aws

Guide de l'utilisateur

Amazon EventBridge

Amazon EventBridge: Guide de l'utilisateur

Table of Contents

Qu'est-ce qu'Amazon EventBridge ?

EventBridge est un service sans serveur qui se sert des événements pour connecter les composants de l'application entre eux, ce qui vous permet de créer plus facilement des applications évolutives pilotées par les événements. Une architecture pilotée par les événements est un style de création de systèmes logiciels faiblement couplés qui fonctionnent ensemble en émettant des événements et en y répondant. L'architecture pilotée par les événements peut vous aider à gagner en agilité et à créer des applications fiables et évolutives.

Utilisez EventBridge pour router les événements en provenance de certaines sources, telles que les applications développées en interne, les services AWS et les logiciels tiers, vers des applications grand public à l'échelle de votre organisation. EventBridge met à votre disposition des moyens simples et cohérents pour ingérer, filtrer, transformer et transmettre les événements et favoriser ainsi la création rapide d'applications.

La vidéo suivante propose une brève présentation des fonctionnalités d'Amazon EventBridge :

EventBridge offre deux façons de traiter les événements : les bus d'événements et les canaux.

- Les bus d'événements sont des routeurs qui reçoivent les événements et les transmettent le cas échéant à des cibles. Les bus d'événements sont idéaux pour router les événements provenant de nombreuses sources vers de nombreuses cibles, avec une transformation facultative des événements avant leur livraison à une cible.

  La vidéo suivante offre une présentation générale des bus d'événements :

- Canaux EventBridge Pipes est prévu pour les intégrations point à point ; chaque canal reçoit les événements d'une source unique pour les traiter et les livrer à une cible unique. Les canaux incluent également la prise en charge des transformations avancées et l'enrichissement des événements avant leur livraison à une cible.

Les canaux et les bus d'événements sont souvent utilisés conjointement. Un cas d'utilisation courant est la création d'un canal avec un bus d'événements en tant que cible ; le canal envoie les événements au bus d'événements, qui les transmet ensuite à plusieurs cibles. Par exemple, vous pouvez créer un canal dont la source est un flux DynamoDB, ainsi qu'un bus d'événements faisant office de cible. Le canal reçoit les événements du flux DynamoDB et les envoie au bus d'événements,

qui les envoie ensuite à plusieurs cibles en fonction des règles que vous avez spécifiées pour le bus d'événements.

EventBridge est une évolution d'Amazon CloudWatch Events

Auparavant, EventBridge s'appelait Amazon CloudWatch Events. Le bus d'événements par défaut et les règles que vous avez créées dans CloudWatch Events s'affichent également dans la console EventBridge. Comme EventBridge utilise la même API CloudWatch Events, votre code qui utilise l'API CloudWatch Events existante est inchangé.

EventBridge s'appuie sur les capacités de CloudWatch Events avec des fonctionnalités telles que les événements partenaires, le registre de schémas et EventBridge Pipes. Les nouvelles fonctionnalités ajoutées à EventBridge ne sont pas ajoutées à CloudWatch Events. Pour de plus amples informations, veuillez consulter ???.

Toutes les fonctionnalités de CloudWatch Events auxquelles vous êtes habitué sont également présentes dans EventBridge, notamment :

- ???
- ???
- ???
- ???

Les fonctionnalités EventBridge qui reposent sur les événements et qui en accroissent les capacités sont les suivantes :

- ???
- ???
- ???
- ???

EventBridge Configuration et prérequis d'Amazon

Pour utiliser Amazon EventBridge, vous avez besoin d'un AWS compte. Votre compte vous permet d'utiliser des services tels qu'Amazon EC2 pour générer des événements visibles dans la EventBridge console. Vous pouvez également installer et configurer le AWS Command Line Interface (AWS CLI) pour utiliser une interface de ligne de commande pour voir les événements.

Rubriques

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

1. Ouvrez https://portal.aws.amazon.com/billing/signup.
2. Suivez les instructions en ligne.

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

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

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

Création d'un utilisateur doté d'un accès administratif

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

Sécurisez votre Utilisateur racine d'un compte AWS

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

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

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

Création d'un utilisateur doté d'un accès administratif

1. Activez IAM Identity Center.

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

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

Connectez-vous en tant qu'utilisateur disposant d'un accès administratif

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

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

Attribuer l'accès à des utilisateurs supplémentaires

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

   Pour obtenir des instructions, voir Création d'un ensemble d'autorisations dans le guide de AWS IAM Identity Center l'utilisateur.
2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

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

Connectez-vous à la EventBridge console Amazon

Pour vous connecter à la EventBridge console Amazon

- Connectez-vous à la EventBridge console Amazon AWS Management Console et ouvrez-la à l'adresse https://console.aws.amazon.com/events/.

