemple, toutes les 15 minutes du lundi au vendredi, à partir de l'heure actuelle).
4. Pour Targets (Cibles), choisissez Add target (Ajouter une cible), puis sélectionnez Appel d'API CreateSnapshot EC2. Vous devrez peut-être faire défiler la liste des cibles possibles vers le haut pour trouver Appel d'API CreateSnapshot EC2.
5. Pour Volume ID (ID du volume), saisissez l'ID du volume Amazon EBS ciblé.
6. Choisissez Create a new role for this specific resource. Le nouveau rôle accorde à la cible des autorisations pour accéder aux ressources en votre nom.
7. Sélectionnez Configure details.
8. Dans Rule definition, saisissez un nom et une description pour la règle.
9. Choisissez Create rule.

Étape 2 : Test de la règle

Vous pouvez vérifier votre règle en affichant votre premier instantané après l'avoir pris.

Pour tester la règle

1. Ouvrez la console Amazon EC2 sur https://console.aws.amazon.com/ec2/.
2. Dans le panneau de navigation, choisissez Elastic Block Store, Snapshots.
3. Vérifiez que le premier instantané apparaît dans la liste.
4. (Facultatif) Lorsque vous avez terminé, vous pouvez désactiver la règle pour empêcher la prise d'autres instantanés.
   a. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
   b. Dans le volet de navigation, choisissez Events, Rules.
   c. Sélectionnez la règle, puis choisissez Actions, Désactiver.
   d. Lorsque vous êtes invité à confirmer l'opération, choisissez Désactiver.

Didacticiel : Planifier des fonctions AWS Lambda avec CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez configurer une règle permettant d'exécuter une fonction AWS Lambda selon un calendrier. Ce didacticiel explique comment utiliser AWS Management Console ou l'AWS CLI pour créer la règle. Si vous souhaitez utiliser la AWS CLI, mais que celle-ci n'est pas installée, consultez le Guide de l'utilisateur AWS Command Line Interface.

CloudWatch Events ne fournit pas de précision de deuxième niveau dans les expressions de planification. Le niveau de résolution maximum lors de l'utilisation d'une expression cron est d'une minute. En raison de la nature distribuée de CloudWatch Events et des services cible, le délai entre le moment où la règle planifiée est déclenchée et celui où le service cible lance l'exécution de la ressource cible peut être de plusieurs secondes. Votre règle planifiée est déclenchée au cours de cette minute, mais pas précisément à la seconde exacte.

Étape 1 : Création d'une fonction AWS Lambda

Créez une fonction Lambda pour enregistrer les événements planifiés. Vous spécifiez cette fonction lors de la création de votre règle.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Si vous utilisez Lambda pour la première fois, une page de bienvenue s'affiche. Sélectionnez Pour commencer. Sinon, choisissez Create a Lambda Function.
3. Sur la page Select blueprint, tapez `hello` comme filtre, puis choisissez le plan hello-world.
4. Sur la page Configure triggers, choisissez Next.
5. Sur la page Configure function, procédez comme suit :

   a. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction « LogScheduledEvent ».

   b. Modifiez l'exemple de code pour la fonction Lambda. Exemples :

   ```
   'use strict';

   exports.handler = (event, context, callback) => {
       console.log('LogScheduledEvent');
       console.log('Received event:', JSON.stringify(event, null, 2));
       callback(null, 'Finished');
   };
   ```

   c. Pour Role (Rôle), choisissez Choose an existing role (Sélectionner un rôle existant). Pour Existing role (Rôle existant), sélectionnez votre rôle d'exécution de base. Sinon, créez un rôle d'exécution de base.

   d. Choisissez Suivant.

6. Sur la page Review, choisissez Create function.

Étape 2 : Création d'une règle

Créez une règle pour exécuter votre fonction Lambda selon une planification.

Pour créer une règle avec la console

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Sélectionnez Programme.
   b. Choisissez Fixed rate of et précisez l'intervalle de calendrier (par exemple, 5 minutes).
4. Pour Targets (Cibles), sélectionnez Add target (Ajouter une cible), Function Lambda (Fonction Lambda).
5. DansFunction, sélectionnez la fonction Lambda que vous avez créée.
6. Sélectionnez Configure details.
7. Dans Rule definition, saisissez un nom et une description pour la règle.
8. Choisissez Create rule.

Si vous préférez, vous pouvez créer la règle avec l' AWS CLI. Vous devez d'abord autoriser la règle à appeler votre fonction Lambda. Vous pouvez ensuite créer la règle et ajouter la fonction Lambda comme cible.

Pour créer une règle avec l' AWS CLI

1. Utilisez la commande put-rule suivante pour créer une règle qui se déclenche automatiquement selon un calendrier :

```
aws events put-rule \
--name my-scheduled-rule \
--schedule-expression 'rate(5 minutes)'
```

   Lorsque cette règle se déclenche, elle génère un événement qui sert d'entrée pour les cibles de cette règle. Voici un exemple d'événement :

```
{
    "version": "0",
    "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
    "detail-type": "Scheduled Event",
    "source": "aws.events",
    "account": "123456789012",
    "time": "2015-10-08T16:53:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule"
    ],
    "detail": {}
}
```

2. Utilisez la commande add-permission suivante pour approuver le mandataire du service CloudWatch Events (events.amazonaws.com) et définir les autorisations selon la règle avec l'Amazon Resource Name (ARN) spécifié :

```
aws lambda add-permission \
--function-name LogScheduledEvent \
--statement-id my-scheduled-event \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/my-scheduled-rule
```

3. Utilisez la commande put-targets suivante pour ajouter la fonction Lambda que vous avez créée à cette règle de telle sorte qu'elle s'exécute toutes les cinq minutes :

```
aws events put-targets --rule my-scheduled-rule --targets file://targets.json
```

Créez le fichier `targets.json` contenant les éléments suivants :

```
[
  {
    "Id": "1",
    "Arn": "arn:aws:lambda:us-east-1:123456789012:function:LogScheduledEvent"
  }
]
```

Étape 3 : Vérification de la règle

Au moins cinq minutes après avoir effectué l'étape 2, vous pouvez vérifier que votre fonction Lambda a été appelée.

Pour tester la règle

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Événements, Règles, sélectionnez le nom de la règle que vous avez créée, puis choisissez Show metrics for the rule.
3. Pour afficher la sortie de la fonction Lambda, procédez de la manière suivante :
   a. Dans le volet de navigation, sélectionnez Logs.
   b. Sélectionnez le nom du groupe de journaux pour la fonction Lambda (/aws/lambda/function-name).
   c. Sélectionnez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez lancée.
4. (Facultatif) Lorsque vous avez terminé, vous pouvez désactiver la règle.
   a. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
   b. Dans le volet de navigation, choisissez Events, Rules.
   c. Sélectionnez la règle, puis choisissez Actions, Désactiver.
   d. Lorsque vous êtes invité à confirmer l'opération, choisissez Désactiver.

Didacticiel : Définir AWS Systems Manager Automation en tant que cible CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez utiliser CloudWatch Events pour appeler AWS Systems Manager Automation, selon une planification régulière ou lorsque des événements spécifiques sont détectés. Ce didacticiel part du principe que vous appelez Systems Manager Automation sur la base de certains événements.

Pour créer la règle CloudWatch Events

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :

   a. Choisissez Modèle d'événement, puis Créer un modèle d'événement correspondant aux événements par service.

   b. Pour Nom du service et Type d'événement, choisissez le service et le type d'événement à utiliser comme déclencheur.

      En fonction du service et du type d'événement que vous choisissez, vous devrez peut-être spécifier des options supplémentaires dans la Source de l'événement.
4. Dans Cibles, choisissez Ajouter une cible, SSM Automation.
5. Pour Document, choisissez le document Systems Manager à exécuter lorsque la cible est déclenchée.
6. (Facultatif) Pour spécifier une certaine version du document, choisissez Configure document version.
7. Sous Configurer le(s) paramètre(s), choisissez Aucun paramètre ou Constant.

   Si vous choisissez Constant, spécifiez les constantes que vous souhaitez transmettre à l'exécution du document.
8. CloudWatch Events peut créer le rôle IAM nécessaire à l'exécution de votre événement :

   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant.
9. Sélectionnez Configure details. Dans Rule definition, saisissez un nom et une description pour la règle.
10. Choisissez Create rule.

Didacticiel : Relayer des événements vers un flux Amazon Kinesis avec CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez relayer des événements d'appels d'API AWS dans CloudWatch Events vers un flux dans Amazon Kinesis.

Prerequisite

Installez la AWS CLI. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS Command Line Interface.

Étape 1 : créer un flux Amazon Kinesis

Utilisez la commande create-stream suivante pour créer un flux.

```
aws kinesis create-stream --stream-name test --shard-count 1
```

Lorsque le statut du flux est `ACTIVE`, le flux est prêt. Utilisez la commande describe-stream suivante pour vérifier le statut du flux :

```
aws kinesis describe-stream --stream-name test
```

Étape 2 : Création d'une règle

Par exemple, créez une règle pour envoyer des événements à votre flux lorsque vous arrêtez une instance Amazon EC2.

Pour créer une règle

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Choisissez Modèle d'événement.
   b. Choisissez Build event pattern to match events by service.
   c. Choisissez EC2, notification de changement d'état d'instance.
   d. Choisissez Specific state(s), puis Running.
4. Dans Targets, choisissez Add target, puis Flux Kinesis.
5. Dans le champ Flux, sélectionnez le flux que vous avez créé.

6. Choisissez Create a new role for this specific resource.
7. Sélectionnez Configure details.
8. Dans Rule definition, saisissez un nom et une description pour la règle.
9. Choisissez Create rule.

Étape 3 : Test de la règle

Pour tester la règle, arrêtez une instance Amazon EC2. Après avoir attendu quelques minutes que l'instance soit arrêtée, vérifiez vos métriques CloudWatch afin de vous assurer que votre fonction a été appelée.

Pour tester votre règle en arrêtant une instance

1. Ouvrez la console Amazon EC2 sur https://console.aws.amazon.com/ec2/.
2. Lancez une instance. Pour de plus amples informations, veuillez consulter Lancer votre instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.
3. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
4. Dans le volet de navigation, choisissez Événements, Règles, sélectionnez le nom de la règle que vous avez créée, puis choisissez Show metrics for the rule.
5. (Facultatif) Lorsque vous avez terminé, vous pouvez mettre hors service l'instance. Pour de plus amples informations, veuillez consulter Terminer votre instance dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Etape 4 : Vérification que l'événement est relayé

Vous pouvez obtenir l'enregistrement auprès du flux afin de vérifier que l'événement a été relayé.

Pour obtenir l'enregistrement

1. Utilisez la commande get-shard-iterator suivante pour commencer la lecture à partir de votre flux Kinesis :

```
aws kinesis get-shard-iterator --shard-id shardId-000000000000 --shard-iterator-type
 TRIM_HORIZON --stream-name test
```

   Voici un exemple de sortie :

```
{
    "ShardIterator": "AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp
+KEd9I6AJ9ZG4lNR1EMi+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd
+efGN2aHFdkH1rJl4BL9Wyrk+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg="
}
```

2. Utilisez la commande get-records suivante pour obtenir l'enregistrement. L'itérateur de partition est celui que vous avez obtenu à l'étape précédente :

```
aws kinesis get-records --shard-
iterator AAAAAAAAAAHSywljv0zEgPX4NyKdZ5wryMzP9yALs8NeKbUjp1IxtZs1Sp+KEd9I6AJ9ZG4lNR1EMi
+9Md/nHvtLyxpfhEzYvkTZ4D9DQVz/mBYWRO6OTZRKnW9gd+efGN2aHFdkH1rJl4BL9Wyrk
+ghYG22D2T1Da2EyNSH1+LAbK33gQweTJADBdyMwlo5r6PqcP2dzhg=
```

   Si la commande aboutit, elle demande des enregistrements de votre flux correspondant à la partition spécifiée. Vous pouvez recevoir zéro ou plusieurs enregistrements. Les enregistrements renvoyés

peuvent ne pas représenter tous les enregistrements de votre flux. Si vous ne recevez pas les données attendues, continuez d'appeler `get-records`.

Les enregistrements dans Kinesis sont codés en Base64. Cependant, la prise en charge des flux dans le AWS CLI ne fournit pas de décodage en base64. Si vous utilisez un décodeur base64 pour décoder manuellement les données, vous voyez qu'il s'agit de l'événement relayé vers le flux sous forme JSON.

Didacticiel : Exécuter une tâche Amazon ECS quand un fichier est chargé sur un compartiment Amazon S3

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez utiliser CloudWatch Events pour exécuter des tâches Amazon ECS lorsque certains événements AWS se produisent. Dans ce didacticiel, vous configurez une règle CloudWatch Events qui exécute une tâche Amazon ECS chaque fois qu'un fichier est chargé sur un certain compartiment Amazon S3 à l'aide de l'opération PUT Amazon S3.

Le didacticiel part du principe que vous avez déjà créé la définition de la tâche dans Amazon ECS.

Pour exécuter une tâche Amazon ECS chaque fois qu'un fichier est chargé sur un compartiment S3 à l'aide de l'opération PUT.

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :
   a. Choisissez Modèle d'événement.
   b. Choisissez Build event pattern to match events by service.
   c. Pour Service Name, choisissez Simple Storage Service (S3).
   d. Dans la liste déroulante Event Type (Type d'événement), choisissez Object Level Operations (Opérations de niveau objet).
   e. Choisissez Specific operation(s) (Opération(s) spécifique(s)), puis PutObject.
   f. Choisissez Specific bucket(s) by name (Compartiments par nom), puis entrez le nom du compartiment.
4. Pour Targets (Cibles), procédez comme suit :
   a. Choisissez Add target (Ajouter une cible), ECS task (Tâche ECS).
   b. Pour Cluster et Task Definition (Définition de tâche), sélectionnez les ressources que vous avez créées.
   c. Pour Launch Type (Type de lancement), sélectionnez `FARGATE` ou `EC2`. `FARGATE` ne s'affiche que dans les régions où AWS Fargate est pris en surcharge.

d. (Facultatif) Spécifiez une valeur pour Groupe de tâches. Si Launch Type (Type de lancement) a pour valeur `FARGATE`, spécifiez le cas échéant la Platform Version (Version de la plateforme). Spécifiez uniquement la partie numérique de la version de la plateforme, telle que 1.1.0.

e. (Facultatif) Spécifiez une révision de définition de tâche et un nombre de tâches. Si vous ne spécifiez pas une révision de définition de tâche, c'est la plus récente qui est utilisée.

f. Si votre définition de tâche utilise le mode réseau awsvpc, vous devez spécifier les sous-réseaux et les groupes de sécurité. Tous les sous-réseaux et groupes de sécurité doivent se trouver dans le même VPC.

   Si vous spécifiez plusieurs groupes de sécurité ou sous-réseaux, séparez-les par des virgules, pas par des espaces.

   Pour Subnets (Sous-réseaux), spécifiez la valeur `subnet-id` complète pour chaque sous-réseau, comme dans l'exemple suivant :

   ```
   subnet-123abcd,subnet-789abcd
   ```

g. Choisissez si vous autorisez que l'adresse IP publique soit assignée automatiquement.

h. CloudWatch Events peut créer le rôle IAM nécessaire à l'exécution de votre tâche :

   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant. Il doit s'agir d'un rôle possédant déjà les autorisations nécessaires pour appeler la version. CloudWatch Events n'accorde pas d'autorisations supplémentaires au rôle que vous sélectionnez.

5. Sélectionnez Configure details.
6. Dans Rule definition, saisissez un nom et une description pour la règle.
7. Choisissez Create rule.

Didacticiel : Planifier des versions automatisées avec CodeBuild

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Dans l'exemple de ce didacticiel, vous planifiez CodeBuild pour exécuter une version à 20 h 00 GMT chaque soir de semaine. Vous pouvez également transmettre une constante à CodeBuild à utiliser pour cette version planifiée.

Pour créer une règle planifiant une version de projet CodeBuild chaque soir à 20 h 00

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Events, Create rule.
3. Dans Event source, effectuez l'une des opérations suivantes :

   a. Sélectionnez Programme.
   b. Sélectionnez Cron expression (Expression Cron) et spécifiez les éléments suivants comme l'expression : 0 20 ? * MON-FRI *. Pour plus d'informations sur les expressions cron, consultez Expression de planification des règles (p. 34).
4. Dans Targets (Cibles), choisissez Add target (Ajouter une cible), puis CodeBuild project (Projet CodeBuild).
5. Pour Project ARN, saisissez l'ARN du projet de version.
6. Dans ce didacticiel, nous avons ajouté l'étape facultative de transmission d'un paramètre à CodeBuild afin de remplacer la valeur par défaut. Cette étape n'est pas nécessaire lorsque vous définissez CodeBuild comme cible. Pour transmettre le paramètre, choisissez Configure input, Constant (JSON text).

   Dans la zone située sous Constant (JSON text), entrez ce qui suit pour définir le délai de remplacement sur 30 minutes pour les versions planifiées suivantes : { "timeoutInMinutesOverride": 30 }

   Pour plus d'informations sur les paramètres que vous pouvez transmettre, consultez StartBuild. Vous ne pouvez pas transmettre le paramètre `projectName` dans ce champ. Au lieu de cela, spécifiez le projet à l'aide de l'ARN de Project ARN.
7. CloudWatch Events peut créer le rôle IAM nécessaire à l'exécution de votre projet de version :

   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Pour utiliser un rôle IAM que vous avez créé auparavant, choisissez Utiliser le rôle existant. Il doit s'agir d'un rôle possédant déjà les autorisations nécessaires pour appeler la version. CloudWatch Events n'accorde pas d'autorisations supplémentaires au rôle que vous sélectionnez.
8. Sélectionnez Configure details
9. Dans Rule definition, saisissez un nom et une description pour la règle.
10. Choisissez Create rule.

Didacticiel : Enregistrer les changements d'état des instances Amazon EC2

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Dans l'exemple de ce didacticiel, vous allez créer une règle qui entraîne des notifications de modification dans Amazon EC2 à enregistrer dans CloudWatch Logs.

Pour créer une règle pour enregistrer les notifications de changement d'état Amazon EC2 dans CloudWatch Logs

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Events (Événements), puis Create rule (Créer une règle).
3. Dans Event source, effectuez l'une des opérations suivantes :

   a. Choisissez Modèle d'événement.
   b. Pour Service Name (Nom du service), choisissez EC2.
   c. Pour Event Type (Type d'événement), choisissez EC2 Instance State-change Notification (Notification de changement d'état de l'instance EC2).
4. Pour Targets, choisissez Add target. Dans la liste des services, choisissez le groupe de journaux CloudWatch.
5. Pour Log Group (Groupe de journaux), saisissez un nom pour que le groupe de journaux reçoive les notifications de changement d'état.
6. Sélectionnez Configure details.
7. Dans Rule definition (Définition de la règle), saisissez un nom et une description pour la règle.
8. Choisissez Create rule.

Expression de planification des règles

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez créer des règles qui se déclenchent automatiquement selon une planification automatisée dans CloudWatch Events au moyen d'expressions cron ou rate. Tous les événements planifiés utilisent le fuseau horaire UTC et la précision minimale pour les horaires est de 1 minute.

CloudWatch Events prend en charge les expressions cron et de fréquence. Les expressions de fréquence sont plus simples à définir, mais ne proposent pas le calendrier précis de contrôle que proposent les expressions cron. Par exemple, une expression cron vous permet de définir une règle qui se déclenche à une heure spécifiée sur un jour de chaque semaine ou mois. En revanche, les expressions de fréquence déclenchent une règle à une fréquence standard, par exemple une fois toutes les heures ou une fois par jour.

> **Note**
>
> CloudWatch Events ne fournit pas de précision de deuxième niveau dans les expressions de planification. Le niveau de résolution maximum lors de l'utilisation d'une expression cron est d'une minute. En raison de la nature distribuée de CloudWatch Events et des services cible, le délai entre le moment où la règle planifiée est déclenchée et celui où le service cible lance l'exécution de la ressource cible peut être de plusieurs secondes. Votre règle planifiée est déclenchée au cours de cette minute, mais pas précisément à la seconde exacte.

Formats

- Expressions cron (p. 34)
- Expressions de fréquence (p. 37)

Expressions cron

Ces expressions se composent de six champs obligatoires qui sont séparés par des espaces.

Syntaxe

```
cron(fields)
```

Champ	Valeurs	Caractères génériques
Minutes	0-59	, - * /
Heures	0-23	, - * /
Jour du mois	1-31	, - * ? / L W
Mois	1-12 ou JAN-DEC	, - * /
Day-of-week	1-7 ou SUN-SAT	, - * ? L #

Champ	Valeurs	Caractères génériques
Année	1970-2199	, - * /

Wildcards

- Le caractère générique , (virgule) inclut des valeurs supplémentaires. Dans le champ Mois, JAN,FEB,MAR englobe janvier, février et mars.
- Le caractère générique - (tiret) spécifie des plages. Dans le champ Jour, 1-15 englobe les jours 1 à 15 du mois spécifié.
- Le caractère générique * (astérisque) inclut toutes les valeurs du champ. Dans le champ Hours, * inclut toutes les heures. Vous ne pouvez pas utiliser * à la fois dans les champs jour-du-mois et jour-de-la-semaine. Si vous l'utilisez dans un champ, vous devez utiliser ? dans l'autre.
- Le caractère générique / (barre oblique) spécifie les incréments. Dans le champ Minutes, vous pouvez entrer 1/10 pour spécifier toutes les dix minutes, à partir de la première minute de l'heure (par exemple, les 11e, 21e, 31e minutes, et ainsi de suite).
- Le caractère générique ? (point d'interrogation) indique l'un ou l'autre. Dans le champ Day-of-month, vous pouvez entrer 7, et si peu importe pour vous le jour de la semaine auquel correspond le 7, vous pouvez entrer ? dans le champ Day-of-week (Jour de la semaine).
- Le caractère générique L dans les champs Day-of-month ou Day-of-week spécifie le dernier jour du mois ou de la semaine.
- Le caractère générique **`W`** dans le champ Day-of-month spécifie un jour de la semaine. Dans le champ Day-of-month, **`3W`** indique le jour le plus proche du troisième jour de la semaine du mois.
- Le caractère générique # dans le champ Day-of-week spécifie une certaine instance du jour de la semaine spécifié dans un mois. Par exemple, 3#2 correspond au deuxième mardi du mois : le 3 fait référence à 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.

  Note

  Si vous utilisez un caractère « # », vous ne pouvez définir qu'une seule expression dans le champ « day-of-week » (jour de la semaine). Par exemple, `"3#1,6#3"` n'est pas valide, car il est interprété comme deux expressions.

Restrictions

- Vous ne pouvez pas spécifier les champs Day-of-month et Day-of-week dans une même expression cron. Si vous spécifiez une valeur dans l'un de ces champs, vous devez utiliser un caractère générique ? (point d'interrogation) dans l'autre.
- Les expressions cron qui entraînent des fréquences d'une rapidité supérieure à 1 minute ne sont pas prises en charge.

Examples

Vous pouvez utiliser les exemples de chaînes cron suivants lorsque vous créez une règle avec planification.

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
0 USD	10	*	*	?	*	Exécuter à 10 h 00

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
						(UTC) chaque jour
15	12	*	*	?	*	Exécuter à 12 h 15 (UTC) chaque jour
0 USD	18	?	*	MON-FRI	*	Exécuter à 18 h 00 (UTC) du lundi au vendredi
0 USD	8	1	*	?	*	Exécuter à 8 h 00 (UTC) chaque 1er jour du mois
0/15	*	*	*	?	*	Exécuter toutes les 15 minutes
0/10	*	?	*	MON-FRI	*	Exécuter toutes les 10 minutes du lundi au vendredi
0/5	8-17	?	*	MON-FRI	*	Exécuter toutes les 5 minutes du lundi au vendredi entre 8 h 00 et 17 h 55 (UTC)

Les exemples suivants indiquent comment utiliser les expressions Cron avec la commande d'AWS CLI put-rule. Le premier exemple crée une règle qui est déclenchée chaque jour à 12 h 00 UTC.

```
aws events put-rule --schedule-expression "cron(0 12 * * ? *)" --name MyRule1
```

L'exemple suivant crée une règle qui est déclenchée chaque jour 5 et 35 minutes après 14 h 00 UTC.

```
aws events put-rule --schedule-expression "cron(5,35 14 * * ? *)" --name MyRule2
```

L'exemple suivant crée une règle qui est déclenchée à 10 h 15 UTC, le dernier vendredi de chaque mois, pendant les années 2002 à 2005.

```
aws events put-rule --schedule-expression "cron(15 10 ? * 6L 2002-2005)" --name MyRule3
```

Expressions de fréquence

Une expression de fréquence démarre au moment où vous créez la règle d'événement planifié, puis s'exécute selon le calendrier défini.

Les expressions de fréquence comportent deux champs obligatoires. Ces champs sont séparés par un espace.

Syntaxe

```
rate(value unit)
```

valeur

Nombre positif.

unité

Unité de temps. Des unités différentes sont nécessaires pour les valeurs de 1 (par exemple `minute`) et les valeurs supérieures à 1, (par exemple, `minutes`).

Valeurs valides : minute | minutes | heure | heures | jour | jours

Restrictions

Si la valeur est égale à 1, l'unité doit être au singulier. De même, pour les valeurs supérieures à 1, l'unité doit être au pluriel. Par exemple, les fréquences rate(1 hours) et rate(5 hour) ne sont pas valides, mais les fréquences rate(1 hour) et rate(5 hours) sont valides.

Examples

Les exemples suivants indiquent comment utiliser les expressions rate avec la commande d'AWS CLI put-rule. Le premier exemple déclenche la règle toutes les minutes, le deuxième la déclenche toutes les cinq minutes, le suivant une fois par heure, et l'exemple final une fois par jour.

```
aws events put-rule --schedule-expression "rate(1 minute)" --name MyRule2
```

```
aws events put-rule --schedule-expression "rate(5 minutes)" --name MyRule3
```

```
aws events put-rule --schedule-expression "rate(1 hour)" --name MyRule4
```

```
aws events put-rule --schedule-expression "rate(1 day)" --name MyRule5
```

Modèles d'événements dans CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Les événements d'Amazon CloudWatch Events sont représentés comme des objets JSON. Pour plus de détails sur les objets JSON, consultez le document RFC 7159. Voici un exemple d'événement :

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/ i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": " i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

Il est important de se souvenir des détails suivants sur un événement :

- Ils ont tous les mêmes champs de niveau supérieur (ceux figurant dans l'exemple ci-dessus) qui ne sont jamais absents.
- Le contenu du champ de niveau supérieur detail est différent en fonction du service à l'origine de l'événement et de l'événement lui-même. La combinaison des champs source et detail-type sert à identifier les champs et les valeurs trouvés dans le champ detail. Pour voir des exemples d'événements générés par les services AWS, consultez Types d'événements CloudWatch Events.

Chaque champ d'événement est décrit ci-dessous.

version ;

Par défaut, ce paramètre est défini sur 0 (zéro) pour tous les événements.

id

Une valeur unique est générée pour chaque événement. Cela peut être utile dans le suivi des événements tandis qu'ils passent par les règles et les cibles, et par le processus de traitement.

detail-type

Identifie, en combinaison avec le champ source, les champs et les valeurs qui apparaissent dans le champ detail.

Tous les événements fournis via CloudTrail ont `AWS API Call via CloudTrail` comme valeur pour `detail-type`. Pour de plus amples informations, veuillez consulter Événements remis via CloudTrail (p. 82).

source

Identifie le service à l'origine de l'événement. Tous les événements générés au sein d'AWS commencent par « AWS ». Les événements générés par un client peuvent présenter n'importe quelle valeur ici, tant que celle-ci ne commence pas par « AWS ». Nous recommandons d'utiliser des chaînes Java domaine-nom inversées de style nom de package.

Pour trouver la valeur correcte de `source` pour un service AWS, consultez le tableau de la section relative aux espaces de noms du service AWS. Par exemple, la valeur `source` pour Amazon CloudFront est `aws.cloudfront`.

compte

Le nombre à 12 chiffres identifiant un compte AWS.

time

L'horodatage d'événement, qui peut être spécifié par le service à l'origine de l'événement. Si l'événement s'étend sur un intervalle de temps, le service peut choisir d'indiquer l'heure de début, auquel cas cette valeur pourra être sensiblement antérieure à l'heure de réception réelle de l'événement.

région

Indique la région AWS à l'origine de l'événement.

resources

Ce tableau JSON contient des ARN qui identifient les ressources liées à l'événement. L'inclusion de ces ARN est laissée à la discrétion du service. Par exemple, les changements de statut des instances Amazon EC2 incluent les ARN d'instances Amazon EC2, les événements incluent à la fois les ARN des instances et des groupes Auto Scaling, mais les appels d'API avec AWS CloudTrail n'incluent pas les ARN de ressources.

detail

Un objet JSON, dont le contenu est laissé à la discrétion du service à l'origine de l'événement. Le contenu du détail de l'exemple ci-dessus est très simple : juste deux champs. AWS Les événements d'appel d'API ont des objets de détail comportant environ 50 champs imbriqués sur plusieurs niveaux.

Modèles d'événements

Les règles utilisent des modèles d'événements pour sélectionner des événements et les acheminer vers des cibles. Un modèle correspond ou pas à un événement. Les modèles d'événements sont représentés en tant qu'objets JSON avec une structure similaire à celle des événements, par exemple :

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "detail": {
    "state": [ "running" ]
  }
```

```
}
```

Il est important de se souvenir des points suivants sur la correspondance des modèles d'événements :

- Pour qu'un modèle corresponde à un événement, cet événement doit contenir tous les noms de domaine figurant dans le modèle. Les noms de domaine doivent apparaître dans l'événement avec la même structure imbriquée.
- Les autres champs de l'événement non mentionnés dans le modèle sont ignorés ; en effet, un caractère générique « * » : « * » est prévu pour les champs non mentionnés.
- La correspondance est exacte (caractère par caractère), sans changement de casse ou autre type de normalisation de chaîne.
- Les valeurs à faire correspondre suivent les règles JSON : chaînes délimitées par guillemets, nombres et mots clés `true`, `false` et `null` sans guillemets.
- La correspondance des nombres est au niveau de représentation de la chaîne. Par exemple, 300, 300,0 et 3.0e2 ne sont pas considérés égaux.

Lorsque vous écrivez des modèles pour correspondre à des événements, vous pouvez utiliser l'API `TestEventPattern` ou la commande CLI `test-event-pattern` pour vous assurer que votre modèle corresponde aux événements souhaités. Pour plus d'informations, consultez TestEventPattern ou test-event-pattern.

Les modèles d'événements suivants correspondraient à l'événement en haut de cette page. Le premier modèle correspond car l'une des valeurs d'instances spécifiées dans le modèle correspond à l'événement (et le modèle ne précise aucun autre champ non contenu dans l'événement). Le deuxième correspond car l'état « terminated » est contenu dans l'événement.

```
{
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
    "arn:aws:ec2:us-east-1:123456789012:instance/i-abcdefgh"
  ]
}
```

```
{
  "detail": {
    "state": [ "terminated" ]
  }
}
```

Ces modèles d'événements ne correspondent pas à l'événement en haut de cette page. Le premier modèle ne correspond pas car il spécifie une valeur « pending » pour l'état, valeur qui n'apparaît pas dans l'événement. Le deuxième modèle ne correspond pas car la valeur de la ressource spécifiée n'apparaît pas dans l'événement.

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "detail": {
    "state": [ "pending" ]
  }
}
```

```
{
  "source": [ "aws.ec2" ],
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1::image/ami-12345678" ]
```

```
}
```

Correspondance de valeurs nulles et chaînes vides dans les modèles d'événement

Vous pouvez créer un modèle qui correspond à un champ d'événement comportant une valeur nulle ou une chaîne vide. Pour voir comment cela fonctionne, testez l'exemple d'événement suivant :

```
{
  "version": "0",
  "id": "3e3c153a-8339-4e30-8c35-687ebef853fe",
  "detail-type": "EC2 Instance Launch Successful",
  "source": "aws.autoscaling",
  "account": "123456789012",
  "time": "2015-11-11T21:31:47Z",
  "region": "us-east-1",
  "resources": [
   ],
  "detail": {
    "eventVersion": "",
    "responseElements": null
   }
}
```

Pour faire correspondre des événements où la valeur de `eventVersion` est une chaîne vide, utilisez le modèle suivant, qui correspond à l'événement en exemple.

```
{
  "detail": {
     "eventVersion": [""]
  }
}
```

Pour faire correspondre des événements où la valeur de `responseElements` est nulle, utilisez le modèle suivant, qui correspond à l'événement en exemple.

```
{
  "detail": {
     "responseElements": [null]
  }
}
```

Les valeurs nulles et les chaînes vides ne sont pas interchangeables dans une correspondance de modèle. Un modèle qui est écrit pour détecter les chaînes vides ne capture pas les valeurs de `null`.

Tableaux dans les modèles CloudWatch Events

La valeur de chaque champ d'un modèle constitue un tableau contenant une ou plusieurs valeurs, et le modèle correspond si l'une des valeurs du tableau correspond à la valeur de l'événement. Lorsque la valeur de l'événement est un tableau, le modèle correspond si l'intersection entre le tableau du modèle et le tableau de l'événement n'est pas vide.

Par exemple, si un exemple de modèle d'événement comprend le texte suivant :

```
"resources": [
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:111122223333:instance/i-b188560f",
   "arn:aws:ec2:us-east-1:444455556666:instance/i-b188560f",
]
```

L'exemple de modèle correspond à un événement qui inclut le texte suivant, car le premier élément du tableau du modèle correspond au second élément du tableau de l'événement.

```
"resources": [
   "arn:aws:autoscaling:us-east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/ASGTerminate",
   "arn:aws:ec2:us-east-1:123456789012:instance/i-b188560f"
]
```

Exemples d'événements CloudWatch Events provenant de services pris en charge

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Les services AWS dans la liste suivante émettent des événements qui peuvent être détectés par CloudWatch Events.

En outre, vous pouvez également utiliser CloudWatch Events avec des services qui n'émettent pas d'événements et ne sont pas affichés sur cette page, en surveillant les événements livrés via CloudTrail. Pour plus d'informations, consultez Événements remis via CloudTrail (p. 82).

Types d'événements

Événements Amazon Augmented AI

Pour obtenir des exemples d'événements générés par Amazon Augmented AI, consultez Utilisation des événements dans Amazon Augmented AI.

Événements Application Auto Scaling

Pour obtenir des exemples d'événements générés par Application Auto Scaling, veuillez consulter Événements Application Auto Scaling et EventBridge.

AWS BatchÉvénements

Pour voir des exemples d'événements générés par AWS Batch, consultez Événements AWS Batch.

Événements Amazon CloudWatch Events planifiés

Voici un exemple d'événement planifié :

```
{
  "id": "53dc4d37-cffa-4f76-80c9-8b7d4a4d2eaa",
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  "account": "123456789012",
  "time": "2019-10-08T16:53:06Z",
  "region": "us-east-1",
  "resources": [ "arn:aws:events:us-east-1:123456789012:rule/MyScheduledRule" ],
  "detail": {}
}
```

Événements Amazon Chime

Pour obtenir des exemples d'événements générés par Amazon Chime, veuillez consulter Automatisation d'Amazon Chime avec EventBridge.

Événements provenant de CloudWatch

Pour obtenir des exemples d'événements de CloudWatch, consultez Événements d'alarme et EventBridge dans le Guide de l'utilisateur AWS CodeBuild.

Événements CodeBuild

Pour voir des exemples d'événements CodeBuild, consultez Référence du format d'entrée des notifications de génération dans le Guide de l'utilisateur AWS CodeBuild.

Événements CodeCommit

Pour voir des exemples d'événements CodeCommit, consultez Surveillance des événements CodeCommit dans EventBridge et CloudWatch Events dans le Guide de l'utilisateur AWS CodeCommit.

AWS CodeDeployÉvénements

Voici des exemples d'événements pour CodeDeploy. Pour plus d'informations, veuillez consulter Surveillance des déploiements à l'aide de CloudWatch Events dans le Guide de l'utilisateur AWS CodeDeploy.

CodeDeploy Deployment State-change Notification (Notification de changement d'état de déploiement CodeDeploy

L'état d'un déploiement a évolué.

```
{
  "account": "123456789012",
  "region": "us-east-1",
  "detail-type": "CodeDeploy Deployment State-change Notification",
  "source": "aws.codedeploy",
  "version": "0",
  "time": "2016-06-30T22:06:31Z",
  "id": "c071bfbf-83c4-49ca-a6ff-3df053957145",
  "resources": [
    "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication",
    "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/
myDeploymentGroup"
  ],
  "detail": {
    "instanceGroupId": "9fd2fbef-2157-40d8-91e7-6845af69e2d2",
    "region": "us-east-1",
    "application": "myApplication",
    "deploymentId": "d-123456789",
    "state": "SUCCESS",
    "deploymentGroup": "myDeploymentGroup"
```

```
  }
}
```

CodeDeploy Instance State-change Notification (Notification de changement d'état d'une instance CodeDeploy

L'état d'une instance qui appartient à un groupe de déploiement a évolué.

```
{
  "account": "123456789012",
  "region": "us-east-1",
  "detail-type": "CodeDeploy Instance State-change Notification",
  "source": "aws.codedeploy",
  "version": "0",
  "time": "2016-06-30T23:18:50Z",
  "id": "fb1d3015-c091-4bf9-95e2-d98521ab2ecb",
  "resources": [
    "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa",
    "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/
myDeploymentGroup",
    "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication"
  ],
  "detail": {
    "instanceId": "i-0000000aaaaaaaaaa",
    "region": "us-east-1",
    "state": "SUCCESS",
    "application": "myApplication",
    "deploymentId": "d-123456789",
    "instanceGroupId": "8cd3bfa8-9e72-4cbe-a1e5-da4efc7efd49",
    "deploymentGroup": "myDeploymentGroup"
  }
}
```

Evénements CodePipeline

Voici des exemples d'événements pour CodePipeline.

Changement d'état d'exécution de pipeline

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Pipeline Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": "1",
    "state": "STARTED",
    "execution-id": "01234567-0123-0123-0123-012345678901"
  }
}
```

Changement d'état d'exécution d'étape

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Stage Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": "1",
    "execution-id": "01234567-0123-0123-0123-012345678901",
    "stage": "Prod",
    "state": "STARTED"
  }
}
```

Changement d'état d'exécution d'action

Dans cet exemple, il y a deux champs `region`. Celui du haut correspond au nom de la région AWS dans laquelle l'action dans le pipeline cible est exécutée. Dans cet exemple, il s'agit de `us-east-1`. L'attribut `region` de la section `detail` correspond à la région AWS dans laquelle l'événement a été créé. C'est la même que la région où le pipeline a été créé. Dans cet exemple, il s'agit de `us-west-2`.