Informations d’identification de compte

Bien que vous puissiez utiliser vos informations d'identification d'utilisateur root pour y accéder EventBridge, nous vous recommandons d'utiliser un compte AWS Identity and Access Management (IAM) à la place. Si vous utilisez un compte IAM pour y accéder EventBridge, vous devez disposer des autorisations suivantes.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "events:*"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:events:*:*:*"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": "events.amazonaws.com"
        }
      }
    }
  ]
}
```

Pour plus d'informations, consultez Authentification.

Configurez le AWS Command Line Interface

Vous pouvez utiliser le AWS CLI pour effectuer des EventBridge opérations.

Pour plus d'informations sur l'installation et la configuration du AWS CLI, voir Getting Set Up with the AWS Command Line Interface dans le guide de AWS Command Line Interface l'utilisateur.

Points de terminaison régionaux

Vous devez activer les points de terminaison régionaux par défaut pour pouvoir les utiliser EventBridge. Pour plus d'informations, consultez la section Activation et désactivation AWS STS dans une AWS région dans le guide de l'utilisateur IAM.

Commencer à utiliser Amazon EventBridge

La base EventBridge est de créer des règles qui acheminent les événements vers une cible. Dans cette section, vous allez créer une règle de base. Pour obtenir des didacticiels sur des scénarios spécifiques et des cibles spécifiques, consultez Didacticiels Amazon EventBridge.

Création d'une règle dans Amazon EventBridge

Pour créer une règle pour les événements, vous devez spécifier une action à effectuer lors de la EventBridge réception d'un événement qui correspond au modèle d'événement défini dans la règle. Lorsqu'un événement correspond, EventBridge envoie l'événement à la cible spécifiée et déclenche l'action définie dans la règle.

Lorsqu'un AWS service de votre AWS compte émet un événement, il est toujours redirigé vers le bus d'événements par défaut de votre compte. Pour écrire une règle qui correspond aux événements des AWS services de votre compte, vous devez l'associer au bus d'événements par défaut.

Pour créer une règle pour un AWS service

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle.

   Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle corresponde aux événements provenant de votre compte, sélectionnez Bus d’événements par défaut AWS . Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.
6. Pour Type de règle, choisissez Règle avec un modèle d’événement.
7. Choisissez Suivant.
8. Pour Source d'événement, choisissez Services AWS .
9. (Facultatif) Pour Exemples d’événements, choisissez le type d’événement.

10. Pour Modèle d’événement, effectuez l’une des actions suivantes :
    - Pour utiliser un modèle pour créer votre modèle d’événement, choisissez Formulaire de modèle d’événement et choisissez la Source d’événement et le Type d’événement. Si vous choisissez Tous les événements comme type d'événement, tous les événements émis par ce AWS service seront conformes à la règle.

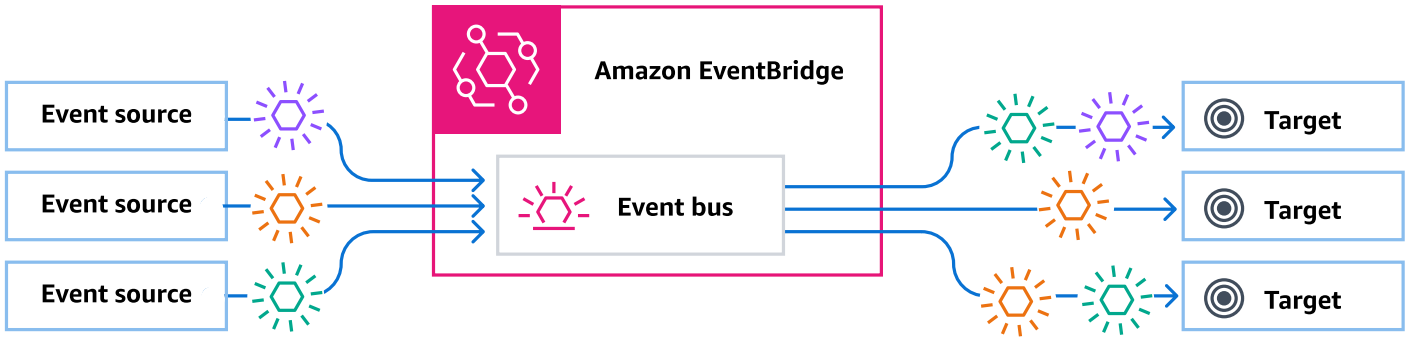
      Pour personnaliser le modèle, choisissez Custom pattern (JSON editor) (Modèle personnalisé [éditeur JSON]) et effectuez vos modifications.
    - Pour utiliser un modèle d'événement personnalisé, choisissez Custom pattern (JSON editor) (Modèle personnalisé [éditeur JSON]) et créez votre modèle d'événement.
11. Choisissez Suivant.
12. Pour Types de cibles, choisissez service AWS .
13. Pour Sélectionner une cible, choisissez le AWS service auquel vous souhaitez envoyer des informations en cas de EventBridge détection d'un événement correspondant au modèle d'événement.
14. Les champs affichés varient en fonction du service que vous choisissez. Entrez les informations spécifiques requises pour ce type de cible.