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "CodePipeline Action Execution State Change",
  "source": "aws.codepipeline",
  "account": "123456789012",
  "time": "2017-04-22T03:31:47Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:codepipeline:us-east-1:123456789012:pipeline:myPipeline"
  ],
  "detail": {
    "pipeline": "myPipeline",
    "version": 1,
    "execution-id": "01234567-0123-0123-0123-012345678901",
    "stage": "Prod",
    "action": "myAction",
    "state": "STARTED",
    "region":"us-west-2",
    "type": {
      "owner": "AWS",
      "category": "Deploy",
      "provider": "CodeDeploy",
      "version": 1
    }
  }
}
```

AWS ConfigÉvénements

Pour plus d'informations sur les événements AWS Config, consultez Surveillance de AWS Config à l'aide d'Amazon CloudWatch Events dans le Guide du développeurAWS Config.

Événements Amazon EBS

Pour plus d'informations sur les événements Amazon EBS, consultez Amazon CloudWatch Events pour Amazon EBS dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Événements Amazon EC2 Auto Scaling

Pour plus d'informations sur les événements Auto Scaling, consultez Utiliser CloudWatch Events lorsque votre groupe Auto Scaling évolue dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Événements de recommandation de rééquilibrage des instances Amazon EC2

Pour plus d'informations sur les événements pour les recommandations de rééquilibrage d'instance EC2, consultez Surveillance des signaux de recommandation de rééquilibrage dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Événements d'interruption d'instances Spot Amazon EC2

Pour plus d'informations sur les événements d'interruption d'instances Spot, consultez Avis d'interruption des instances Spot dans le Guide de l'utilisateur Amazon EC2 pour les instances Linux.

Événements de modification de l'état Amazon EC2

Voici des exemples d'événements pour des instances Amazon EC2 lorsque l'état de l'instance change.

Notification de changement d'état de l'instance EC

Cet exemple illustre une instance dans l'état `pending`. Les autres valeurs possibles pour `state` comprennent `running`, `shutting-down`, `stopped`, `stopping` et `terminated`.

```
{
   "id":"7bf73129-1428-4cd3-a780-95db273d1602",
   "detail-type":"EC2 Instance State-change Notification",
   "source":"aws.ec2",
   "account":"123456789012",
   "time":"2019-11-11T21:29:54Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"
   ],
   "detail":{
      "instance-id":"i-abcd1111",
      "state":"pending"
   }
}
```

Événements Amazon Elastic Container Registry

Amazon ECR envoie des événements d'actions d'image à EventBridge. Les événements sont envoyés lorsque les images sont poussées, numérisées ou supprimées.

Pour voir des exemples d'événements Amazon ECS, consultez Événements Amazon ECR dans le Guide de l'utilisateur du registre de conteneur Amazon Elastic.

Événements Amazon Elastic Container Service

Amazon ECS envoie deux types d'événements à EventBridge : événements d'instance de conteneur et événements de tâche. Les événements d'instance de conteneur sont envoyés uniquement si vous utilisez le type de lancement EC2 pour vos tâches. Pour les tâches qui utilisent le type de lancement Fargate, vous ne recevez que les événements d'état des tâches. Amazon ECS suit l'état des instances de conteneur et des tâches. Si une de ces ressources change, un événement est déclenché. Ces événements sont classées comme événements de changement d'état d'instance de conteneur ou événements de changement d'état de tâche.

Pour voir des exemples d'événements Amazon ECS, consultez Événements Amazon ECS dans le Guide du développeur du service de conteneur Amazon Elastic.

Événements AWS Elemental MediaConvert

Pour voir des exemples d'événements MediaConvert, consultez Utilisation de CloudWatch Events pour surveiller les tâches AWS Elemental MediaConvert dans le Guide de l'utilisateur AWS Elemental MediaConvert.

Événements AWS Elemental MediaPackage

Pour voir des exemples d'événements MediaPackage, consultez Surveillance d'AWS Elemental MediaPackage avec Amazon CloudWatch Events dans le Guide de l'utilisateur AWS Elemental MediaPackage.

Événements AWS Elemental MediaStore

Pour voir des exemples d'événements MediaStore, consultez Automatisation d'AWS Elemental MediaStore avec CloudWatch Events dans le Guide de l'utilisateur AWS Elemental MediaStore.

Événements Amazon EMR

Les événements rapportés par Amazon EMR ont `aws.emr` comme valeur pour `Source`, tandis que les événements d'API Amazon EMR rapportés via CloudTrail ont `aws.elasticmapreduce` comme valeur pour `Source`.

Voici des exemples d'événements rapportés par Amazon EMR.

Changement d'état des politiques Amazon EMR Auto Scaling

```
{
   "version":"0",
   "id":"2f8147ab-8c48-47c6-b0b6-3ee23ec8d300",
   "detail-type":"EMR Auto Scaling Policy State Change",
   "source":"aws.emr",
   "account":"123456789012",
   "time":"2016-12-16T20:42:44Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "resourceId":"ig-X2LBMHTGPCBU",
      "clusterId":"j-1YONHTCP3YZKC",
      "state":"PENDING",
      "message":"AutoScaling policy modified by user request",
      "scalingResourceType":"INSTANCE_GROUP"
   }
}
```

Changement d'état de cluster Amazon EMR – Starting (Démarrage en cours

```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:43:05Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "STARTING",
    "message": "Amazon EMR cluster j-123456789ABCD (Development Cluster) was requested at
 2016-12-16 20:42 UTC and  is being created."
  }
}
```

Changement d'état de cluster Amazon EMR – Terminated (Résilié

```
{
  "version": "0",
  "id": "1234abb0-f87e-1234-b7b6-000000123456",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T21:00:23Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"USER_REQUEST\",\"message\":\"Terminated by user
 request\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "TERMINATED",
    "message": "Amazon EMR Cluster jj-123456789ABCD (Development Cluster) has terminated at
 2016-12-16 21:00 UTC with a reason of USER_REQUEST."
  }
}
```

Changement d'état de groupe d'instances Amazon EMR


```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Instance Group State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:57:47Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "market": "ON_DEMAND",
    "severity": "INFO",
    "requestedInstanceCount": "2",
    "instanceType": "m3.xlarge",
    "instanceGroupType": "CORE",
    "instanceGroupId": "ig-ABCDEFGHIJKL",
    "clusterId": "j-123456789ABCD",
    "runningInstanceCount": "2",
    "state": "RUNNING",
    "message": "The resizing operation for instance group ig-ABCDEFGHIJKL in Amazon EMR
 cluster j-123456789ABCD (Development Cluster) is complete. It now has an instance count of
 2. The resize started at 2016-12-16 20:57 UTC and took 0 minutes to complete."
  }
}
```


Changement de statut d'étape Amazon EMR


```
{
  "version": "0",
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Step Status Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:53:09Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "ERROR",
    "actionOnFailure": "CONTINUE",
    "stepId": "s-ZYXWVUTSRQPON",
    "name": "CustomJAR",
    "clusterId": "j-123456789ABCD",
    "state": "FAILED",
    "message": "Step s-ZYXWVUTSRQPON (CustomJAR) in Amazon EMR cluster j-123456789ABCD
 (Development Cluster) failed at 2016-12-16 20:53 UTC."
  }
}
```


Événement Amazon GameLift

Voici des exemples d'événements Amazon GameLift. Pour plus d'informations, consultez Référence des événements FlexMatch dans le Guide du développeur Amazon GameLift.

Recherche de correspondance


```
{
  "version": "0",
  "id": "cc3d3ebe-1d90-48f8-b268-c96655b8f013",
```

```
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T21:15:36.421Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T21:15:35.676Z",
        "players": [
          {
            "playerId": "player-1"
          }
        ]
      }
    ],
    "estimatedWaitMillis": "NOT_AVAILABLE",
    "type": "MatchmakingSearching",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1"
        }
      ]
    }
  }
}
```

Correspondance potentielle créée

```
{
  "version": "0",
  "id": "fce8633f-aea3-45bc-aeba-99d639cad2d4",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T21:17:41.178Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T21:15:35.676Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-08T21:17:40.657Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue"
          }
```

```
        ]
      }
    ],
    "acceptanceTimeout": 600,
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 3,
        "failedCount": 0
      }
    ],
    "acceptanceRequired": true,
    "type": "PotentialMatchCreated",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "team": "blue"
        }
      ]
    },
    "matchId": "3faf26ac-f06e-43e5-8d86-08feff26f692"
  }
}
```

Acceptation de la correspondance

```
{
  "version": "0",
  "id": "b3f76d66-c8e5-416a-aa4c-aa1278153edc",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:04:42.660Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T20:01:35.305Z",
        "players": [
          {
            "playerId": "player-1",
```

```
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T20:04:16.637Z",
        "players": [
          {
            "playerId": "player-2",
            "team": "blue",
            "accepted": false
          }
        ]
      }
    ],
    "type": "AcceptMatch",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "team": "blue",
          "accepted": false
        }
      ]
    },
    "matchId": "848b5f1f-0460-488e-8631-2960934d13e5"
  }
}
```

Acceptation de correspondance terminée

```
{
  "version": "0",
  "id": "b1990d3d-f737-4d6c-b150-af5ace8c35d3",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-08T20:43:14.621Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-08T20:30:40.972Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-08T20:33:14.111Z",
        "players": [
          {
            "playerId": "player-2",
```

```
            "team": "blue"
          }
        ]
      }
    ],
    "acceptance": "TimedOut",
    "type": "AcceptMatchCompleted",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "team": "blue"
        }
      ]
    },
    "matchId": "a0d9bd24-4695-4f12-876f-ea6386dd6dce"
  }
}
```

Réussite de la correspondance

```
{
  "version": "0",
  "id": "5ccb6523-0566-412d-b63c-1569e00d023d",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T19:59:09.159Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T19:58:59.277Z",
        "players": [
          {
            "playerId": "player-1",
            "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
            "team": "red"
          }
        ]
      },
      {
        "ticketId": "ticket-2",
        "startTime": "2017-08-09T19:59:08.663Z",
        "players": [
          {
            "playerId": "player-2",
            "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
            "team": "blue"
          }
        ]
      }
    ],
    "type": "MatchmakingSucceeded",
    "gameSessionInfo": {
      "gameSessionArn": "arn:aws:gamelift:us-west-2:123456789012:gamesession/836cf48d-
bcb0-4a2c-bec1-9c456541352a",
```

```
      "ipAddress": "192.168.1.1",
      "port": 10777,
      "players": [
        {
          "playerId": "player-1",
          "playerSessionId": "psess-6e7c13cf-10d6-4756-a53f-db7de782ed67",
          "team": "red"
        },
        {
          "playerId": "player-2",
          "playerSessionId": "psess-786b342f-9c94-44eb-bb9e-c1de46c472ce",
          "team": "blue"
        }
      ]
    },
    "matchId": "c0ec1a54-7fec-4b55-8583-76d67adb7754"
  }
}
```

Expiration de la correspondance

```
{
  "version": "0",
  "id": "fe528a7d-46ad-4bdc-96cb-b094b5f6bf56",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:11:35.598Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "reason": "TimedOut",
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T20:01:35.305Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      }
    ],
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 3,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 3,
```

```
        "failedCount": 0
      }
    ],
    "type": "MatchmakingTimedOut",
    "message": "Removed from matchmaking due to timing out.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    }
  }
}
```

Annulation de la correspondance

```
{
  "version": "0",
  "id": "8d6f84da-5e15-4741-8d5c-5ac99091c27f",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-09T20:00:07.843Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "reason": "Cancelled",
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-09T19:59:26.118Z",
        "players": [
          {
            "playerId": "player-1"
          }
        ]
      }
    ],
    "ruleEvaluationMetrics": [
      {
        "ruleName": "EvenSkill",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "EvenTeams",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "FastConnection",
        "passedCount": 0,
        "failedCount": 0
      },
      {
        "ruleName": "NoobSegregation",
        "passedCount": 0,
        "failedCount": 0
      }
    ],
    "type": "MatchmakingCancelled",
```

```
    "message": "Cancelled by request.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1"
        }
      ]
    }
  }
}
```

Echec de la correspondance

```
{
  "version": "0",
  "id": "025b55a4-41ac-4cf4-89d1-f2b3c6fd8f9d",
  "detail-type": "GameLift Matchmaking Event",
  "source": "aws.gamelift",
  "account": "123456789012",
  "time": "2017-08-16T18:41:09.970Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:gamelift:us-west-2:123456789012:matchmakingconfiguration/SampleConfiguration"
  ],
  "detail": {
    "tickets": [
      {
        "ticketId": "ticket-1",
        "startTime": "2017-08-16T18:41:02.631Z",
        "players": [
          {
            "playerId": "player-1",
            "team": "red"
          }
        ]
      }
    ],
    "customEventData": "foo",
    "type": "MatchmakingFailed",
    "reason": "UNEXPECTED_ERROR",
    "message": "An unexpected error was encountered during match placing.",
    "gameSessionInfo": {
      "players": [
        {
          "playerId": "player-1",
          "team": "red"
        }
      ]
    },
    "matchId": "3ea83c13-218b-43a3-936e-135cc570cba7"
  }
}
```

AWS GlueÉvénements

Voici le format pour les événements AWS Glue.

Exécution de tâche réussie

```
{
    "version":"0",
```

```
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-09-07T18:57:21Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"INFO",
        "state":"SUCCEEDED",
        "jobRunId":"jr_abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789",
        "message":"Job run succeeded"
    }
}
```

Echec de l'exécution de la tâche

```
{
    "version":"0",
    "id":"abcdef01-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-09-07T06:02:03Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"ERROR",
        "state":"FAILED",
        "jobRunId":"jr_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef",
        "message":"JobName:MyJob and
 JobRunId:jr_0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef failed to
 execute with exception Role arn:aws:iam::123456789012:role/Glue_Role should be given
 assume role permissions for Glue Service."
    }
}
```

Expiration

```
{
    "version":"0",
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Glue Job State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-11-20T20:22:06Z",
    "region":"us-east-1",
    "resources":[],
    "detail":{
        "jobName":"MyJob",
        "severity":"WARN",
        "state":"TIMEOUT",
        "jobRunId":"jr_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
        "message":"Job run timed out"
    }
}
```

Exécution de la tâche arrêtée

```
{
```

```
   "version":"0",
   "id":"abcdef00-1234-5678-9abc-def012345678",
   "detail-type":"Glue Job State Change",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2017-11-20T20:22:06Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
       "jobName":"MyJob",
       "severity":"INFO",
       "state":"STOPPED",
       "jobRunId":"jr_abc0123456789abcdef0123456789abcdef0123456789abcdef0123456789def",
       "message":"Job run stopped"
    }
}
```

Analyse démarrée

```
{
   "version":"0",
   "id":"05efe8a2-c309-6884-a41b-3508bcdc9695",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"561226563745",
   "time":"2017-11-11T01:09:46Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "accountId":"561226563745",
      "crawlerName":"S3toS3AcceptanceTestCrawlera470bd94-9e00-4518-8942-e80c8431c322",
      "startTime":"2017-11-11T01:09:46Z",
      "state":"Started",
      "message":"Crawler Started"
   }
}
```

Analyse réussie

```
{
   "version":"0",
   "id":"3d675db5-59b9-6388-b8e8-e0a9b6d567a9",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"561226563745",
   "time":"2017-11-11T01:25:00Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "tablesCreated":"0",
      "warningMessage":"N/A",
      "partitionsUpdated":"0",
      "tablesUpdated":"0",
      "message":"Crawler Succeeded",
      "partitionsDeleted":"0",
      "accountId":"561226563745",
      "runningTime (sec)":"7",
      "tablesDeleted":"0",
      "crawlerName":"SchedulerTestCrawler51fb3a8b-1015-49f0-a969-ca126680b94b",
```

```
      "completionDate":"2017-11-11T01:25:00Z",
      "state":"Succeeded",
      "partitionsCreated":"0",
      "cloudWatchLogLink":"https://console.aws.amazon.com/
cloudwatch/home?region=us-east-1#logEventViewer:group=/aws-glue/
crawlers;stream=SchedulerTestCrawler51fb3a8b-1015-49f0-a969-ca126680b94b"
   }
}
```

Échec de l'analyse

```
{
   "version":"0",
   "id":"f7965b59-470f-2e06-bb89-a8cebaabefac",
   "detail-type":"Glue Crawler State Change",
   "source":"aws.glue",
   "account":"782104008917",
   "time":"2017-10-20T05:10:08Z",
   "region":"us-east-1",
   "resources":[

   ],
   "detail":{
      "crawlerName":"test-crawler-notification",
      "errorMessage":"Internal Service Exception",
      "accountId":"1234",
      "cloudWatchLogLink":"https://console.aws.amazon.com/cloudwatch/home?region=us-
east-1#logEventViewer:group=/aws-glue/crawlers;stream=test-crawler-notification",
      "state":"Failed",
      "message":"Crawler Failed"
   }
}
```

L'exécution de la tâche est à l'état Starting

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"STARTING",
          "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in STARTING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

L'exécution de la tâche est à l'état Running

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
```

```
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"RUNNING",
    "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in RUNNING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

L'exécution de la tâche est à l'état Stopping

```
{
   "version":"0",
   "id":"66fbc5e1-aac3-5e85-63d0-856ec669a050",
   "detail-type":"Glue Job Run Status",
   "source":"aws.glue",
   "account":"123456789012",
   "time":"2018-04-24T20:57:34Z",
   "region":"us-east-1",
   "resources":[],
   "detail":{
      "jobName":"MyJob",
      "severity":"INFO",
      "notificationCondition":{
         "NotifyDelayAfter":1.0
      },
      "state":"STOPPING",
    "jobRunId":"jr_6aa58e7a3aa44e2e4c7db2c50e2f7396cb57901729e4b702dcb2cfbbeb3f7a86",
      "message":"Job is in STOPPING state",
      "startedOn":"2018-04-24T20:55:47.941Z"
   }
}
```

AWS Glue Changement d'état du tableau de catalogue de données

```
{
    "version": "0",
    "id": "2617428d-715f-edef-70b8-d210da0317a0",
    "detail-type": "Glue Data Catalog Table State Change",
    "source": "aws.glue",
    "account": "123456789012",
    "time": "2019-01-16T18:16:01Z",
    "region": "eu-west-1",
    "resources": [
        "arn:aws:glue:eu-west-1:123456789012:table/d1/t1"
    ],
    "detail": {
        "databaseName": "d1",
        "changedPartitions": [
            "[C.pdf, dir3]",
            "[D.doc, dir4]"
        ],
        "typeOfChange": "BatchCreatePartition",
```

```
        "tableName": "t1"
    }
}
```

AWS Glue Changement d'état de la base de données de catalogue de données

Dans l'exemple suivant, le `typeofChange` est `CreateTable`. D'autres valeurs possibles pour ce champ sont `CreateDatabase` et `UpdateTable`.

```
{
    "version": "0",
    "id": "60e7ddc2-a588-5328-220a-21c060f6c3f4",
    "detail-type": "Glue Data Catalog Database State Change",
    "source": "aws.glue",
    "account": "123456789012",
    "time": "2019-01-16T18:08:48Z",
    "region": "eu-west-1",
    "resources": [
        "arn:aws:glue:eu-west-1:123456789012:table/d1/t1"
    ],
    "detail": {
        "databaseName": "d1",
        "typeOfChange": "CreateTable",
        "changedTables": [
            "t1"
        ]
    }
}
```

AWS Ground StationÉvénements

Pour plus d'informations sur les exemples d'événements AWS Ground Station, consultez Automatisation de AWS Ground Station avec CloudWatch Events dans le Guide de l'utilisateur AWS Ground Station.

Événements Amazon GuardDuty

Pour plus d'informations sur des exemples d'événements Amazon GuardDuty, consultez Surveillance d'Amazon GuardDuty avec Amazon CloudWatch Events dans le Guide de l'utilisateur d'Amazon GuardDuty.

AWS HealthÉvénements

Voici le format des événements AWS Personal Health Dashboard (AWS Health). Pour plus d'informations, consultez Gestion des événements AWS Health avec Amazon CloudWatch Events dans le Guide de l'utilisateur AWS Health.

AWS Health Format des événements

```
{
  "version": "0",
  "id": "7bf73129-1428-4cd3-a780-95db273d1602",
  "detail-type": "AWS Health Event",
```

```
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "region",
  "resources": [],
  "detail": {
    "eventArn": "arn:aws:health:region::event/id",
    "service": "service",
    "eventTypeCode": "AWS_service_code",
    "eventTypeCategory": "category",
    "startTime": "Sun, 05 Jun 2016 05:01:10 GMT",
    "endTime": "Sun, 05 Jun 2016 05:30:57 GMT",
    "eventDescription": [{
      "language": "lang-code",
      "latestDescription": "description"
    }]
    ...
  }
}
```

eventTypeCategory

Code de catégorie d'événement. Les valeurs possibles sont `issue`, `accountNotification` et `scheduledChange`.

eventTypeCode

Identifiant unique du type d'événement. Exemples : `AWS_EC2_INSTANCE_NETWORK_MAINTENANCE_SCHEDULED` et `AWS_EC2_INSTANCE_REBOOT_MAINTENANCE_SCHEDULED`. Les événements qui incluent `MAINTENANCE_SCHEDULED` sont généralement transmise environ deux semaines avant le `startTime`.

id

Identifiant unique de l'événement.

web

Le service AWS concerné par l'événement. Par exemple, `EC2`, `S3` ou `REDSHIFT` ou `RDS`.

Problème d'API Elastic Load Balancing

```
{
  "version": "0",
  "id": "121345678-1234-1234-1234-123456789012",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "ap-southeast-2",
  "resources": [],
  "detail": {
    "eventArn": "arn:aws:health:ap-southeast-2::event/
AWS_ELASTICLOADBALANCING_API_ISSUE_90353408594353980",
    "service": "ELASTICLOADBALANCING",
    "eventTypeCode": "AWS_ELASTICLOADBALANCING_API_ISSUE",
    "eventTypeCategory": "issue",
    "startTime": "Sat, 11 Jun 2016 05:01:10 GMT",
    "endTime": "Sat, 11 Jun 2016 05:30:57 GMT",
    "eventDescription": [{
      "language": "en_US",
      "latestDescription": "A description of the event will be provided here"
  }
```

```
}
```

Performances dégradées du lecteur de stockage d'instances Amazon EC

```
{
  "version": "0",
  "id": "121345678-1234-1234-1234-123456789012",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2016-06-05T06:27:57Z",
  "region": "us-west-2",
  "resources": [
    "i-abcd1111"
  ],
  "detail": {
    "eventArn": "arn:aws:health:us-west-2::event/
AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED_90353408594353980",
    "service": "EC2",
    "eventTypeCode": "AWS_EC2_INSTANCE_STORE_DRIVE_PERFORMANCE_DEGRADED",
    "eventTypeCategory": "issue",
    "startTime": "Sat, 05 Jun 2016 15:10:09 GMT",
    "eventDescription": [{
      "language": "en_US",
      "latestDescription": "A description of the event will be provided here"
    }],
    "affectedEntities": [{
      "entityValue": "i-abcd1111",
      "tags": {
        "stage": "prod",
        "app": "my-app"
  }
}
```

AWS KMSÉvénements

Voici des exemples d'événements AWS Key Management Service (AWS KMS). Pour de plus amples informations, veuillez consulter Événements AWS KMS dans le Guide du développeur AWS Key Management Service.

Rotation de clé CMK dans KMS

AWS KMS a effectué une rotation automatique d'une clé CMK.

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-25T21:05:33Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Expiration d'éléments de clé importés KMS

AWS KMS a supprimé une clé expirée de CMK.

```
{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-22T20:12:19Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Suppression d'une clé CMK dans KMS

AWS KMS a effectué la suppression CMK programmée.

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2016-08-19T03:23:45Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

Evénements Amazon Macie

Pour obtenir des exemples d'événements générés par Amazon Macie, consultez Schéma d'événement pour les résultats Amazon Macie.

AWS Management ConsoleÉvénements de connexion

AWS Management ConsoleLes événements de connexion peuvent être détectés par CloudWatch Events uniquement dans la région USA Est (Virginie du Nord).

Voici un exemple d'événement de connexion à la console :

```
{
  "id": "6f87d04b-9f74-4f04-a780-7acf4b0a9b38",
```

```
  "detail-type": "AWS Console Sign In via CloudTrail",
  "source": "aws.signin",
  "account": "123456789012",
  "time": "2016-01-05T18:21:27Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "Root",
            "principalId": "123456789012",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012"
            },
      "eventTime": "2016-01-05T18:21:27Z",
      "eventSource": "signin.amazonaws.com",
      "eventName": "ConsoleLogin",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "0.0.0.0",
      "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36",
      "requestParameters": null,
      "responseElements": {
            "ConsoleLogin": "Success"
            },
      "additionalEventData": {
            "LoginTo": "https://console.aws.amazon.com/console/home?state=hashArgs
%23&isauthcode=true",
            "MobileVersion": "No",
            "MFAUsed": "No" },
      "eventID": "324731c0-64b3-4421-b552-dfc3c27df4f6",
      "eventType": "AwsConsoleSignIn"
      }
}
```


AWS OpsWorksÉvénements Stacks

Voici des exemples d'événements Stacks AWS OpsWorks.

AWS OpsWorks Changement d'état d'instance Stacks

Indique un changement d'état d'une instance Stacks AWS OpsWorks. Les états d'instance sont les suivants.

- `booting`
- `connection_lost`
- `online`
- `pending`
- `rebooting`
- `requested`
- `running_setup`
- `setup_failed`
- `shutting_down`
- `start_failed`
- `stopping`
- `stop_failed`
- `stopped`

- `terminating`
- `terminated`

```
{
  "version": "0",
  "id": "dc5fa8df-48f1-2108-b1b9-1fe5ebcf2296",
  "detail-type": "OpsWorks Instance State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-25T11:12:23Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50z3e4z500z"
  ],
  "detail": {
    "initiated_by": "user",
    "hostname": "testing1",
    "stack-id": "acd3df16-e859-4598-8414-377b12a902da",
    "layer-ids": [
      "d1a0cb7f-c7e9-4a63-811c-976f0267b2c8"
    ],
    "instance-id": "a648d98f-fdd8-4323-952a-a50z3e4z500z",
    "ec2-instance-id": "i-08b1c2b67aa292276",
    "status": "requested"
  }
}
```

Le champ `initiated_by` est uniquement renseigné lorsque l'instance se trouve dans l'état `requested`, `terminating` ou `stopping`. Le champ `initiated_by` peut comporter l'une des valeurs suivantes.

- `user` - Un utilisateur a demandé le changement d'état de l'instance à l'aide de l'API ou d AWS Management Console.
- `auto-scaling` - La fonction de dimensionnement automatique d'AWS OpsWorks Stacks a initié le changement d'état de l'instance.
- `auto-healing` – La fonction de réparation automatique d'AWS OpsWorks Stacks a initié le changement d'état de l'instance.

AWS OpsWorks Changement d'état de commande Stacks

Un changement s'est produit dans l'état d'une commande AWS OpsWorks Stacks. Les états suivants sont des états de commande.

- `expired`– Délai d'expiration d'une commande.
- `failed` - Un échec de la commande générale s'est produit.
- `skipped` – Une commande a été ignorée, car l'instance a un autre état dans AWS OpsWorks Stacks que dans Amazon EC2.
- `successful` – Une commande a réussi.
- `superseded` – Une commande a été ignorée, car elle aurait appliqué des modifications de configuration qui l'ont déjà été.

```
{
  "version": "0",
  "id": "96c778b6-a40e-c8c1-aafc-c9852a3a7b52",
  "detail-type": "OpsWorks Command State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
```

```
  "time": "2018-01-26T08:54:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50a3e4e500f"
  ],
  "detail": {
    "command-id": "acc9f4f3-a3ec-4fab-b70f-c7d04e71e3ec",
    "instance-id": "a648d98f-fdd8-4323-952a-a50a3e4e500f",
    "type": "setup",
    "status": "successful"
  }
}
```

AWS OpsWorks Changement d'état de déploiement Stacks

Un changement s'est produit dans l'état d'un déploiement d'AWS OpsWorks Stacks. Les états suivants sont des états de déploiement.

- `running`
- `successful`
- `failed`

```
{
  "version": "0",
  "id": "b8230afa-60c7-f43f-b632-841c1cfb22ff",
  "detail-type": "OpsWorks Deployment State Change",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-25T11:15:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:opsworks:us-east-1:123456789012:instance/a648d98f-fdd8-4323-952a-a50a3e4e500f"
  ],
  "detail": {
    "duration": 16,
    "stack-id": "acd3df16-e859-4598-8414-377b12a902da",
    "instance-ids": [
      "a648d98f-fdd8-4323-952a-a50a3e4e500f"
    ],
    "deployment-id": "606419dc-418e-489c-8531-bff9770fc346",
    "command": "configure",
    "status": "successful"
  }
}
```

Le champ `duration` est uniquement renseigné lorsqu'un déploiement est terminé, et affiche le temps en secondes.

AWS OpsWorks Alerte Stacks

Une erreur de service AWS OpsWorks Stacks est intervenue.

```
{
  "version": "0",
  "id": "f99faa6f-0e27-e398-95bb-8f190806d275",
  "detail-type": "OpsWorks Alert",
  "source": "aws.opsworks",
  "account": "123456789012",
  "time": "2018-01-20T16:51:29Z",
  "region": "us-east-1",
  "resources": [],
```

```
  "detail": {
    "stack-id": "2f48f2be-ac7d-4dd5-80bb-88375f94db7b",
    "instance-id": "986efb74-69e8-4c6d-878e-5b77c054cbb0",
    "type": "InstanceStop",
    "message": "The shutdown of the instance timed out. Please try stopping it again."
  }
}
```

Événements SageMaker

Pour plus d'informations sur les exemples d'événements SageMaker, consultez Automatisation de SageMaker avec Amazon EventBridge dans le Guide du développeur SageMaker

AWS Security HubÉvénements

Pour plus d'informations sur les exemples d'événements Security Hub, consultez Surveillance d'AWS Security Hub avec Amazon CloudWatch Events dans le Guide de l'utilisateur AWS Security Hub.

AWS Server Migration ServiceÉvénements

Voici des exemples d'événements pour AWS Server Migration Service.

Suppression de la notification des tâches de réplication

```
{
    "version": "0",
    "id": "5630992d-92cd-439f-f2a8-92c8212aee24",
    "detail-type": "Server Migration Job State Change",
    "source": "aws.sms",
    "account": "123456789012",
    "time": "2018-02-07T22:30:11Z",
    "region": "us-west-1",
    "resources": [
        "arn:aws:sms:us-west-1:123456789012:sms-job-21a64348"
    ],
    "detail": {
        "state": "Deleted",
        "replication-run-id": "N/A",
        "replication-job-id": "sms-job-21a64348",
        "version": "1.0"
    }
}
```

Notification des tâches de réplication terminées

```
{
    "version": "0",
    "id": "3f9c59cc-f941-522a-be6d-f08e44ff1715",
    "detail-type": "Server Migration Job State Change",
    "source": "aws.sms",
    "account": "123456789012",
    "time": "2018-02-07T22:54:00Z",
    "region": "us-west-1",
    "resources": [
        "arn:aws:sms:us-west-1:123456789012:sms-job-2ea64347",
```

```
        "arn:aws:sms:us-west-1:123456789012:sms-job-2ea64347/sms-run-e1a64388"
    ],
    "detail": {
        "state": "Completed",
        "replication-run-id": "sms-run-e1a64388",
        "replication-job-id": "sms-job-2ea64347",
        "ami-id": "ami-746d6314",
        "version": "1.0"
    }
}
```

AWS Systems ManagerÉvénements

Voici des exemples d'événements pour AWS Systems Manager. Pour plus d'informations, veuillez consulter Surveillance des événements Systems Manager avec Amazon EventBridge dans le Guide de l'utilisateur AWS Systems Manager.

Types d'événement Systems Manager

- AWS Systems ManagerÉvénements Automation (p. 71)
- AWS Systems ManagerÉvénements Change Calendar (p. 72)
- AWS Systems ManagerÉvénements de conformité (p. 73)
- AWS Systems ManagerÉvénements de fenêtres de maintenance (p. 74)
- AWS Systems ManagerÉvénements Parameter Store (p. 76)
- AWS Systems ManagerÉvénements Run Command de (p. 77)
- AWS Systems ManagerÉvénements du gestionnaire d'états (p. 78)

AWS Systems ManagerÉvénements Automation

Notification de changement de statut d'une étape d'automatisation

```
{
  "version": "0",
  "id": "eeca120b-a321-433e-9635-dab369006a6b",
  "detail-type": "EC2 Automation Step Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-1:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-1:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "EndTime": "Nov 29, 2016 7:43:25 PM",
    "StartTime": "Nov 29, 2016 7:43:23 PM",
    "Time": 2630.0,
    "StepName": "runFixedCmds",
    "Action": "aws:runCommand"
  }
}
```

Notification de changement de statut d'exécution de l'automatisation

```
{
  "version": "0",
  "id": "d290ece9-1088-4383-9df6-cd5b4ac42b99",
  "detail-type": "EC2 Automation Execution Status-change Notification",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2016-11-29T19:43:35Z",
  "region": "us-east-1",
  "resources": ["arn:aws:ssm:us-east-1:123456789012:automation-
execution/333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "arn:aws:ssm:us-east-1:123456789012:automation-definition/runcommand1:1"],
  "detail": {
    "ExecutionId": "333ba70b-2333-48db-b17e-a5e69c6f4d1c",
    "Definition": "runcommand1",
    "DefinitionVersion": 1.0,
    "Status": "Success",
    "StartTime": "Nov 29, 2016 7:43:20 PM",
    "EndTime": "Nov 29, 2016 7:43:26 PM",
    "Time": 5753.0,
    "ExecutedBy": "arn:aws:iam::123456789012:user/userName"
  }
}
```

AWS Systems ManagerÉvénements Change Calendar

Voici des exemples d'événements pour Change Calendar AWS Systems Manager.

> **Note**
>
> Les modifications d'état pour les planifications partagées à partir d'autres AWS ne sont pas prises en charge actuellement.

Calendrier OPEN (OUVERT

```
{
    "version": "0",
    "id": "47a3f03a-f30d-1011-ac9a-du3bdEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "111222333444",
    "time": "2020-09-19T18:00:07Z",
    "region": "us-east-2",
    "resources": [
        "arn:aws:ssm:us-east-2:111222333444:document/MyCalendar"
    ],
    "detail": {
        "state": "OPEN",
        "atTime": "2020-09-19T18:00:07Z",
        "nextTransitionTime": "2020-10-11T18:00:07Z"
    }
}
```

Calendrier CLOSED (FERMÉ

```
{
    "version": "0",
    "id": "f30df03a-1011-ac9a-47a3-f761eEXAMPLE",
    "detail-type": "Calendar State Change",
    "source": "aws.ssm",
    "account": "111222333444",
    "time": "2020-09-17T21:40:02Z",
    "region": "us-east-2",
```

```
    "resources": [
        "arn:aws:ssm:us-east-2:111222333444:document/MyCalendar"
    ],
    "detail": {
        "state": "CLOSED",
        "atTime": "2020-08-17T21:40:00Z",
        "nextTransitionTime": "2020-09-19T18:00:07Z"
    }
}
```

AWS Systems ManagerÉvénements de conformité

Voici des exemples d'événements pour la conformité de AWS Systems Manager.

Conforme à l'association

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:03:26Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Non conforme à l'association

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:02:31Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "last-runtime": "2017-01-01T10:10:10Z",
    "compliance-status": "non_compliant",
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-type": "Association"
  }
}
```

Conforme aux correctifs

```
{
```

```
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:03:26Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

Non conforme aux correctifs

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Configuration Compliance State Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-07-17T19:02:31Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ssm:us-west-1:461348341421:managed-instance/i-01234567890abcdef"
  ],
  "detail": {
    "resource-type": "managed-instance",
    "resource-id": "i-01234567890abcdef",
    "compliance-status": "non_compliant",
    "compliance-type": "Patch",
    "patch-baseline-id": "PB789",
    "severity": "critical"
  }
}
```

AWS Systems ManagerÉvénements de fenêtres de maintenance

Voici des exemples d'événements pour les fenêtres de maintenance de Systems Manager.

Enregistrer une cible

L'autre valeur d'état valide est `DEREGISTERED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Target Registration Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T00:58:37Z",
   "region":"us-east-1",
   "resources":[
```

```
      "arn:aws:ssm:us-west-2:001312665065:maintenancewindow/mw-0ed7251d3fcf6e0c2",
      "arn:aws:ssm:us-west-2:001312665065:windowtarget/
e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6"
   ],
   "detail":{
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-7d3ca86c32a6",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "status":"REGISTERED"
   }
}
```

Type d'exécution de fenêtre

Les autres valeurs d'état valides sont `PENDING`, `IN_PROGRESS`, `SUCCESS`, `FAILED`, `TIMED_OUT` et `SKIPPED_OVERLAPPING`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Execution State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "status":"TIMED_OUT"
   }
}
```

Type d'exécution de tâche

Les autres valeurs d'état valides sont `IN_PROGRESS`, `SUCCESS`, `FAILED` et `TIMED_OUT`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Execution State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:56Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.759Z",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "end-time":"2016-11-16T01:00:56.847Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "status":"TIMED_OUT"
   }
}
```

Cible de tâche traitée

Les autres valeurs d'état valides sont `IN_PROGRESS`, `SUCCESS`, `FAILED` et `TIMED_OUT`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window Task Target Invocation State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T01:00:57Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "start-time":"2016-11-16T01:00:56.427Z",
      "end-time":"2016-11-16T01:00:57.070Z",
      "window-id":"mw-0ed7251d3fcf6e0c2",
      "window-execution-id":"b60fb56e-776c-4e5c-84ee-123456789012",
      "task-execution-id":"6417e808-7f35-4d1a-843f-123456789012",
      "window-target-id":"e7265f13-3cc5-4f2f-97a9-123456789012",
      "status":"TIMED_OUT",
      "owner-information":"Owner"
   }
}
```

Changement d'état de la fenêtre

Les valeurs d'état valides sont `ENABLED` et `DISABLED`.

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-0123456789ab",
   "detail-type":"Maintenance Window State-change Notification",
   "source":"aws.ssm",
   "account":"012345678901",
   "time":"2016-11-16T00:58:37Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ssm:us-west-2:0123456789ab:maintenancewindow/mw-123456789012345678"
   ],
   "detail":{
      "window-id":"mw-123456789012",
      "status":"DISABLED"
   }
}
```

AWS Systems ManagerÉvénements Parameter Store

Vous trouverez ci-dessous des exemples d'événements pour Systems Manager Parameter Store.