15. Pour de nombreux types de cibles, EventBridge nécessite des autorisations pour envoyer des événements à la cible. Dans ces cas, EventBridge vous pouvez créer le rôle IAM nécessaire à l'exécution de votre règle. Effectuez l’une des actions suivantes :
    - Pour créer un rôle IAM automatiquement, sélectionnez Create a new role for this specific resource.
    - Pour utiliser un rôle IAM que vous avez créé précédemment, choisissez Utiliser le rôle existant et sélectionnez le rôle existant dans la liste déroulante.
16. (Facultatif) Pour Additional settings (Paramètres supplémentaires), procédez comme suit :
    a. Pour Maximum age of event (Âge maximal de l'événement), saisissez une valeur comprise entre une minute (00:01) et 24 heures (24:00).
    b. Pour Retry attempts (Nouvelles tentatives), saisissez un nombre compris entre 0 et 185.
    c. Pour la file d'attente de lettres mortes, choisissez si vous souhaitez utiliser une file d'attente Amazon SQS standard comme file d'attente de lettres mortes. EventBridge envoie les événements correspondant à cette règle à la file d'attente des lettres mortes s'ils ne sont pas correctement transmis à la cible. Effectuez l’une des actions suivantes :
       - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.

- Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.
- Choisissez Sélectionnez une file d'attente Amazon SQS dans un autre AWS compte en tant que file d'attente de lettres mortes, puis entrez l'ARN de la file d'attente à utiliser. Vous devez associer à la file d'attente une politique basée sur les ressources qui EventBridge autorise l'envoi de messages. Pour plus d'informations, consultez Octroi d'autorisations à la file d'attente de lettres mortes.

17. (Facultatif) Sélectionnez Add another target (Ajouter une autre cible) pour ajouter une nouvelle cible pour cette règle.
18. Choisissez Suivant.
19. (Facultatif) Saisissez une ou plusieurs balises pour la règle. Pour plus d'informations, consultez EventBridge Balises Amazon.
20. Choisissez Suivant.
21. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Amazon EventBridge Event Bus

Un bus d'événements est un routeur qui reçoit des événements et les livre à zéro ou plusieurs destinations, ou cibles. Les bus d'événements sont idéaux pour router les événements provenant de nombreuses sources vers de nombreuses cibles, avec une transformation facultative des événements avant leur livraison à une cible.



Les règles associées au bus d'événements évaluent les événements au fur et à mesure qu'ils arrivent. Chaque règle vérifie si un événement correspond au modèle de la règle. Si l'événement correspond, EventBridge envoie l'événement

Vous associez une règle à un bus d'événements spécifique, de sorte que la règle ne s'applique qu'aux événements reçus par ce bus d'événements.

> **Note**
>
> Vous pouvez également traiter des événements à l'aide de EventBridge Pipes. EventBridge Les tubes sont destinés aux point-to-point intégrations ; chaque canal reçoit des événements provenant d'une source unique pour les traiter et les transmettre à une cible unique.
> Les canaux incluent également la prise en charge des transformations avancées et l'enrichissement des événements avant leur livraison à une cible. Pour plus d'informations, consultez ???.

Rubriques

- Fonctionnement des bus d'événements
- Concepts EventBridge d'Amazon Event Bus
- Création d'un bus EventBridge d'événements Amazon

- Mettre à jour un bus EventBridge d'événements Amazon
- Supprimer un bus d' EventBridge événements Amazon
- Autorisations pour les bus d'événements Amazon EventBridge
- Génération d'un modèle AWS CloudFormation à partir d'un bus d'événements Amazon EventBridge

Fonctionnement des bus d’événements

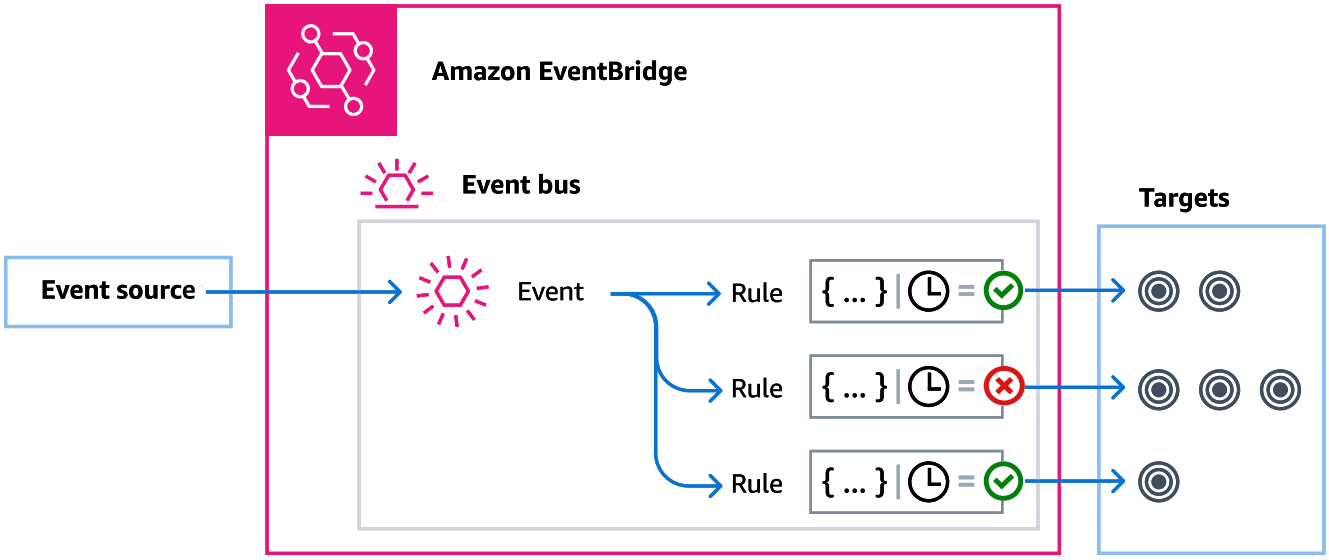
Les bus d’événements vous permettent de router les événements provenant de plusieurs sources vers plusieurs destinations, ou cibles.

Globalement, voici leur fonctionnement :