Création de paramètre

```
{
  "version": "0",
  "id": "6a7e4feb-b491-4cf7-a9f1-bf3703497718",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:43:48Z",
  "region": "us-east-1",
```

```
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Create",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Mise à jour de paramètre

```
{
  "version": "0",
  "id": "9547ef2d-3b7e-4057-b6cb-5fdf09ee7c8f",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:44:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Update",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

Suppression de paramètre

```
{
  "version": "0",
  "id": "80e9b391-6a9b-413c-839a-453b528053af",
  "detail-type": "Parameter Store Change",
  "source": "aws.ssm",
  "account": "123456789012",
  "time": "2017-05-22T16:45:48Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ssm:us-east-1:123456789012:parameter/foo"
  ],
  "detail": {
    "operation": "Delete",
    "name": "foo",
    "type": "String",
    "description": "Sample Parameter"
  }
}
```

AWS Systems ManagerÉvénements Run Command de

Notification de changement de statut d'Exécuter la commande

```
{
    "version": "0",
```

```
    "id": "51c0891d-0e34-45b1-83d6-95db273d1602",
    "detail-type": "EC2 Command Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "expire-after": "2016-07-14T22:01:30.049Z",
        "parameters": {
            "executionTimeout": ["3600"],
            "commands": ["date"]
        },
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

Notification de changement de statut d'un appel d'Exécuter la commande

```
{
    "version": "0",
    "id": "4780e1b8-f56b-4de5-95f2-95db273d1602",
    "detail-type": "EC2 Command Invocation Status-change Notification",
    "source": "aws.ssm",
    "account": "123456789012",
    "time": "2016-07-10T21:51:32Z",
    "region": "us-east-1",
    "resources": ["arn:aws:ec2:us-east-1:123456789012:instance/i-abcd1111"],
    "detail": {
        "command-id": "e8d3c0e4-71f7-4491-898f-c9b35bee5f3b",
        "document-name": "AWS-RunPowerShellScript",
        "instance-id": "i-9bb89e2b",
        "requested-date-time": "2016-07-10T21:51:30.049Z",
        "status": "Success"
    }
}
```

AWS Systems ManagerÉvénements du gestionnaire d'états

Changements de statut d'association dans State Manager

```
{
   "version":"0",
   "id":"db839caf-6f6c-40af-9a48-25b2ae2b7774",
   "detail-type":"EC2 State Manager Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-05-16T23:01:10Z",
   "region":"us-west-1",
   "resources":[
      "arn:aws:ssm:us-west-1::document/AWS-RunPowerShellScript"
   ],
   "detail":{
      "association-id":"6e37940a-23ba-4ab0-9b96-5d0a1a05464f",
      "document-name":"AWS-RunPowerShellScript",
      "association-version":"1",
      "document-version":"Optional.empty",
```

```
      "targets":"[{\"key\":\"InstanceIds\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2017-02-13T17:22:54.458Z",
      "last-successful-execution-date":"2017-05-16T23:00:01Z",
      "last-execution-date":"2017-05-16T23:00:01Z",
      "last-updated-date":"2017-02-13T17:22:54.458Z",
      "status":"Success",
      "association-status-aggregated-count":"{\"Success\":1}",
      "schedule-expression":"cron(0 */30 * * * ? *)",
      "association-cwe-version":"1.0"
   }
}
```

Changements de statut d'association d'instance dans State Manager

```
{
   "version":"0",
   "id":"6a7e8feb-b491-4cf7-a9f1-bf3703467718",
   "detail-type":"EC2 State Manager Instance Association State Change",
   "source":"aws.ssm",
   "account":"123456789012",
   "time":"2017-02-23T15:23:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:ec2:us-east-1:123456789012:instance/i-12345678",
      "arn:aws:ssm:us-east-1:123456789012:document/my-custom-document"
   ],
   "detail":{
      "association-id":"34fcb7e0-9a14-4984-9989-0e04e3f60bd8",
      "instance-id":"i-12345678",
      "document-name":"my-custom-document",
      "document-version":"1",
      "targets":"[{\"key\":\"instanceids\",\"values\":[\"i-12345678\"]}]",
      "creation-date":"2017-02-23T15:23:48Z",
      "last-successful-execution-date":"2017-02-23T16:23:48Z",
      "last-execution-date":"2017-02-23T16:23:48Z",
      "status":"Success",
      "detailed-status":"",
      "error-code":"testErrorCode",
      "execution-summary":"testExecutionSummary",
      "output-url":"sampleurl",
      "instance-association-cwe-version":"1"
   }
}
```

AWS Step FunctionsÉvénements

Pour des exemples d'événements Step Functions, consultez Exemples d'événement Step Functions dans le Guide du développeur AWS Step Functions.

Événements de modification de balise sur les ressources AWS

Voici un exemple d'événement de balise.

```
{
    "version": "0",
```

```
    "id": "ffd8a6fe-32f8-ef66-c85c-111111111111",
    "detail-type": "Tag Change on Resource",
    "source": "aws.tag",
    "account": "123456789012",
    "time": "2018-09-18T20:41:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ec2:us-east-1:123456789012:instance/i-0000000aaaaaaaaaa"
    ],
    "detail": {
        "changed-tag-keys": [
            "key2",
            "key3"
        ],
        "service": "ec2",
        "resource-type": "instance",
        "version": 5,
        "tags": {
            "key4": "value4",
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

AWS Trusted AdvisorÉvénements

Voici des exemples d'événements pour AWS Trusted Advisor. Pour plus d'informations, veuillez consulter Surveillance des résultats des contrôles Trusted Advisor avec Amazon CloudWatch Events dans le Guide de l'utilisateur AWS Support.

Instances Amazon EC2 sous-exploitées

```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T20:07:49Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "check-name": "Low Utilization Amazon EC2 Instances",
    "check-item-detail": {
      "Day 1": "0.1%  0.00MB",
      "Day 2": "0.1%  0.00MB",
      "Day 3": "0.1%  0.00MB",
      "Region/AZ": "ca-central-1a",
      "Estimated Monthly Savings": "$9.22",
      "14-Day Average CPU Utilization": "0.1%",
      "Day 14": "0.1%  0.00MB",
      "Day 13": "0.1%  0.00MB",
      "Day 12": "0.1%  0.00MB",
      "Day 11": "0.1%  0.00MB",
      "Day 10": "0.1%  0.00MB",
      "14-Day Average Network I/O": "0.00MB",
      "Number of Days Low Utilization": "14 days",
      "Instance Type": "t2.micro",
      "Instance ID": "i-01234567890abcdef",
      "Day 8": "0.1%  0.00MB",
```

```
      "Instance Name": null,
      "Day 9": "0.1%  0.00MB",
      "Day 4": "0.1%  0.00MB",
      "Day 5": "0.1%  0.00MB",
      "Day 6": "0.1%  0.00MB",
      "Day 7": "0.1%  0.00MB"
    },
    "status": "WARN",
    "resource_id": "arn:aws:ec2:ca-central-1:123456789012:instance/i-01234567890abcdef",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```

Optimisation des programmes Elastic Load Balancer

```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T20:07:03Z",
  "region": "us-east-2",
  "resources": [],
  "detail": {
    "check-name": "Load Balancer Optimization ",
    "check-item-detail": {
      "Instances in Zone a": "1",
      "Status": "Yellow",
      "Instances in Zone b": "0",
      "# of Zones": "2",
      "Region": "eu-central-1",
      "Load Balancer Name": "my-load-balance",
      "Instances in Zone e": null,
      "Instances in Zone c": null,
      "Reason": "Single AZ",
      "Instances in Zone d": null
    },
    "status": "WARN",
    "resource_id": "arn:aws:elasticloadbalancing:eu-central-1:123456789012:loadbalancer/my-
load-balancer",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```

Exposed Access Keys

```
{
  "version": "0",
  "id": "1234abcd-ab12-123a-123a-1234567890ab",
  "detail-type": "Trusted Advisor Check Item Refresh Notification",
  "source": "aws.trustedadvisor",
  "account": "123456789012",
  "time": "2018-01-12T19:38:24Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "check-name": "Exposed Access Keys",
    "check-item-detail": {
      "Case ID": "12345678-1234-1234-abcd-1234567890ab",
      "Usage (USD per Day)": "0",
      "User Name (IAM or Root)": "my-username",
      "Deadline": "1440453299248",
```

```
      "Access Key ID": "AKIAIOSFODNN7EXAMPLE",
      "Time Updated": "1440021299248",
      "Fraud Type": "Exposed",
      "Location": "www.example.com"
    },
    "status": "ERROR",
    "resource_id": "",
    "uuid": "aa12345f-55c7-498e-b7ac-123456789012"
  }
}
```

Événements WorkSpaces

Pour plus d'informations sur les événements WorkSpaces, consultez Surveillance de vos instances WorkSpaces à l'aide de CloudWatch Events dans le Guide d'administration Amazon WorkSpaces.

Événements remis via CloudTrail

Vous pouvez également utiliser CloudWatch Events avec des services qui n'émettent pas d'événements et ne figurent pas dans cette page. AWS CloudTrail est un service qui enregistre automatiquement des événements tels que des appels d'API AWS. Vous pouvez créer des règles CloudWatch Events qui se déclenchent sur les informations capturées par CloudTrail. Pour plus d'informations sur CloudTrail, consultez Qu'est-ce qu'AWS CloudTrail ?. Pour plus d'informations sur la création d'une règle CloudWatch Events qui utilise CloudTrail, consultez Création d'une règle CloudWatch Events qui se déclenche lors d'un appel d'API AWS à l'aide de AWS CloudTrail (p. 8).

Tous les événements fournis via CloudTrail ont `AWS API Call via CloudTrail` comme valeur pour `detail-type`.

Certaines occurrences dans AWS peuvent être rapportées à CloudWatch Events par le service lui-même et par CloudTrail, mais de différentes façons. Par exemple, un appel d'API Amazon EC2 qui lance ou résilie une instance génère des événements disponibles à CloudWatch Events via CloudTrail. Toutefois, les changements d'état de l'instance Amazon EC2, de « en cours d'exécution » à « en cours d'arrêt », par exemple, sont des événements CloudWatch Events en eux-mêmes.

Voici un exemple d'un événement livré via CloudTrail. L'événement a été généré par un appel d'API AWS à Amazon S3 afin de créer un compartiment.

```
{
    "version": "0",
    "id": "36eb8523-97d0-4518-b33d-ee3579ff19f0",
    "detail-type": "AWS API Call via CloudTrail",
    "source": "aws.s3",
    "account": "123456789012",
    "time": "2016-02-20T01:09:13Z",
    "region": "us-east-1",
    "resources": [],
    "detail": {
        "eventVersion": "1.03",
        "userIdentity": {
            "type": "Root",
            "principalId": "123456789012",
            "arn": "arn:aws:iam::123456789012:root",
            "accountId": "123456789012",
            "sessionContext": {
                "attributes": {
                    "mfaAuthenticated": "false",
```

```
                    "creationDate": "2016-02-20T01:05:59Z"
                }
            }
        },
        "eventTime": "2016-02-20T01:09:13Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "CreateBucket",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "100.100.100.100",
        "userAgent": "[S3Console/0.4]",
        "requestParameters": {
            "bucketName": "bucket-test-iad"
        },
        "responseElements": null,
        "requestID": "9D767BCC3B4E7487",
        "eventID": "24ba271e-d595-4e66-a7fd-9c16cbf8abae",
        "eventType": "AwsApiCall"
    }
}
```


AWSLes événements d'appel d'API qui font plus de 256 Ko ne sont pas pris en charge. Pour plus d'informations sur les appels d'API que vous pouvez utiliser comme déclencheurs pour les règles, consultez Services pris en charge par l'historique des événements CloudTrail.

Envoi et réception d'événements entre comptes AWS

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez configurer votre compte AWS pour envoyer des événements vers d'autres comptes AWS ou recevoir des événements à partir d'un autre compte. Ceci peut s'avérer utile si les comptes appartiennent à la même organisation, ou appartiennent à des organisations qui sont partenaires ou ont une relation similaire.

Si vous configurez votre compte pour envoyer ou recevoir des événements, vous spécifiez quels comptes AWS individuels peuvent envoyer des événements ou en recevoir des vôtres. Si vous utilisez la fonctionnalité AWS Organizations, vous pouvez spécifier une organisation et accorder l'accès à tous ses comptes. Pour de plus amples informations, veuillez consulter Présentation d'AWS Organizations dans le Guide de l'utilisateur AWS Organizations.

Le processus global se présente comme suit :

- Sur le compte récepteur, modifiez les autorisations du bus d'événement par défaut pour autoriser un ou plusieurs comptes AWS spécifiés (ou tous les comptes AWS) à envoyer des événements au compte récepteur.
- Sur le compte expéditeur, configurez une ou plusieurs règles comportant le bus d'événement par défaut du compte récepteur en tant que cible.

  Si le compte expéditeur est autorisé à envoyer des événements parce qu'il fait partie d'une organisation AWS qui dispose des autorisations, le compte expéditeur doit également avoir un rôle IAM avec des politiques qui lui permettent d'envoyer des événements au compte récepteur. Si vous utilisez la AWS Management Console pour créer la règle qui cible le compte récepteur, cette opération est effectuée automatiquement. Si vous utilisez la AWS CLI, vous devez créer le rôle manuellement.
- Sur le compte récepteur, configurez une ou plusieurs des règles qui correspondent à des événements provenant du compte expéditeur.

La région AWS dans laquelle le compte récepteur ajoute des autorisations au bus d'événement par défaut doit être la même région que celle où le compte expéditeur crée la règle pour envoyer des événements au compte récepteur.

Les événements envoyés d'un compte à un autre sont facturés au compte expéditeur en tant qu'événements personnalisés. Le compte récepteur n'est pas facturé. Pour plus d'informations, consultez Tarification Amazon CloudWatch.

Si un compte récepteur configure une règle qui envoie des événements reçus d'un compte expéditeur à un troisième compte, ces événements ne sont pas envoyés au troisième compte.

Activation de votre AWScompte pour recevoir des événements d'autres comptes AWS

Pour recevoir des événements d'autres comptes ou organisations, vous devez d'abord modifier les autorisations sur le bus d'événement par défaut de votre compte. Le bus d'événement par défaut accepte les événements de services AWS, d'autres comptes AWS autorisés et des appels `PutEvents`.

Lorsque vous modifiez les autorisations sur votre bus d'événement par défaut pour accorder une autorisation à d'autres comptes AWS, vous pouvez spécifier les comptes par ID de compte ou d'organisation. Ou vous pouvez choisir de recevoir les événements de tous les comptes AWS.

> Warning
>
> Si vous choisissez de recevoir des événements de tous les comptes AWS, veillez à créer des règles qui correspondent uniquement aux événements à recevoir des autres. Pour créer des règles plus sûres, veillez à ce que le modèle d'événement de chaque règle contienne un champ `Account` avec l'ID de compte d'un ou de plusieurs comptes à partir desquels recevoir des événements. Les règles qui ont un modèle d'événement contenant un champ Account ne correspondent pas aux événements envoyés à partir des comptes qui ne sont pas répertoriés dans le champ `Account`. Pour de plus amples informations, veuillez consulter Modèles d'événements dans CloudWatch Events (p. 38).

Pour permettre à votre compte de recevoir les événements d'autres comptes AWS à l'aide de la console

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Event Buses, Add Permission.
3. Sélectionnez AWS Account or Organization (Compte ou organisation AWS).

   Si vous choisissez AWS Account (Compte AWS), saisissez l'ID de compte AWS à 12 chiffres du compte à partir duquel recevoir des événements. Pour recevoir des événements de tous les autres comptes AWS, choisissez Everybody(*).

   Si vous choisissez Organisation, choisissez Mon organisation pour accorder des autorisations à tous les comptes de l'organisation dont le compte actuel est membre. Ou choisissez Une autre organisation et entrez son ID d'organisation. Vous devez inclure le préfixe `o-` lorsque vous saisissez l'ID d'organisation.
4. Choisissez Add (Ajouter).
5. Vous pouvez répéter ces étapes pour ajouter d'autres comptes ou organisations.

Pour permettre à votre compte de recevoir les événements d'autres comptes AWS à l'aide de l'AWS CLI

1. Pour permettre à un compte AWS spécifique d'envoyer des événements, exécutez la commande suivante :

   ```
   aws events put-permission --action events:PutEvents --statement-id MySid --
   principal SenderAccountID
   ```

   Pour permettre à une organisation AWS d'envoyer des événements, exécutez la commande suivante :

   ```
   aws events put-permission --action events:PutEvents --statement-id MySid
                    --principal \* --condition '{"Type" : "StringEquals", "Key":
    "aws:PrincipalOrgID", "Value": "SenderOrganizationID"}'
   ```

Pour permettre à tous les autres comptes AWS d'envoyer des événements, exécutez la commande suivante :

```
aws events put-permission --action events:PutEvents --statement-id MySid --principal \*
```

Vous pouvez exécuter `aws events put-permission` plusieurs fois pour accorder des autorisations à des organisations et à des comptes AWS individuels, mais vous ne pouvez pas spécifier à la fois un compte individuel et une organisation dans une même commande.

2. Après avoir défini des autorisations pour votre bus d'événement par défaut, vous pouvez le cas échéant utiliser la commande `describe-event-bus` pour vérifier les autorisations :

```
aws events describe-event-bus
```

Envoi d'événements à un autre compte AWS

Pour envoyer des événements à un autre compte, configurez une règle CloudWatch Events qui comporte le bus d'événement par défaut d'un autre compte AWS comme cible. Le bus d'événements par défaut du compte récepteur doit également être configuré pour recevoir des événements de votre compte.

Pour envoyer des événements de votre compte vers un autre compte AWS à l'aide de la console

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Événements, Créer une règle.
3. Pour Source de l'événement, choisissez Modèle d'événement et sélectionnez le nom du service et les types d'événement à envoyer à l'autre compte.
4. Sélectionnez Add Target.
5. Pour Target (Cible), sélectionnez Event bus in another AWS account. (Bus d'événement dans un autre compte AWS). Pour ID de compte, saisissez l'ID de compte à 12 chiffres du compte AWS auquel envoyer des événements.
6. Un rôle IAM est nécessaire lorsque ce compte expéditeur est autorisé à envoyer des événements, car le compte récepteur a accordé des autorisations pour l'intégralité d'une organisation.

   - Pour créer un rôle IAM automatiquement, choisissez Create a new role for this specific resource.
   - Sinon, choisissez Utiliser le rôle existant. Choisissez un rôle possédant déjà les autorisations nécessaires pour appeler la version. CloudWatch Events n'accorde pas d'autorisations supplémentaires au rôle que vous sélectionnez.
7. En bas de la page, choisissez Configure Details.
8. Tapez un nom et une description pour la règle, puis choisissez Create Rule.

Pour envoyer des événements à un autre compte AWS à l'aide de l'AWS CLI

1. Si le compte expéditeur est autorisé à envoyer des événements parce qu'il fait partie d'une organisation AWS à laquelle le compte récepteur a accordé des autorisations, le compte expéditeur doit également avoir un rôle avec des stratégies qui lui permettent d'envoyer des événements au compte récepteur. Cette étape explique comment créer ce rôle.