1. Une source d'événement, qui peut être un AWS service, votre propre application personnalisée ou un fournisseur SaaS, envoie un événement à un bus d'événements.
2. EventBridge évalue ensuite l'événement par rapport à chaque règle définie pour ce bus d'événements.

   Pour chaque événement correspondant à une règle, envoie EventBridge ensuite l'événement aux cibles spécifiées pour cette règle. Facultativement, dans le cadre de la règle, vous pouvez également spécifier comment EventBridge transformer l'événement avant de l'envoyer à la ou aux cibles.

   Un événement peut correspondre à plusieurs règles, et chaque règle peut spécifier jusqu’à cinq cibles. (Il se peut qu'un événement ne corresponde à aucune règle, auquel cas aucune action n' EventBridge est entreprise.)



Prenons un exemple utilisant le bus d'événements EventBridge par défaut, qui reçoit automatiquement les événements des AWS services :

1. Vous créez une règle sur le bus d'événements par défaut pour l'événement `EC2 Instance State-change Notification` :
   - Vous indiquez que la règle correspond aux événements où le `state` d'une instance Amazon EC2 est passé à `running`.

     Pour ce faire, spécifiez le code JSON qui définit les attributs et les valeurs auxquels un événement doit correspondre pour déclencher la règle. C'est ce qu'on appelle un modèle d'événement.

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

   - Vous indiquez que la cible de la règle doit être une fonction Lambda donnée.
2. Chaque fois qu'une instance Amazon EC2 change d'état, Amazon EC2 (la source de l'événement) envoie automatiquement cet événement au bus d'événements par défaut.

3. EventBridge évalue tous les événements envoyés au bus d'événements par défaut par rapport à la règle que vous avez créée.

   Si l'événement correspond à votre règle (c'est-à-dire s'il s'agit d'une instance Amazon EC2 dont l'état est changé`running`), EventBridge envoie l'événement à la cible spécifiée. Dans ce cas, il s'agit de la fonction Lambda.

La vidéo suivante décrit ce que sont les bus d'événements et ce qu'ils font : Que sont les bus d'événements ?

La vidéo suivante décrit les différents bus d'événements et explique dans quels cas les utiliser : Différences entre les bus d'événements

Concepts EventBridge d'Amazon Event Bus

Voici un examen approfondi des principaux composants d'une architecture orientée événement construite sur des bus d'événements.

Bus d'événements

Un bus d'événements est un routeur qui reçoit des événements et les livre à zéro ou plusieurs destinations, ou cibles. Utilisez un bus d'événements lorsque vous devez router les événements provenant de nombreuses sources vers de nombreuses cibles, avec une transformation facultative des événements avant leur livraison à une cible.

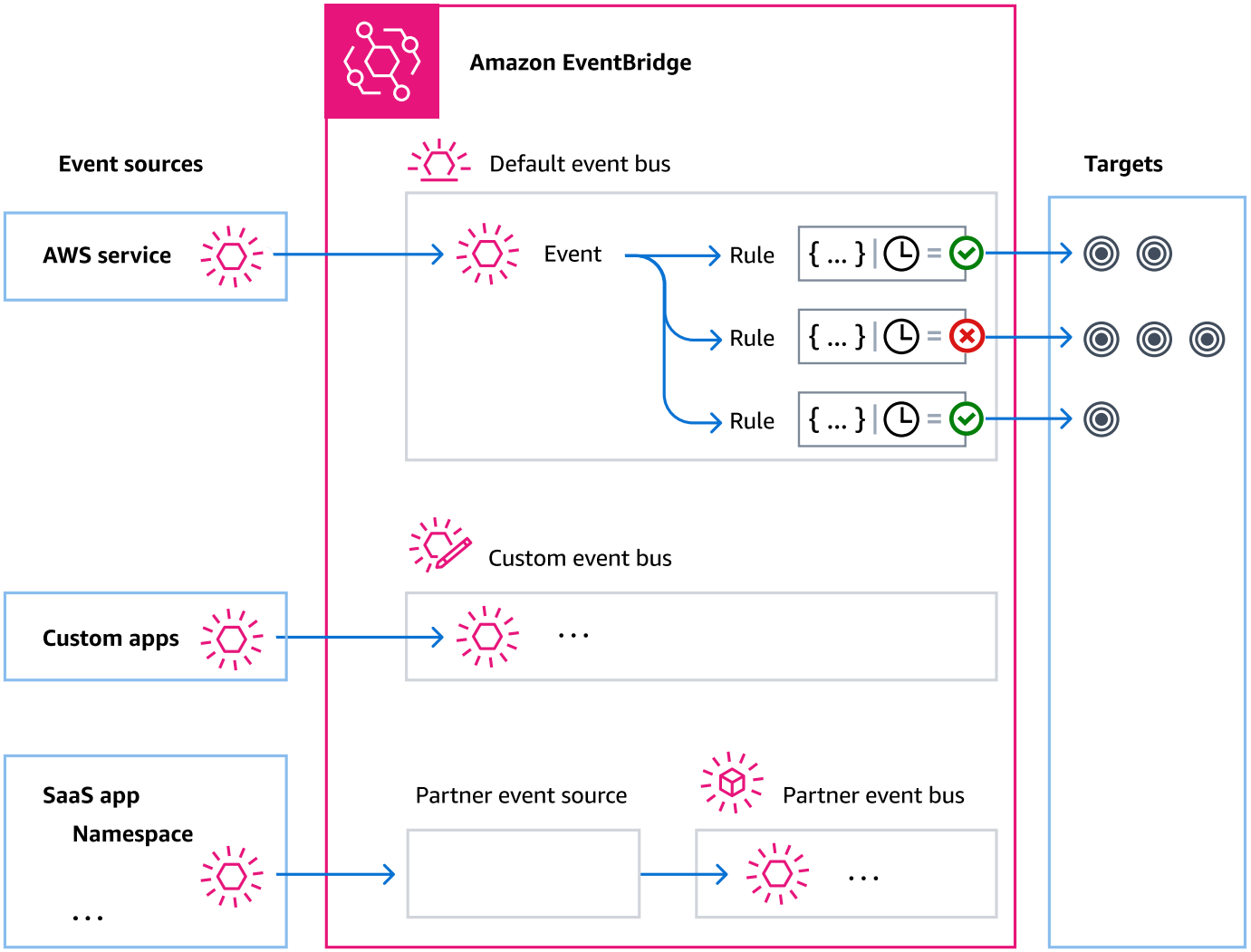
Votre compte inclut un bus d'événements par défaut qui reçoit automatiquement les événements des AWS services. Vous pouvez également :

- Créez des bus d'événements supplémentaires, appelés bus d'événements personnalisés et spécifiez les événements qu'ils reçoivent.
- Créez des bus d'événements partenaires, qui reçoivent les événements des partenaires SaaS.

Les cas d'utilisation courants des bus d'événements incluent :

- Utilisation d'un bus d'événements comme agent entre différentes charges de travail, différents services ou systèmes.

- Utilisation de plusieurs bus d'événements dans vos applications pour répartir le trafic d'événements. Par exemple, la création d'un bus pour traiter les événements contenant des informations d'identification personnelles (PII), et d'un autre bus pour les événements qui n'en contiennent pas.
- Agrégation d'événements en envoyant des événements provenant de plusieurs bus d'événements vers un bus d'événements centralisé. Ce bus centralisé peut se trouver dans le même compte que les autres bus, mais il peut également se trouver dans un autre région.