   Si le compte expéditeur a reçu l'autorisation d'envoyer des événements au moyen de son ID de compte AWS et non via une organisation, cette étape est facultative. Vous pouvez passer à l'étape 2.

a. Si le compte expéditeur a été accordé des autorisations via une organisation, créez le rôle IAM nécessaire. Pour commencer, créez un fichier nommé `assume-role-policy-document.json` avec le contenu suivant :

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

b. Pour créer le rôle, entrez la commande suivante :

```
$ aws iam create-role \
--profile sender \
--role-name event-delivery-role \
--assume-role-policy-document file://assume-role-policy-document.json
```

c. Créez un fichier nommé `permission-policy.json` avec le contenu suivant:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:${receiver_account_id}:event-bus/default"
      ]
    }
  ]
}
```

d. Entrez la commande suivante pour associer cette stratégie au rôle :

```
$ aws iam put-role-policy \
--profile sender \
--role-name event-delivery-role \
--policy-name EventBusDeliveryRolePolicy
--policy-document file://permission-policy.json
```

2. Utilisez la commande `put-rule` pour créer une règle qui correspond aux types d'événement à envoyer à l'autre compte.
3. Ajoutez le bus d'événement par défaut de l'autre compte comme cible de la règle.

   Si le compte expéditeur a reçu des autorisations pour envoyer des événements par son ID de compte, la spécification d'un rôle n'est pas nécessaire. Exécutez la commande suivante :

```
aws events put-targets --rule NameOfRuleMatchingEventsToSend --targets
 "Id"="MyId","Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/default"
```

Si le compte expéditeur a reçu des autorisations pour envoyer des événements par son organisation, spécifiez un rôle, comme dans l'exemple suivant :

```
aws events put-targets --rule NameOfRuleMatchingEventsToSend --targets
 "Id"="MyId","Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/
default","RoleArn"="arn:aws:iam:${sender_account_id}:role/event-delivery-role"
```

Écriture de règles qui correspondent à des événements d'un autre compte AWS

Si votre compte est configuré pour recevoir des événements d'autres comptes AWS, vous pouvez écrire des règles qui correspondent à ces événements. Définissez le modèle d'événement de la règle pour correspondre aux événements que vous recevez de l'autre compte.

Sauf si vous spécifiez `account` dans le modèle d'événement d'une règle, toutes les règles de votre compte, les nouvelles et les existantes, qui correspondent à des événements que vous recevez d'autres comptes se déclenchent en fonction de ces événements. Si vous recevez des événements d'un autre compte, et que vous souhaitez qu'une règle se déclenche uniquement sur ce modèle d'événement lorsqu'elle est générée à partir de votre propre compte, vous devez ajouter `account` et spécifier l'ID de votre propre compte dans le modèle d'événement de la règle.

Si vous configurez votre compte AWS pour accepter des événements de tous les comptes AWS, nous vous recommandons vivement d'ajouter `account` à chaque règle CloudWatch Events dans votre compte. Ceci évite que les règles de votre compte se déclenchent sur des événements de comptes AWS inconnus. Lorsque vous spécifiez le champ `account` dans la règle, vous pouvez spécifier les ID de compte de plusieurs comptes AWS dans le champ.

Pour qu'une règle se déclenche sur un événement correspondant à partir de n'importe quel compte AWS auquel vous avez accordé des autorisations, ne spécifiez pas * dans le champ `account` de la règle. En effet, la règle ne correspondrait à aucun événement, car * n'apparaît jamais dans le champ `account` d'un événement. Au lieu de cela, contentez-vous d'omettre le champ `account` à partir de la règle.

Pour écrire une règle correspondant à des événements d'un autre compte à l'aide de la console

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, choisissez Événements, Créer une règle.
3. Pour Event Source, choisissez Event Pattern, puis sélectionnez le nom du service et les types d'événement auxquels la règle doit correspondre.
4. En regard de Event Pattern Preview, choisissez Edit.
5. Dans la fenêtre de modification, ajoutez une ligne `Account` qui spécifie que les comptes AWS qui envoient cet événement doivent correspondre à la règle. Par exemple, si la fenêtre de modification montre ce qui suit à l'origine :

```
{
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ]
}
```

Ajoutez ce qui suit pour que la règle corresponde aux notifications de volume EBS envoyées par les comptes AWS 123456789012 et 111122223333 :

```
{
  "account": [
    "123456789012","111122223333"
  ],
  "source": [
    "aws.ec2"
  ],
  "detail-type": [
    "EBS Volume Notification"
  ]
}
```

6. Après avoir modifié le modèle d'événement, choisissez Enregistrer.
7. Terminez de créer la règle comme d'habitude, en définissant une ou plusieurs cibles dans votre compte.

Pour écrire une règle correspondant à des événements d'un autre compte AWS à l'aide de l'AWS CLI

- Utilisez la commande `put-rule`. Dans le champ `Account` du modèle d'événement de la règle, spécifiez les autres comptes AWS qui doivent correspondre à la règle. L'exemple de règle suivant correspond aux modifications d'état d'instance Amazon EC2 dans les comptes AWS 123456789012 et 111122223333 :

```
aws events put-rule --name "EC2InstanceStateChanges" --event-pattern "{\"account\":
["123456789012", "111122223333"],\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2
 Instance State-change Notification\"]}"  --role-arn "arn:aws:iam::123456789012:role/
MyRoleForThisRule"
```

Migrer une relation expéditeur à destinataire pour utiliser AWS Organizations

Si vous disposez d'un compte expéditeur doté d'autorisations octroyées directement à son ID de compte, et que vous souhaitez révoquer ces autorisations et accorder au compte expéditeur l'accès en octroyant les autorisations à une organisation, vous devez effectuer des étapes supplémentaires. Ces étapes vous permettent de vous assurer que les événements provenant du compte expéditeur peuvent toujours parvenir au compte destinataire. En effet, les comptes autorisés à envoyer des événements via une organisation doivent également utiliser un rôle IAM pour cela.

Pour ajouter les autorisations nécessaires afin de migrer une relation expéditeur à destinataire

1. Dans le compte expéditeur, créez un rôle IAM avec des politiques lui permettant d'envoyer des événements au compte destinataire.

   a. Créez un fichier nommé `assume-role-policy-document.json`, avec le contenu suivant :

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

```
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

b. Pour créer le rôle IAM, saisissez la commande suivante :

```
$ aws iam create-role \
--profile sender \
--role-name event-delivery-role \
--assume-role-policy-document file://assume-role-policy-document.json
```

c. Créez un fichier nommé `permission-policy.json` avec le contenu suivant:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:${receiver_account_id}:event-bus/default"
      ]
    }
  ]
}
```

d. Entrez la commande suivante pour associer cette stratégie au rôle :

```
$ aws iam put-role-policy \
--profile sender \
--role-name event-delivery-role \
--policy-name EventBusDeliveryRolePolicy
--policy-document file://permission-policy.json
```

2. Modifiez chaque règle existante dans le compte expéditeur comportant le bus d'événement par défaut du compte destinataire en tant que cible. Modifiez la règle en ajoutant le rôle créé à l'étape 1 aux informations cibles. Utilisez la commande suivante:

```
aws events put-targets --rule Rulename --targets
 "Id"="MyID","Arn"="arn:aws:events:region:$ReceiverAccountID:event-bus/
default","RoleArn"="arn:aws:iam:${sender_account_id}:role/event-delivery-role"
```

3. Dans le compte destinataire, exécutez la commande suivante pour accorder des autorisations pour les comptes de l'organisation afin d'envoyer des événements au compte destinataire :

```
aws events put-permission --action events:PutEvents --statement-id Sid-For-Organization
 --principal \* --condition '{"Type" : "StringEquals", "Key": "aws:PrincipalOrgID",
 "Value": "SenderOrganizationID"}'
```

Vous pouvez également révoquer les autorisations accordées initialement directement au compte expéditeur :

```
aws events remove-permission --statement-id Sid-for-SenderAccount
```

Ajout d'événements avec PutEvents

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

L'action `PutEvents` envoie plusieurs événements à CloudWatch Events dans une seule et même demande. Pour de plus amples informations, consultez PutEvents dans la documentation de référence Amazon CloudWatch Events et put-events dans la documentation de référence des commandes de AWS CLI.

Chaque demande `PutEvents` peut prendre en charge un nombre d'entrées limité. Pour de plus amples informations, veuillez consulter Quotas CloudWatch Events (p. 116). L'opération `PutEvents` tente de traiter toutes les entrées dans l'ordre naturel de la demande. Chaque événement a un identifiant unique qui est attribué par CloudWatch Events après avoir appelé `PutEvents`.

Dans l'exemple suivant, le code Java envoie deux événements identiques à CloudWatch Events :

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

PutEventsRequest request = new PutEventsRequest()
        .withEntries(requestEntry, requestEntry);

PutEventsResult result = awsEventsClient.putEvents(request);

for (PutEventsResultEntry resultEntry : result.getEntries()) {
    if (resultEntry.getEventId() != null) {
        System.out.println("Event Id: " + resultEntry.getEventId());
    } else {
        System.out.println("Injection failed with Error Code: " +
 resultEntry.getErrorCode());
    }
}
```

Le résultat `PutEvents` comprend un tableau d'entrées de réponse. Chaque entrée de ce tableau correspond directement à une entrée du tableau de demande utilisant l'ordre naturel, depuis le haut vers le bas de la demande et de la réponse. Le tableau de réponse `Entries` comprend toujours le même nombre d'entrées que le tableau de demande.

Gestion des défaillances lors de l'utilisation de PutEvents

Par défaut, les défaillances d'entrées individuelles d'une demande n'arrêtent pas le traitement des entrées suivantes dans la demande. Cela signifie qu'un tableau d'entrées de réponse comprend des entrées

traitées avec et sans succès. Vous devez détecter les entrées traitées sans succès et les inclure dans un appel ultérieur.

Les entrées traitées avec succès incluent une valeur d'ID, tandis que celles traitées sans succès incluent les valeurs `ErrorCode` et `ErrorMessage`. Le paramètre `ErrorCode` reflète le type d'erreur. `ErrorMessage` fournit des informations plus détaillées sur l'erreur. L'exemple ci-dessous comporte trois entrées de résultat pour une demande `PutEvents`. La deuxième entrée a échoué, ce qui se reflète dans la réponse.

Exemple : syntaxe de réponse PutEvents

```
{
    "FailedEntryCount": 1,
    "Entries": [
        {
            "EventId": "11710aed-b79e-4468-a20b-bb3c0c3b4860"
        },
        {   "ErrorCode": "InternalFailure",
            "ErrorMessage": "Internal Service Failure"
        },
            "EventId": "d804d26a-88db-4b66-9eaf-9a11c708ae82"
        }
    ]
}
```

Les entrées qui ont été traitées sans succès peuvent être incluses dans les demandes `PutEvents` ultérieures. Tout d'abord, vérifiez le paramètre `FailedRecordCount` de `PutEventsResult` afin de savoir si la demande comporte des enregistrements d'échecs. Si c'est bien le cas, chaque `Entry` comportant un `ErrorCode` non nul doit être ajoutée à une demande suivante. Pour un exemple de ce type de gestionnaire, reportez-vous au code suivant.

Exemple : gestionnaire de défaillances PutEvents

```
PutEventsRequestEntry requestEntry = new PutEventsRequestEntry()
        .withTime(new Date())
        .withSource("com.mycompany.myapp")
        .withDetailType("myDetailType")
        .withResources("resource1", "resource2")
        .withDetail("{ \"key1\": \"value1\", \"key2\": \"value2\" }");

List<PutEventsRequestEntry> putEventsRequestEntryList = new ArrayList<>();
for (int i = 0; i < 3; i++) {
    putEventsRequestEntryList.add(requestEntry);
}

PutEventsRequest putEventsRequest = new PutEventsRequest();
putEventsRequest.withEntries(putEventsRequestEntryList);
PutEventsResult putEventsResult = awsEventsClient.putEvents(putEventsRequest);

while (putEventsResult.getFailedEntryCount() > 0) {
    final List<PutEventsRequestEntry> failedEntriesList = new ArrayList<>();
    final List<PutEventsResultEntry> PutEventsResultEntryList =
 putEventsResult.getEntries();
    for (int i = 0; i < PutEventsResultEntryList.size(); i++) {
        final PutEventsRequestEntry putEventsRequestEntry =
 putEventsRequestEntryList.get(i);
        final PutEventsResultEntry putEventsResultEntry = PutEventsResultEntryList.get(i);
        if (putEventsResultEntry.getErrorCode() != null) {
            failedEntriesList.add(putEventsRequestEntry);
        }
    }
    putEventsRequestEntryList = failedEntriesList;
```

```
    putEventsRequest.setEntries(putEventsRequestEntryList);
    putEventsResult = awsEventsClient.putEvents(putEventsRequest);
    }
```

Envoi d'événements à l'aide de l' AWS CLI

Vous pouvez utiliser l'AWS CLI pour envoyer des événements personnalisés. L'exemple suivant place un événement personnalisé dans CloudWatch Events :

```
aws events put-events \
--entries '[{"Time": "2016-01-14T01:02:03Z", "Source": "com.mycompany.myapp", "Resources":
 ["resource1", "resource2"], "DetailType": "myDetailType", "Detail": "{ \"key1\":
 \"value1\", \"key2\": \"value2\" }"}]'
```

Vous pouvez également créer un fichier, par exemple entries.json comme ceci :

```
[
  {
    "Time": "2016-01-14T01:02:03Z",
    "Source": "com.mycompany.myapp",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }"
  }
]
```

Vous pouvez utiliser l'AWS CLI pour lire les entrées de ce fichier et envoyer des événements. À l'invite de commande, entrez :

```
aws events put-events --entries file://entries.json
```

Calcul de la taille des entrées d'événements PutEvents

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez injecter des événements personnalisés dans CloudWatch Events au moyen de l'action `PutEvents`. Vous pouvez injecter plusieurs événements à l'aide de l'action `PutEvents` tant que la taille totale des entrées est inférieure à 256 Ko. Vous pouvez calculer au préalable la taille des entrées d'événements, comme indiqué ci-dessous. Vous pouvez ensuite regrouper plusieurs entrées d'événement en une seule demande pour plus d'efficacité.

> **Note**
>
> La limite de taille est imposée à l'entrée. Même si l'entrée est inférieure à la limite de taille, cela ne signifie pas que l'événement dans CloudWatch Events est également inférieur à cette taille. Au contraire, la taille de l'événement est toujours supérieure à la taille de l'entrée en raison des caractères nécessaires et des clés de la représentation JSON de l'événement. Pour de plus amples informations, veuillez consulter Modèles d'événements dans CloudWatch Events (p. 38).

La taille de `PutEventsRequestEntry` est calculée de la manière suivante :

- Si le paramètre `Time` est spécifié, sa taille est de 14 octets.
- La taille des paramètres `Source` et `DetailType` est égale au nombre d'octets de leur forme UTF-8 codée.
- Si le paramètre `Detail` est spécifié, sa taille est égale au nombre d'octets de sa forme UTF-8 codée.
- Si le paramètre `Resources` est spécifié, la taille de chaque entrée est égale au nombre d'octets de sa forme UTF-8 codée.

Le code Java de l'exemple suivant calcule la taille d'un objet `PutEventsRequestEntry` donné :

```
int getSize(PutEventsRequestEntry entry) {
    int size = 0;
    if (entry.getTime() != null) {
        size += 14;
    }
    size += entry.getSource().getBytes(StandardCharsets.UTF_8).length;
    size += entry.getDetailType().getBytes(StandardCharsets.UTF_8).length;
    if (entry.getDetail() != null) {
        size += entry.getDetail().getBytes(StandardCharsets.UTF_8).length;
    }
    if (entry.getResources() != null) {
        for (String resource : entry.getResources()) {
            if (resource != null) {
                size += resource.getBytes(StandardCharsets.UTF_8).length;
            }
        }
    }
    return size;
}
```

Utilisation de CloudWatch Events avec des points de terminaison de VPC d'interface

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Si vous utilisez Amazon Virtual Private Cloud (Amazon VPC) pour héberger vos ressources AWS, vous pouvez établir une connexion privée entre votre VPC et CloudWatch Events. Vous pouvez utiliser cette connexion pour permettre à CloudWatch Events de communiquer avec vos ressources sur votre VPC sans passer par le réseau Internet public.

Amazon VPC est un service AWS que vous pouvez utiliser pour lancer des ressources AWS 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 Events, vous définissez un point de terminaison de VPC d'interface pour CloudWatch Events. Ce type de point de terminaison vous permet de connecter votre VPC à des services AWS. Le point de terminaison assure une connectivité scalable et fiable à CloudWatch Events, sans qu'une passerelle Internet, une instance NAT (Network Address Translation) ou une connexion VPN ne soit nécessaire. 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 de VPC d'interface reposent sur AWS PrivateLink, une technologie AWS qui active une communication privée entre les services AWS à l'aide d'une interface réseau Elastic avec des adresses IP privées. Pour de plus amples informations, veuillez consulter Nouveauté : AWS PrivateLink pour les services AWS.

Les étapes suivantes s'adressent aux utilisateurs d'Amazon VPC. Pour plus d'informations, consultez Démarrez dans le Amazon VPC Guide de l'utilisateur.

Availability

CloudWatch Events prend actuellement en charge les points de terminaison de VPC dans les régions suivantes :

- US East (Ohio)
- US East (N. Virginia)
- US West (N. California)
- US West (Oregon)
- Asia Pacific (Mumbai)

- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Paris)
- South America (São Paulo)

Création du point de terminaison de VPC pour CloudWatch Events

Pour commencer à utiliser CloudWatch Events avec votre VPC, créez un point de terminaison de VPC d'interface pour CloudWatch Events. Le nom de service à choisir est com.amazonaws.***Region***.events. Pour plus d'informations, veuillez consulter Création d'un point de terminaison d'interface dans le Amazon VPC Guide de l'utilisateur.

Vous n'avez pas besoin de modifier les paramètres pour CloudWatch Events. CloudWatch Events appelle d'autres services AWS à l'aide de points de terminaison publics ou de points de terminaison de VPC d'interface privés, selon ceux utilisés. Par exemple, si vous créez un point de terminaison VPC d'interface pour CloudWatch Events, et que vous avez déjà une règle CloudWatch Events qui envoie des notifications à Amazon SNS lorsqu'elle est déclenchée, les notifications commencent à transiter via le point de terminaison VPC d'interface.

Contrôle de l'accès à votre point de terminaison de VPC CloudWatch Events

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, nous lui attachons une stratégie par défaut pour vous qui autorise un accès total au service. Une politique de point de terminaison n'annule pas et ne remplace pas les politiques utilisateur IAM ou les politiques propres au service. Il s'agit d'une stratégie distincte qui contrôle l'accès depuis le point de terminaison jusqu'au service spécifié.

Les stratégies de point de terminaison doivent être écrites au format JSON.

Pour de plus amples informations, veuillez consulter Contrôle de l'accès aux services avec des points de terminaison d'un VPC dans le Amazon VPC Guide de l'utilisateur.