Événements

Dans sa forme la plus simple, un EventBridge événement est un objet JSON envoyé à un bus ou à un canal d'événements.

Dans le contexte de l’architecture basée sur les événements (EDA), un événement représente souvent un indicateur de changement dans une ressource ou un environnement.

Pour plus d’informations, consultez ???.

Sources des évènements

EventBridge peut recevoir des événements provenant de sources d'événements, notamment :

- AWS services
- Applications personnalisées
- Partenaires de type logiciel en tant que service (SaaS)

Règles

Une règle reçoit des événements entrants et les envoie, le cas échéant, à des cibles pour être traités. Vous pouvez spécifier la manière dont chaque règle invoque ses cibles en fonction des éléments suivants :

- Un modèle d’événement, qui contient un ou plusieurs filtres de mise en correspondance des événements. Les modèles d’événements peuvent inclure des filtres qui mettent en correspondance les éléments suivants :
  - Métadonnées d’événement : données relatives à l’événement, telles que la source de l’événement, ou le compte ou la région d’origine de l’événement.
  - Données d’événement : propriétés de l’événement lui-même. Ces propriétés varient en fonction de l’événement.
  - Contenu de l’événement : valeurs de propriété réelles des données d’événement.
- Un calendrier pour invoquer la ou les cibles à intervalles réguliers.

  Vous pouvez spécifier une règle planifiée dans EventBridge ou à l'aide du EventBridge planificateur.

> **Note**
>
> EventBridge propose Amazon EventBridge Scheduler, un planificateur sans serveur qui vous permet de créer, d'exécuter et de gérer des tâches à partir d'un service géré centralisé. EventBridge Le planificateur est hautement personnalisable et offre une

évolutivité améliorée par rapport aux règles EventBridge planifiées, avec un ensemble plus large d'opérations et de services d'API cibles. AWS
Nous vous recommandons d'utiliser le EventBridge planificateur pour appeler des cibles selon un calendrier. Pour plus d'informations, consultez ???.

Chaque règle est définie pour un bus d'événements spécifique et ne s'applique qu'aux événements de ce bus d'événements.

Une seule règle peut envoyer un événement à cinq cibles au maximum.

Par défaut, vous pouvez configurer jusqu'à 300 règles par bus d'événements. Ce quota peut être porté à des milliers de règles dans la console Service Quotas. Étant donné que la limite de règles s'applique à chaque bus, si vous avez besoin d'encore plus de règles, vous pouvez créer des bus d'événements personnalisés supplémentaires dans votre compte.

Vous pouvez personnaliser la façon dont les événements sont reçus dans votre compte en créant des bus d'événements dotés de différentes autorisations pour différents services.

Pour personnaliser la structure ou la date d'un événement avant de EventBridge le transmettre à une cible, utilisez le transformateur d'entrée pour modifier les informations avant qu'elles ne soient transmises à la cible.

Pour plus d'informations, consultez ???.

Cibles

Une cible est une ressource ou un point de terminaison qui EventBridge envoie un événement lorsque celui-ci correspond au modèle d'événement défini pour une règle.

Une cible peut recevoir plusieurs événements provenant de plusieurs bus d'événements.

Pour plus d'informations, consultez ???.

Fonctionnalités avancées pour les bus d'événements

EventBridge inclut les fonctionnalités suivantes pour vous aider à développer, gérer et utiliser les bus d'événements.

Utilisation de destinations d'API pour activer les appels d'API REST entre les services

EventBridge Les destinations d'API sont des points de terminaison HTTP que vous pouvez définir comme cible d'une règle, de la même manière que vous enverriez des données d'événements à un AWS service ou à une ressource. En utilisant les destinations d'API, vous pouvez utiliser des appels d'API pour router les événements entre les services AWS , les applications SaaS intégrées et vos applications en dehors d' AWS. Lorsque vous créez une destination d'API, vous spécifiez une connexion destinée à son utilisation. Chaque connexion inclut des détails sur le type d'autorisation et les paramètres à utiliser pour être autorisée auprès du point de terminaison de destination d'API.

Archivage-relecture d'événements pour faciliter le développement et la reprise après sinistre

Vous pouvez archiver (ou enregistrer) des événements, puis les relire ultérieurement à partir de l'archive. L'archivage est utile pour :

- Tester une application, car vous disposez d'un magasin d'événements à utiliser plutôt que d'avoir à attendre de nouveaux événements.
- Hydrater un nouveau service dès sa première mise en ligne.
- Renforcer la durabilité de vos applications orientées événement.

Utilisation du registre des schémas pour démarrer rapidement la création de modèles d'événements

Lorsque vous créez des applications sans serveur qui utilisent EventBridge, il peut être utile de connaître la structure des événements typiques sans avoir à générer l'événement. La structure des événements est décrite dans des schémas, qui sont disponibles pour tous les événements générés par les AWS services sur EventBridge.

Pour les événements qui ne sont pas liés AWS aux services, vous pouvez :

- Créer ou charger des schémas personnalisés.
- Utilisez Schema Discovery pour créer EventBridge automatiquement des schémas pour les événements envoyés au bus d'événements.

Une fois que vous disposez d'un schéma pour un événement, vous pouvez télécharger des liaisons de code pour les langages de programmation usuels.

Gestion des ressources et de l'accès avec des politiques

Pour organiser les AWS ressources ou suivre les coûts EventBridge, vous pouvez attribuer une étiquette personnalisée aux AWS ressources. À l'aide de politiques basées sur des balises, vous pouvez contrôler ce que les ressources peuvent et ne peuvent pas faire dans ce cadre EventBridge.

Outre les politiques basées sur les balises, EventBridge prend en charge les politiques basées sur l'identité et les ressources pour contrôler l'accès à. EventBridge Utilisez des politiques basées sur l'identité pour contrôler les autorisations d'un groupe, d'un rôle ou d'un utilisateur. Utilisez des politiques basées sur les ressources pour accorder des autorisations spécifiques à chaque ressource, comme une fonction Lambda ou une rubrique Amazon SNS.

Création d'un bus EventBridge d'événements Amazon

Vous pouvez créer un bus d'événements personnalisé pour recevoir des événements de vos applications. Vos applications peuvent également envoyer des événements au bus d'événements par défaut. Lorsque vous créez un bus d'événements, vous pouvez attacher une politique basée sur les ressources pour accorder des autorisations à d'autres comptes. Les autres comptes peuvent alors envoyer des événements au bus d'événements dans le compte actuel.

La vidéo suivante décrit la création de bus d'événements : Création d'un bus d'événements

Pour créer un bus d'événement personnalisé

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Choisissez Create event bus (Créer un bus d'événement).
4. Saisissez un nom pour le nouveau bus d'événement.
5. Choisissez la forme KMS key EventBridge à utiliser lors du chiffrement des données d'événements stockées sur le bus d'événements.

> **Note**
>
> Les archives et la découverte de schémas ne sont pas prises en charge pour les bus d'événements chiffrés à l'aide d'un clé gérée par le client. Pour activer la découverte d'archives ou de schémas sur un bus d'événements, choisissez d'utiliser un Clé détenue par AWS. Pour plus d'informations, consultez ???.

- Choisissez Utiliser Clé détenue par AWS pour chiffrer EventBridge les données à l'aide d'un Clé détenue par AWS.