Voici un exemple de politique de point de terminaison pour CloudWatch Events. Cette politique permet aux utilisateurs de se connecter à CloudWatch Events via le VPC pour envoyer des journaux à CloudWatch Events, et les empêche d'effectuer d'autres actions CloudWatch Events.

```
{
  "Statement": [
    {
      "Sid": "PutOnly",
```

```
      "Principal": "*",
      "Action": [
        "events:PutEvents"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Pour modifier la politique de point de terminaison de VPC pour CloudWatch Events

1. Ouvrez la console Amazon VPC à l'adresse https://console.aws.amazon.com/vpc/.
2. Dans le volet de navigation, choisissez Endpoints (Points de terminaison).
3. Si vous n'avez pas encore créé le point de terminaison pour CloudWatch Events, sélectionnez Create Endpoint (Créer un point de terminaison). Ensuite, sélectionnez com.amazonaws.***Region***.events et choisissez Create endpoint (Créer un point de terminaison).
4. Sélectionnez le point de terminaison com.amazonaws.***Region***.events, puis l'onglet Policy (Stratégie) dans la partie inférieure de l'écran.
5. Choisissez Edit Policy (Modifier la stratégie), puis apportez les modifications voulues à la stratégie.

Surveillance de l'utilisation à l'aide de métriques CloudWatch

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

CloudWatch Events envoie les métriques à Amazon CloudWatch toutes les minutes.

Métriques CloudWatch Events

L'espace de noms `AWS/Events` inclut les métriques suivantes.

Toutes ces métriques se basent sur l'unité Nombre ; Sum (Somme) et SampleCount (Exemple de comptage) constituent donc les statistiques les plus utiles.

Métrique	Description
`DeadLetterInvocations`	Mesure le nombre de fois où la cible d'une règle n'est pas appelée en réponse à un événement. Cela inclut les appels qui aboutiraient au déclenchement de la même règle à nouveau, entraînant une boucle infinie. Dimensions valides : RuleName Unités : nombre
`Invocations`	Mesure le nombre de fois où la cible est appelée pour une règle en réponse à un événement. Cela inclut les appels réussis ou en échec, mais n'inclut pas les tentatives limitées ou renouvelées jusqu'à ce qu'ils échouent en permanence. Cela n'inclut pas DeadLetterInvocations. **Note** CloudWatch Events n'envoie cette métrique à CloudWatch que si celle-ci a une valeur autre que zéro. Dimensions valides : RuleName Unités : nombre
`FailedInvocations`	Mesure le nombre d'appels qui ont échoué en permanence. Cela n'inclut pas les appels qui sont renouvelés ou qui ont abouti après une nouvelle tentative. Cela n'inclut pas non plus les appels ayant échoué qui sont comptés dans DeadLetterInvocations.

Métrique	Description
	Dimensions valides : RuleName Unités : nombre
`TriggeredRules`	Mesure le nombre de règles déclenchées qui correspondaient à un événement. Dimensions valides : RuleName Unités : nombre
`MatchedEvents`	Mesure le nombre d'événements qui correspondaient à une règle. Dimensions valides : aucune Unités : nombre
`ThrottledRules`	Mesure le nombre de règles déclenchées qui sont limitées. Dimensions valides : RuleName Unités : nombre

Dimensions pour les métriques CloudWatch Events

Les métriques CloudWatch Events ont une seule dimension qui figure ci-dessous.

Dimension	Description
`RuleName`	Filtre les métriques disponibles par nom de règle.

Règles gérées par Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

D'autres services AWS peuvent créer et gérer les règles CloudWatch Events de votre compte AWS nécessaires à certaines fonctions dans ces services. Celles-ci sont appelées des règles gérées.

Lorsqu'un service crée une règle gérée, il peut également créer une politique IAM qui accorde des autorisations à ce service pour créer la règle. Les politiques IAM créées de cette manière sont restreintes étroitement par des autorisations au niveau des ressources, afin d'autoriser uniquement la création des règles nécessaires.

Vous pouvez supprimer des règles gérées à l'aide de l'option Forcer la suppression. Ne le faites que si vous êtes sûr que l'autre service n'a plus besoin de la règle. Dans le cas contraire, la suppression d'une règle gérée entraîne l'arrêt des fonctionnalités qui s'appuient sur celle-ci.

Utilisation de CloudWatch Events avec un kit SDK d'AWS.

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

Documentation des kits SDK	Exemples de code
AWS SDK for C++	AWS SDK for C++ 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
AWS SDK for .NET	AWS SDK for .NET Exemples de code
AWS SDK for PHP	AWS SDK for PHP Exemples 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

Pour obtenir des exemples spécifiques à CloudWatch Events, consultez Exemples de code pour CloudWatch Events qui utilisent AWS kits SDK (p. 102).

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 Events qui utilisent AWS kits SDK

Les exemples de code suivants montrent comment utiliser CloudWatch Events avec un kit SDK d'AWS.

Les exemples sont répartis dans les catégories suivantes :

Actions

Des extraits de code qui vous montrent comment appeler des fonctions de service individuelles.

Pour obtenir la liste complète des guides de développement AWS SDK et des exemples de code, consultez Utilisation de CloudWatch Events avec un kit SDK d'AWS. (p. 101). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

Actions pour CloudWatch Events qui utilisent AWS kits SDK

Les exemples de code suivants montrent comment effectuer des actions CloudWatch Events individuelles avec les Kits AWS SDK. Ces extraits appellent l'API de CloudWatch Events et ne sont pas destinés à être exécutés isolément. Chaque exemple inclut un lien vers GitHub, où vous trouverez des instructions sur la configuration et l'exécution du code en contexte.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez les Références de l'API de CloudWatch Events.

Exemples

Ajouter une cible de fonction Lambda à l'aide d'un kit SDK d'AWS

Les exemples de code suivants montrent comment ajouter un cible de fonction AWS Lambda d'un événement Amazon CloudWatch Events.

Java

SDK pour Java 2.x

```
   public static void putCWTargets(CloudWatchEventsClient cwe, String ruleName,
String functionArn, String targetId ) {

       try {
           Target target = Target.builder()
               .arn(functionArn)
               .id(targetId)
               .build();

           PutTargetsRequest request = PutTargetsRequest.builder()
               .targets(target)
               .rule(ruleName)
               .build();

           PutTargetsResponse response = cwe.putTargets(request);
           System.out.printf(
               "Successfully created CloudWatch events target for rule %s",
               ruleName);
       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour plus d'informations sur l'API, consultez PutTargets dans AWS SDK for Java 2.x Référence de l'API.

JavaScript

SDK pour JavaScript V3

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchEventsClient } from "@aws-sdk/client-cloudwatch-events";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon CloudWatch service client object.
export const cweClient = new CloudWatchEventsClient({ region: REGION });
```

Importez le SDK et les modules client et appelez l'API.

```
// Import required AWS SDK clients and commands for Node.js
import { PutTargetsCommand } from "@aws-sdk/client-cloudwatch-events";
import { cweClient } from "./libs/cloudWatchEventsClient.js";

// Set the parameters
export const params = {
  Rule: "DEMO_EVENT",
  Targets: [
    {
      Arn: "LAMBDA_FUNCTION_ARN", //LAMBDA_FUNCTION_ARN
```

```
      Id: "myCloudWatchEventsTarget",
    },
  ],
};

export const run = async () => {
  try {
    const data = await cweClient.send(new PutTargetsCommand(params));
    console.log("Success, target added; requestID: ", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
// Uncomment this line to run execution within this file.
// run();
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus d'informations sur l'API, consultez PutTargets dans AWS SDK for JavaScript Référence de l'API.

Kit SDK pour JavaScript V2

```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'REGION'});

// Create CloudWatchEvents service object
var cwevents = new AWS.CloudWatchEvents({apiVersion: '2015-10-07'});

var params = {
  Rule: 'DEMO_EVENT',
  Targets: [
    {
      Arn: 'LAMBDA_FUNCTION_ARN',
      Id: 'myCloudWatchEventsTarget',
    }
  ]
};

cwevents.putTargets(params, function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data);
  }
});
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus d'informations sur l'API, consultez PutTargets dans AWS SDK for JavaScript Référence de l'API.

Pour obtenir la liste complète des guides de développement AWS SDK et des exemples de code, consultez Utilisation de CloudWatch Events avec un kit SDK d'AWS. (p. 101). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Créer une règle planifiée CloudWatch Events à l'aide d'un kit SDK d'AWS

Les exemples de code suivants montrent comment créer une règle planifiée Amazon CloudWatch Events.

Java

SDK pour Java 2.x

```
   public static void putCWRule(CloudWatchEventsClient cwe, String ruleName,
String roleArn) {

       try {
           PutRuleRequest request = PutRuleRequest.builder()
               .name(ruleName)
               .roleArn(roleArn)
               .scheduleExpression("rate(5 minutes)")
               .state(RuleState.ENABLED)
               .build();

           PutRuleResponse response = cwe.putRule(request);
           System.out.printf(
                   "Successfully created CloudWatch events rule %s with arn %s",
                   roleArn, response.ruleArn());
       } catch (
           CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour plus d'informations sur l'API, consultez PutRule dans AWS SDK for Java 2.x Référence de l'API.

JavaScript

SDK pour JavaScript V3

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchEventsClient } from "@aws-sdk/client-cloudwatch-events";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon CloudWatch service client object.
export const cweClient = new CloudWatchEventsClient({ region: REGION });
```

Importez le SDK et les modules client et appelez l'API.

```
// Import required AWS SDK clients and commands for Node.js
import { PutRuleCommand } from "@aws-sdk/client-cloudwatch-events";
import { cweClient } from "./libs/cloudWatchEventsClient.js";
```

```
// Set the parameters
export const params = {
  Name: "DEMO_EVENT",
  RoleArn: "IAM_ROLE_ARN", //IAM_ROLE_ARN
  ScheduleExpression: "rate(5 minutes)",
  State: "ENABLED",
};

export const run = async () => {
  try {
    const data = await cweClient.send(new PutRuleCommand(params));
    console.log("Success, scheduled rule created; Rule ARN:", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
// Uncomment this line to run execution within this file.
// run();
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus d'informations sur l'API, consultez PutRule dans AWS SDK for JavaScript Référence de l'API.

Kit SDK pour JavaScript V2

```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'REGION'});

// Create CloudWatchEvents service object
var cwevents = new AWS.CloudWatchEvents({apiVersion: '2015-10-07'});

var params = {
  Name: 'DEMO_EVENT',
  RoleArn: 'IAM_ROLE_ARN',
  ScheduleExpression: 'rate(5 minutes)',
  State: 'ENABLED'
};

cwevents.putRule(params, function(err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.RuleArn);
  }
});
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus d'informations sur l'API, consultez PutRule dans AWS SDK for JavaScript Référence de l'API.

Pour obtenir la liste complète des guides de développement AWS SDK et des exemples de code, consultez Utilisation de CloudWatch Events avec un kit SDK d'AWS. (p. 101). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Envoyer des événements CloudWatch Events à l'aide d'un kit SDK d'AWS

Les exemples de code suivants montrent comment envoyer des événements Amazon CloudWatch Events.

Java

SDK pour Java 2.x

```
   public static void putCWEvents(CloudWatchEventsClient cwe, String resourceArn )
{

       try {

           final String EVENT_DETAILS =
               "{ \"key1\": \"value1\", \"key2\": \"value2\" }";

           PutEventsRequestEntry requestEntry = PutEventsRequestEntry.builder()
                   .detail(EVENT_DETAILS)
                   .detailType("sampleSubmitted")
                   .resources(resourceArn)
                   .source("aws-sdk-java-cloudwatch-example")
                   .build();

           PutEventsRequest request = PutEventsRequest.builder()
                   .entries(requestEntry)
                   .build();

           cwe.putEvents(request);
           System.out.println("Successfully put CloudWatch event");

       } catch (CloudWatchException e) {
           System.err.println(e.awsErrorDetails().errorMessage());
           System.exit(1);
       }
   }
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour plus d'informations sur l'API, consultezPutEventsdansAWS SDK for Java 2.xRéférence de l'API.

JavaScript

SDK pour JavaScript V3

Créez le client dans un module séparé et exportez-le.

```
import { CloudWatchEventsClient } from "@aws-sdk/client-cloudwatch-events";
// Set the AWS Region.
const REGION = "REGION"; //e.g. "us-east-1"
// Create an Amazon CloudWatch service client object.
export const cweClient = new CloudWatchEventsClient({ region: REGION });
```

Importez le SDK et les modules client et appelez l'API.

```
// Import required AWS SDK clients and commands for Node.js
import { PutEventsCommand } from "@aws-sdk/client-cloudwatch-events";
import { cweClient } from "./libs/cloudWatchEventsClient.js";

// Set the parameters
export const params = {
  Entries: [
    {
      Detail: '{ "key1": "value1", "key2": "value2" }',
      DetailType: "appRequestSubmitted",
      Resources: [
        "RESOURCE_ARN", //RESOURCE_ARN
      ],
      Source: "com.company.app",
    },
  ],
};

export const run = async () => {
  try {
    const data = await cweClient.send(new PutEventsCommand(params));
    console.log("Success, event sent; requestID:", data);
    return data; // For unit tests.
  } catch (err) {
    console.log("Error", err);
  }
};
// Uncomment this line to run execution within this file.
// run();
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus d'informations sur l'API, consultezPutEventsdansAWS SDK for JavaScriptRéférence de l'API.

Kit SDK pour JavaScript V2

```
// Load the AWS SDK for Node.js
var AWS = require('aws-sdk');
// Set the region
AWS.config.update({region: 'REGION'});

// Create CloudWatchEvents service object
var cwevents = new AWS.CloudWatchEvents({apiVersion: '2015-10-07'});

var params = {
  Entries: [
    {
      Detail: '{ \"key1\": \"value1\", \"key2\": \"value2\" }',
      DetailType: 'appRequestSubmitted',
      Resources: [
        'RESOURCE_ARN',
      ],
      Source: 'com.company.app'
    }
  ]
};

cwevents.putEvents(params, function(err, data) {
  if (err) {
    console.log("Error", err);
```

```
  } else {
    console.log("Success", data.Entries);
  }
});
```

- Trouvez des instructions et plus de code sur GitHub.
- Pour de plus amples informations, consultez le AWS SDK for JavaScript Guide du développeur.
- Pour plus d'informations sur l'API, consultezPutEventsdansAWS SDK for JavaScriptRéférence de l'API.

Pour obtenir la liste complète des guides de développement AWS SDK et des exemples de code, consultez Utilisation de CloudWatch Events avec un kit SDK d'AWS. (p. 101). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Sécurité d'Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Pour plus d'informations sur la sécurité de CloudWatch Events, consultez Sécurité dans Amazon EventBridge.

Baliser vos ressources Amazon CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Une balise est un attribut personnalisé que vous conférez ou que AWS attribue à une ressource AWS. 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 :

- Identifier et organiser vos ressources AWS. De nombreux services AWS prennent en charge le balisage. Vous pouvez donc attribuer la même balise à des ressources à partir de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à une règle CloudWatch Events que vous affectez à une instance EC2.
- Suivre vos coûts AWS. Vous activez ces balises sur le tableau de bord AWS Billing and Cost Management. AWS utilise les balises pour classer vos coûts et pour vous fournir un rapport mensuel d'allocation des coûts. Pour de plus amples informations, veuillez consulter Utilisation des balises d'allocation des coûts dans le Guide de l'utilisateur AWS Billing.

Les sections suivantes fournissent de plus amples informations sur les balises pour CloudWatch Events.

Ressources prises en charge dans CloudWatch Events

Les ressources suivantes dans CloudWatch Events prennent en charge le balisage :

- Règles

Pour obtenir des informations sur l'ajout et la gestion de balises, veuillez consulter Gestion des balises (p. 112).

Gestion des balises

Les balises comprennent les propriétés `Key` et `Value` d'une ressource. Vous pouvez utiliser la console CloudWatch, la AWS CLI ou l'API CloudWatch Events pour ajouter, modifier ou supprimer les valeurs de ces propriétés. Pour obtenir plus d'informations sur l'utilisation des balises, consultez ce qui suit :

- TagResource, UntagResource et ListTagsForResource dans la documentation de référence d'API Amazon CloudWatch Events
- tag-resource, untag-resource et list-tags-for-resource dans la documentation de référence de la CLI Amazon CloudWatch
- Utilisation de l'éditeur de balises dans le Guide de l'utilisateur Resource Groups

Conventions de dénomination et d'utilisation de balises

Les conventions d'utilisation et d'attribution de noms de base suivantes s'appliquent à l'utilisation des balises avec les ressources CloudWatch Events :

- 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 préfixe `aws:` est interdit pour les balises, car il est réservé à l'utilisation d'AWS. Vous ne pouvez pas modifier ni supprimer des clés ou valeurs de balise ayant ce préfixe. Les balises avec ce préfixe ne sont pas comptabilisées dans votre quota de balises par ressource.

Journalisation des appels d'API Amazon CloudWatch Events avec AWS CloudTrail

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Amazon CloudWatch Events est intégré à AWS CloudTrail, un service qui enregistre les actions effectuées par un utilisateur, un rôle ou un service AWS dans CloudWatch Events. CloudTrail capture les appels d'API effectués par votre compte AWS ou au nom de ce dernier. Les appels capturés incluent les appels de la console CloudWatch et les appels de code aux opérations de l'API CloudWatch Events. Si vous créez un journal journal d'activité, vous pouvez activer la livraison continue des événements CloudTrail dans un compartiment Amazon S3, y compris des événements pour CloudWatch Events. Si vous ne configurez pas de journal de suivi, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). Avec les informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à CloudWatch Events, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur et la date de la demande, ainsi que d'autres détails.

Pour en savoir plus sur CloudTrail, y compris la façon de le configurer et de l'activer, consultez le Guide de l'utilisateur AWS CloudTrail.

Rubriques

Informations CloudWatch Events dans CloudTrail

CloudTrail est activé dans votre compte AWS lors de la création de ce dernier. Quand une activité d'événement prise en charge a lieu dans CloudWatch Events, elle est enregistrée dans un événement CloudTrail avec d'autres événements de service AWS dans Event history (Historique des événements). Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour de plus amples informations, veuillez consulter Affichage des événements avec l'historique des événements CloudTrail.

Pour un enregistrement continu des événements dans votre compte AWS, notamment les événements CloudWatch Events, créez un journal d'activité. Un journal de suivi permet à CloudTrail de livrer des fichiers journaux dans un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS. Le journal d'activité consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre

choix. En outre, vous pouvez configurer d'autres services AWS pour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- Intégrations et services pris en charge par CloudTrail
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers journaux CloudTrail de plusieurs régions et Réception de fichiers journaux CloudTrail de plusieurs comptes

CloudWatch Events prend en charge la journalisation des actions suivantes en tant qu'événements dans des fichiers journaux CloudTrail :

- DeleteRule
- DescribeEventBus
- DescribeRule
- DisableRule
- EnableRule
- ListRuleNamesByTarget
- ListRules
- ListTargetsByRule
- PutPermission
- PutRule
- PutTargets
- RemoveTargets
- TestEventPattern

Chaque événement ou entrée du journal contient des informations sur la personne qui a généré la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec des informations d'identification utilisateur racine ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour de plus amples informations, consultez l'élément userIdentity CloudTrail.

Exemple : entrées du fichier journal CloudWatch Events

Un journal de suivi est une configuration qui permet la livraison d'événements sous forme de fichiers journaux vers un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux CloudTrail peuvent contenir une ou plusieurs entrées de journal. Un événement représente une demande individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne constituent pas une série ordonnée retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'entrée du fichier journal CloudTrail suivante montre qu'un utilisateur a appelé l'action CloudWatch Events PutRule.


```
{
         "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"
}
```

Quotas CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Pour plus d'informations sur les quotas de service CloudWatch Events et EventBridge, veuillez consulter Quotas Amazon EventBridge.

Pour plus d'informations, consultez les rubriques suivantes.

- Amazon EventBridge
- Quotas de service EventBridge
- Documentation de référence d'API Amazon EventBridge

Résolution des problèmes CloudWatch Events

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Vous pouvez utiliser les étapes indiquées dans cette section pour résoudre les problèmes CloudWatch Events.

Rubriques

Ma règle a été déclenchée, mais ma fonction Lambda n'a pas été appelée

Assurez-vous de disposer des autorisations appropriées pour votre fonction Lambda. Exécutez la commande suivante à l'aide de la AWS CLI (remplacez le nom de la fonction par votre fonction et utilisez la région AWS dans laquelle se trouve votre fonction) :

```
aws lambda get-policy --function-name MyFunction --region us-east-1
```

Votre résultat doit être similaire à ce qui suit :

```
{
    "Policy": "{\"Version\":\"2012-10-17\",
    \"Statement\":[
        {\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:events:us-
east-1:123456789012:rule/MyRule\"}},
        \"Action\":\"lambda:InvokeFunction\",
        \"Resource\":\"arn:aws:lambda:us-east-1:123456789012:function:MyFunction\",
        \"Effect\":\"Allow\",
        \"Principal\":{\"Service\":\"events.amazonaws.com\"},
        \"Sid\":\"MyId\"}
    ],
    \"Id\":\"default\"}"
}
```

Si les informations suivantes s'affichent :

```
A client error (ResourceNotFoundException) occurred when calling the GetPolicy operation:
 The resource you requested does not exist.
```

Ou, si vous voyez le résultat, mais ne pouvez pas localiser events.amazonaws.com en tant qu'entité de confiance dans la stratégie, exécutez la commande suivante :

```
aws lambda add-permission \
--function-name MyFunction \
--statement-id MyId \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
```

Note

Si la politique est incorrecte, vous pouvez également modifier la règle dans la console CloudWatch Events en la supprimant, puis en l'ajoutant à nouveau à la règle. La console CloudWatch Events définira les autorisations appropriées sur la cible.
Si vous utilisez une version ou un alias Lambda spécifique, vous devez ajouter le paramètre `--qualifier` dans les commandes `aws lambda get-policy` et `aws lambda add-permission`.