Il s' Clé détenue par AWS agit d'un compte KMS key qui EventBridge possède et gère pour une utilisation dans plusieurs AWS comptes. En général, à moins que vous ne soyez obligé d'auditer ou de contrôler la clé de chiffrement qui protège vos ressources, une Clé détenue par AWS est un bon choix.

Il s’agit de l’option par défaut.

- Choisissez Utiliser clé gérée par le client pour chiffrer les données EventBridge à l'aide de celles clé gérée par le client que vous spécifiez ou créez.

  Clés gérées par le client se trouvent KMS keys dans le AWS compte que vous créez, détenez et gérez. Vous en avez le plein contrôle KMS keys.

  a. Spécifiez un existant clé gérée par le client ou choisissez Créer un nouveau KMS key.

     EventBridge affiche le statut de la clé et tous les alias de clé associés à la clé spécifiée clé gérée par le client.

  b. Choisissez la file d'attente Amazon SQS à utiliser comme file d'attente de lettres mortes (DLQ) pour ce bus d'événements, le cas échéant.

     EventBridge envoie les événements qui ne sont pas correctement chiffrés au DLQ, s'il est configuré, afin que vous puissiez les traiter ultérieurement.

6. Configurez les fonctionnalités optionnelles du bus d'événements :

   - Spécifiez une politique basée sur les ressources en effectuant l'une des opérations suivantes :
     - Entrez la politique qui inclut les autorisations à accorder pour le bus d’événements. Vous pouvez coller une politique provenant d’une autre source ou entrer le code JSON de la politique. Vous pouvez utiliser l'un des exemples de politiques et le modifier en fonction de votre environnement.
     - Pour utiliser un modèle pour la politique, choisissez Charger un modèle. Modifiez la politique en fonction de votre environnement, notamment en ajoutant des actions supplémentaires que vous autorisez le principal de la politique à utiliser.

     Pour plus d'informations sur l'octroi d'autorisations à un bus d'événements par le biais de politiques basées sur les ressources, consultez. ???
   - Activer une archive (facultatif)

Vous pouvez créer une archive des événements afin de pouvoir facilement les rejouer ultérieurement. Par exemple, vous souhaitez peut-être relire des événements pour récupérer suite à des erreurs ou pour valider une nouvelle fonctionnalité de votre application. Pour plus d'informations, consultez ???.

a. Sous Archives, sélectionnez Activé.

b. Spécifiez le nom et la description de l'archive.

> **Note**
>
> Les archives et la découverte de schémas ne sont pas prises en charge pour les bus d'événements chiffrés à l'aide d'un clé gérée par le client. Pour activer la découverte d'archives ou de schémas sur un bus d'événements, choisissez d'utiliser un Clé détenue par AWS. Pour plus d'informations, consultez ???.

- Activer la découverte de schémas (facultatif)

  Activez la découverte de schémas pour déduire EventBridge automatiquement les schémas directement à partir des événements exécutés sur ce bus d'événements. Pour plus d'informations, consultez ???.

  a. Sous Découverte du schéma, sélectionnez Activé.

  > **Note**
  >
  > Les archives et la découverte de schémas ne sont pas prises en charge pour les bus d'événements chiffrés à l'aide d'un clé gérée par le client. Pour activer la découverte d'archives ou de schémas sur un bus d'événements, choisissez d'utiliser un Clé détenue par AWS. Pour plus d'informations, consultez ???.

- Spécifiez les balises (facultatif)

  Une balise est une étiquette d'attribut personnalisée que vous attribuez à une AWS ressource. Utilisez des balises pour identifier et organiser vos AWS ressources. De nombreux AWS services prennent en charge le balisage. Vous pouvez donc attribuer le même tag aux ressources de différents services pour indiquer que les ressources sont liées. Pour plus d'informations, consultez ???.

  a. Sous Balises, choisissez Ajouter une nouvelle balise.

b. Spécifiez une clé et, éventuellement, une valeur pour la nouvelle balise.