```
aws lambda add-permission \
--function-name MyFunction \
--statement-id MyId \
--action 'lambda:InvokeFunction' \
--principal events.amazonaws.com \
--source-arn arn:aws:events:us-east-1:123456789012:rule/MyRule
```

```
--qualifier alias or version
```

Une autre raison pour laquelle la fonction Lambda ne parvient pas à se déclencher provient du fait que la politique que vous voyez lors de l'exécution de `get-policy` contient un champ `SourceAccount`. Un paramètre `SourceAccount` empêche CloudWatch Events d'appeler la fonction.

Je viens de créer/modifier une règle, mais elle ne correspond pas à un événement test

Lorsque vous modifiez une règle ou ses cibles, les événements entrants ne peuvent pas immédiatement commencer ou arrêter de chercher des correspondances aux nouvelles règles ou aux règles mises à jour. Les modifications ne prennent pas effet instantanément. Si les événements ne correspondent toujours pas après un court délai, vous pouvez également consulter plusieurs métriques CloudWatch concernant votre règle, par exemple, `TriggeredRules`, `Invocations` et `FailedInvocations` afin de procéder à un débogage plus approfondi. Pour de plus amples informations sur ces mesures, veuillez consulter Métriques et dimensions Amazon CloudWatch Events dans le Guide de l'utilisateur Amazon CloudWatch.

Si la règle est déclenchée par un événement à partir d'un service AWS, vous pouvez également utiliser l'action `TestEventPattern` pour tester le modèle d'événement de votre règle avec un événement test afin de vous assurer que le modèle d'événement de votre règle est correctement configuré. Pour de plus amples informations, veuillez consulter TestEventPatterndans la documentation de référence d'API Amazon CloudWatch Events.

Ma règle ne s'est pas déclenchée automatiquement à l'heure spécifiée dans le paramètre ScheduleExpression

Les paramètres ScheduleExpression sont exprimés au format UTC. Assurez-vous que vous avez défini le planning pour que la règle se déclenche automatiquement dans le fuseau horaire UTC. Si le paramètre ScheduleExpression est correct, suivez les étapes indiquées dans Je viens de créer/modifier une règle, mais elle ne correspond pas à un événement test (p. 119).

Ma règle ne s'est pas déclenchée au moment prévu

CloudWatch Events ne prend pas en charge la configuration d'une heure de début exacte lorsque vous créez une règle à exécuter à chaque période. Le compte à rebours pour l'exécution commence dès que la règle est créée.

Vous pouvez utiliser une expression cron pour appeler des cibles à un moment précis. Par exemple, vous pouvez utiliser une expression cron pour créer une règle qui se déclenche toutes les 4 heures, à l'heure exacte. Dans la console CloudWatch, vous utilisez l'expression cron `0 0/4 * * ? *`. Avec la AWS CLI, vous utilisez l'expression cron `cron(0 0/4 * * ? *)`. Par exemple, pour créer une règle nommée TestRule qui se déclenche toutes les 4 heures à l'aide de la AWS CLI, vous saisissez les informations suivantes à l'invite de commande :

```
aws events put-rule --name TestRule --schedule-expression 'cron(0 0/4 * * ? *)'
```

Vous pouvez utiliser l'expression cron `0/5 * * * ? *` pour déclencher une règle toutes les 5 minutes. Exemples :

```
aws events put-rule --name TestRule --schedule-expression 'cron(0/5 * * * ? *)'
```

CloudWatch Events ne fournit pas de précision de deuxième niveau dans les expressions de planification. Le niveau de résolution maximum lors de l'utilisation d'une expression cron est d'une minute. En raison de la nature distribuée de CloudWatch Events et des services cible, le délai entre le moment où la règle planifiée est déclenchée et celui où le service cible lance l'exécution de la ressource cible peut être de plusieurs secondes. Votre règle planifiée sera déclenchée au cours de cette minute, mais pas précisément à la seconde exacte.

Ma règle correspond aux appels d'API IAM, mais elle n'a pas été déclenchée

Le service IAM est uniquement disponible dans la région USA Est (Virginie du Nord). Ainsi, les événements d'appels d'API AWS provenant d'IAM sont uniquement disponibles dans cette région. Pour de plus amples informations, veuillez consulter Exemples d'événements CloudWatch Events provenant de services pris en charge (p. 43).

Ma règle ne fonctionne pas, car le rôle IAM qui lui est associé est ignoré lorsque la règle est déclenchée

Les rôles IAM des règles sont uniquement utilisés pour lier des événements aux flux Kinesis. Pour les fonctions Lambda et les rubriques Amazon SNS, vous devez fournir des autorisations basées sur une ressource.

Assurez-vous que vos points de terminaison AWS STS régionaux sont activés. CloudWatch Events s'adresse aux points de terminaison AWS STS régionaux lorsqu'il endosse le rôle IAM que vous avez fourni. Pour de plus amples informations, veuillez consulter Activation et désactivation de AWS STS dans une région AWS dans le Guide de l'utilisateur IAM.

J'ai créé une règle avec un paramètre EventPattern qui doit correspondre à une ressource, mais je ne vois aucun événement correspondant à la règle

La plupart des services AWS interprètent de la même manière les signes deux points (:) et barre oblique (/) dans les Amazon Resource Names (ARN). Cependant, CloudWatch Events recherche une correspondance exacte dans les modèles et règles d'événements. Veillez à utiliser les caractères ARN corrects lors de la création de modèles d'événements, afin qu'ils correspondent à la syntaxe ARN dans l'événement à faire correspondre.

Par ailleurs, le champ des ressources n'est pas rempli pour tous les événements (par exemple, les événements d'appels d'API AWS provenant de CloudTrail).

La diffusion de mon événement à la cible a été retardée

CloudWatch Events essaie de diffuser un événement à une cible pendant un délai maximum de 24 heures, sauf dans les cas où votre ressource cible est limitée. La première tentative a lieu dès que l'événement arrive dans le flux d'événements. Toutefois, si le service cible rencontre des problèmes ou si votre compte est limité, CloudWatch Events reprogramme automatiquement une diffusion ultérieure. Passé un délai de 24 heures après l'arrivée de l'événement, aucune tentative n'est programmée et la métrique `FailedInvocations` est publiée dans CloudWatch. Nous vous recommandons de créer une alarme CloudWatch sur la métrique `FailedInvocations`.

Certains événements ne sont pas livrés à ma cible

Si la cible d'une règle CloudWatch Events est contrainte pendant un certain temps, CloudWatch Events peut ne pas retenter la livraison. Par exemple, si la cible n'est pas prévue pour gérer le trafic d'événement entrant et que le service de la cible limite les demandes effectuées par CloudWatch Events en votre nom, CloudWatch Events peut ne pas retenter la livraison.

Ma règle a été déclenchée plusieurs fois en réponse à un événement. Quelle est la garantie offerte par CloudWatch Events pour le déclenchement de règles ou la diffusion d'événements aux cibles ?

Dans de rares cas, la même règle peut être déclenchée plusieurs fois pour un seul événement ou une seule période planifiée, ou la même cible peut être appelée plusieurs fois pour une règle déclenchée donnée.

Prévention des boucles infinies

Dans CloudWatch Events, il est possible de créer des règles qui conduisent à des boucles infinies, dans lesquelles une règle est exécutée de manière répétée. Par exemple, une règle peut détecter que les listes de contrôle d'accès (ACL) ont été modifiées sur un compartiment S3 et lancer un logiciel pour les modifier afin qu'elles aient l'état souhaité. Si la règle n'est pas correctement écrite, la modification suivante des listes de contrôle d'accès (ACL) déclenche à nouveau la règle, créant ainsi une boucle infinie.

Pour éviter ce problème, écrivez les règles de sorte que les actions déclenchées ne relancent pas la même règle. Par exemple, votre règle pourrait ne s'exécuter que si les listes ACL s'avèrent être dans un état incorrect plutôt qu'après une modification.

Une boucle infinie peut rapidement entraîner des coûts plus importants que prévu. Nous vous recommandons d'utiliser les budgets, qui vous avertissent lorsque les frais dépassent votre quota spécifié. Pour plus d'informations, consultez Gestion des coûts avec les budgets.

Mes événements ne sont pas livrés à la file d'attente Amazon SQS cible

La file d'attente Amazon SQS peut être chiffrée. Si vous créez une règle avec une file d'attente Amazon SQS chiffrée en tant que cible, la section suivante doit être incluse dans votre politique de clé KMS pour que l'événement soit livré avec succès à la file d'attente chiffrée.

```
{
                "Sid": "Allow CWE to use the key",
                "Effect": "Allow",
                "Principal": {
                                "Service": "events.amazonaws.com"
                },
                "Action": [
                                "kms:Decrypt",
                                "kms:GenerateDataKey"
                ],
                "Resource": "*"
}
```

Ma règle est déclenchée, mais je ne vois aucun message publié dans ma rubrique Amazon SNS

Assurez-vous de disposer de l'ensemble d'autorisations correct pour votre rubrique Amazon SNS. Exécutez la commande suivante à l'aide de la AWS CLI (remplacez l'ARN de la rubrique par votre rubrique et utilisez la région AWS dans laquelle est située votre rubrique) :

```
aws sns get-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic"
```

Vos attributs de stratégie doivent être similaires à ce qui suit :

```
"{\"Version\":\"2012-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:Receive\",
\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}},{\"Sid\":
\"Allow_Publish_Events\",
\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}]}"
```

Si une stratégie similaire à ce qui suit s'affiche, cela signifie que seule la stratégie par défaut est définie :

```
"{\"Version\":\"2008-10-17\",
\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",
\"Effect\":\"Allow\",
\"Principal\":{\"AWS\":\"*\"},
\"Action\":[\"SNS:Subscribe\",
\"SNS:ListSubscriptionsByTopic\",
\"SNS:DeleteTopic\",
\"SNS:GetTopicAttributes\",
\"SNS:Publish\",
\"SNS:RemovePermission\",
\"SNS:AddPermission\",
\"SNS:Receive\",
\"SNS:SetTopicAttributes\"],
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\",
\"Condition\":{\"StringEquals\":{\"AWS:SourceOwner\":\"123456789012\"}}}]}"
```

Si vous ne voyez pas `events.amazonaws.com` avec l'autorisation de publication dans votre stratégie, utilisez l'AWS CLI pour définir l'attribut de stratégie de la rubrique.

Copiez la stratégie actuelle et ajoutez la déclaration suivante à la liste des déclarations :

```
{\"Sid\":\"Allow_Publish_Events\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},
\"Action\":\"sns:Publish\",
\"Resource\":\"arn:aws:sns:us-east-1:123456789012:MyTopic\"}
```

La nouvelle stratégie doit ressembler à celle décrite précédemment.

Définissez les attributs de la rubrique avec l' AWS CLI:

```
aws sns set-topic-attributes --region us-east-1 --topic-arn "arn:aws:sns:us-
east-1:123456789012:MyTopic" --attribute-name Policy --attribute-value NEW_POLICY_STRING
```

> **Note**
>
> Si la politique est incorrecte, vous pouvez également modifier la règle dans la console CloudWatch Events en la supprimant, puis en l'ajoutant à nouveau à la règle. CloudWatch Events définit les autorisations correctes sur la cible.

Ma rubrique Amazon SNS dispose toujours d'autorisations pour CloudWatch Events, même après la suppression de la règle associée à la rubrique Amazon SNS

Lorsque vous créez une règle avec Amazon SNS comme cible, CloudWatch Events ajoute en votre nom l'autorisation d'accéder à votre rubrique Amazon SNS. Si vous supprimez la règle peu de temps après l'avoir créée, il est possible que CloudWatch Events ne puisse pas supprimer l'autorisation de votre rubrique Amazon SNS. Si cela se produit, vous pouvez supprimer l'autorisation de la rubrique à l'aide de la commande Amazon SNS set-topic-attributes.

Quelles clés de condition IAM puis-je utiliser avec CloudWatch Events ?

CloudWatch Events prend en charge les clés de condition à l'échelle d'AWS (consultez la page Clés disponibles dans le Guide de l'utilisateur IAM), ainsi que les clés de condition spécifiques aux services suivantes.

Comment puis-je savoir quand les règles CloudWatch Events sont interrompues ?

Vous pouvez utiliser l'alarme suivante pour être averti lorsque vos règles CloudWatch Events sont interrompues.

Pour créer une alarme pour vous alerter lorsque les règles sont interrompues

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Sélectionnez Créer une alarme. Dans le volet Métriques CloudWatch par catégorie, choisissez Events Metrics.
3. Dans la liste des métriques, sélectionnez FailedInvocations.
4. Au-dessus du graphique, choisissez Statistique, Somme.
5. Pour Période, choisissez une valeur, par exemple, 5 minutes. Choisissez Suivant.
6. Sous Seuil de l'alarme, pour Nom, saisissez un nom unique pour l'alarme, par exemple, myFailedRules. Pour Description, entrez une description de l'alarme, par exemple, Les règles ne diffusent pas les événements aux cibles.
7. Pour is, choisissez >= et 1. Pour pour, entrez 10.
8. Sous Actions, pour Whenever this alarm (Chaque fois que cette alarme), choisissez State is ALARM (L'état est ALARME).
9. Pour Send notification to (Envoyer une notification à), sélectionnez une rubrique Amazon SNS existante ou créez-en une. Pour créer une rubrique, choisissez New list. Saisissez un nom pour la nouvelle rubrique Amazon SNS, par exemple, myFailedRules.
10. Pour Email list, tapez une liste séparée par des virgules des adresses e-mail pour être informé lorsque l'alarme passe à l'état ALARME.
11. Sélectionnez Créer une alarme.

Historique du document

> **Note**
>
> Amazon EventBridge est la meilleure façon de gérer vos événements. CloudWatch Events et EventBridge représentent les mêmes services et API sous-jacents, mais EventBridge offre davantage de fonctionnalités. Les modifications que vous apportez dans CloudWatch ou EventBridge apparaîtront dans chaque console. Pour plus d'informations, consultez la section Amazon EventBridge.

Le tableau ci-après décrit les modifications importantes apportées dans chaque version du Guide de l'utilisateur CloudWatch Events, à partir de juin 2018. Pour recevoir les notifications des mises à jour de cette documentation, abonnez-vous à un flux RSS.

update-history-change	update-history-description	update-history-date
Prise en charge du balisage (p. 125)	Vous pouvez maintenant baliser certaines ressources CloudWatch Events. Pour plus d'informations, veuillez consulter Balisage de vos ressources Amazon CloudWatch Events dans le Guide de l'utilisateur Amazon CloudWatch Events.	21 mars 2019
Prise en charge des points de terminaison Amazon VPC (p. 125)	Vous pouvez maintenant établir une connexion privée entre votre VPC et CloudWatch Events. Pour plus d'informations, veuillez consulter Utilisation de CloudWatch Events avec les points de terminaison de VPC d'interface dans le Guide de l'utilisateur Amazon CloudWatch Events.	28 juin 2018

Le tableau suivant décrit les modifications majeures apportées au Guide de l'utilisateur Amazon CloudWatch Events.

Modification	Description	Date de parution
CodeBuild en tant que cible	Ajout de CodeBuild comme cible des règles d'événements. Pour de plus amples informations, veuillez consulter Didacticiel : Planifier des versions automatisées avec CodeBuild (p. 31).	13 décembre 2017
AWS Batch comme cible	Ajout d'AWS Batch comme cible des règles d'événements. Pour plus d'informations, consultez Événements AWS Batch.	8 septembre 2017

Modification	Description	Date de parution
Événements CodePipeline et AWS Glue	Ajout de la prise en charge des événements CodePipeline et AWS Glue. Pour de plus amples informations, veuillez consulter Evénements CodePipeline (p. 46) et AWS GlueÉvénements (p. 58).	8 septembre 2017
Événements CodeBuild et CodeCommit	Ajout de la prise en charge des événements CodeBuild et CodeCommit. Pour de plus amples informations, veuillez consulter Événements CodeBuild (p. 45).	3 août 2017
Autres cibles prises en charge	CodePipeline et Amazon Inspector peuvent être des cibles d'événements.	29 juin 2017
Prise en charge de l'envoi et de la réception d'événements entre des comptes AWS	Un compte AWS peut envoyer des événements à un autre compte AWS. Pour de plus amples informations, veuillez consulter Envoi et réception d'événements entre comptes AWS (p. 84).	29 juin 2017
Autres cibles prises en charge	Vous pouvez maintenant définir deux services AWS supplémentaires comme cibles pour les actions d'événements : les instances Amazon EC2 (via Run Command) et les machines d'état Step Functions. Pour de plus amples informations, veuillez consulter Mise en route avec Amazon CloudWatch Events (p. 6).	7 mars 2017
Événements Amazon EMR	Ajout de la prise en charge des événements pour Amazon EMR. Pour de plus amples informations, veuillez consulter Événements Amazon EMR (p. 49).	7 mars 2017
Événements AWS Health	Ajout de la prise en charge d'événements pour AWS Health. Pour de plus amples informations, veuillez consulter AWS HealthÉvénements (p. 63).	1 décembre 2016
Événements Amazon Elastic Container Service	Ajout de la prise en charge des événements pour Amazon ECS. Pour de plus amples informations, veuillez consulter Événements Amazon Elastic Container Service (p. 49).	21 novembre 2016
AWS Trusted AdvisorÉvénements	Ajout de la prise en charge des événements pour Trusted Advisor. Pour de plus amples informations, veuillez consulter AWS Trusted AdvisorÉvénements (p. 80).	18 novembre 2016
Événements Amazon Elastic Block Store	Ajout de la prise en charge des événements pour Amazon EBS. Pour de plus amples informations, veuillez consulter Événements Amazon EBS (p. 48).	14 novembre 2016
AWS CodeDeployÉvénements	Ajout de la prise en charge des événements pour CodeDeploy. Pour de plus amples informations, veuillez consulter AWS CodeDeployÉvénements (p. 45).	9 septembre 2016
Événements planifiés avec une granularité d'une minute	Ajout de la prise en charge des événements planifiés avec une granularité d'une minute. Pour de plus amples informations, veuillez consulter Expressions cron (p. 34) et Expressions de fréquence (p. 37).	19 avril 2016

Modification	Description	Date de parution
Files d'attente Amazon Simple Queue Service en tant que cibles	Ajout de la prise en charge des files d'attente Amazon SQS en tant que cibles. Pour de plus amples informations, veuillez consulter Qu'est-ce qu'Amazon CloudWatch Events ? (p. 1).	30 mars 2016
Événements Auto Scaling	Ajout de la prise en charge des événements pour hooks de cycle de vie Auto Scaling. Pour de plus amples informations, veuillez consulter Événements Amazon EC2 Auto Scaling (p. 48).	24 février 2016
Nouveau service	Première version de CloudWatch Events.	14 janvier 2016

Glossaire AWS

Pour connaître la terminologie la plus récente d'AWS, consultez le glossaire AWS dans la Référence générale d'AWS.