7. Sélectionnez Create (Créer).

Mettre à jour un bus EventBridge d'événements Amazon

Vous pouvez mettre à jour la configuration des bus d'événements après les avoir créés. Cela inclut le bus d'événements par défaut, qui est EventBridge créé automatiquement dans votre compte.

Mettre à jour le KMS key code utilisé pour le chiffrement

> **Note**
>
> Les archives et la découverte de schémas ne sont pas prises en charge pour les bus d'événements chiffrés à l'aide d'un clé gérée par le client. Pour activer la découverte d'archives ou de schémas sur un bus d'événements, choisissez d'utiliser un Clé détenue par AWS. Pour plus d’informations, consultez ???.

Pour modifier le paramètre KMS key utilisé pour le chiffrement au repos sur un bus d'événements à l'aide de la EventBridge console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Choisissez le bus d'événements que vous souhaitez mettre à jour.
4. Sur la page des détails du bus d'événements, choisissez l'onglet Chiffrement.
5. Choisissez la forme KMS key EventBridge à utiliser lors du chiffrement des données d'événements stockées sur le bus d'événements :
   - Choisissez Utiliser Clé détenue par AWS pour chiffrer EventBridge les données à l'aide d'un Clé détenue par AWS.

     Il s' Clé détenue par AWS agit d'un compte KMS key qui EventBridge possède et gère pour une utilisation dans plusieurs AWS comptes. En général, à moins que vous ne soyez obligé d'auditer ou de contrôler la clé de chiffrement qui protège vos ressources, une Clé détenue par AWS est un bon choix.

     Il s’agit de l’option par défaut.

- Choisissez Utiliser clé gérée par le client pour chiffrer les données EventBridge à l'aide de celles clé gérée par le client que vous spécifiez ou créez.

  Clés gérées par le client se trouvent KMS keys dans le AWS compte que vous créez, détenez et gérez. Vous en avez le plein contrôle KMS keys.

  a. Spécifiez un existant clé gérée par le client ou choisissez Créer un nouveau KMS key.

     EventBridge affiche le statut de la clé et tous les alias de clé associés à la clé spécifiée clé gérée par le client.

  b. Choisissez la file d'attente Amazon SQS à utiliser comme file d'attente de lettres mortes (DLQ) pour ce bus d'événements, le cas échéant.

     EventBridge envoie les événements qui ne sont pas correctement chiffrés au DLQ, s'il est configuré, afin que vous puissiez les traiter ultérieurement.

Mettre à jour les autorisations sur un bus d'événements

Vous pouvez accorder des autorisations supplémentaires à un bus d'événements en y attachant une politique basée sur les ressources. Pour obtenir des instructions détaillées sur la mise à jour des autorisations accordées à un bus d'événements, consultez la section Gestion des autorisations de bus d'événements.

Ajouter ou supprimer des archives sur les bus d'événements

Une archive vous permet de capturer des événements afin de pouvoir facilement les rejouer ultérieurement. Par exemple, vous souhaitez peut-être relire des événements pour récupérer suite à des erreurs ou pour valider une nouvelle fonctionnalité de votre application. Pour plus d'informations, consultez la section EventBridge Archiver et rejouer.

> **Note**
>
> Les archives et la découverte de schémas ne sont pas prises en charge pour les bus d'événements chiffrés à l'aide d'un clé gérée par le client. Pour activer la découverte d'archives ou de schémas sur un bus d'événements, choisissez d'utiliser un Clé détenue par AWS. Pour plus d'informations, consultez ???.

Pour ajouter ou supprimer une archive d'un bus d'événements à l'aide de la EventBridge console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Choisissez le bus d'événements que vous souhaitez mettre à jour.
4. Sur la page des détails du bus d'événements, choisissez l'onglet Archives.
5. Effectuez l'une des actions suivantes :
   - Pour ajouter une archive :
     a. Choisissez Créer une archive.
     b. Spécifiez les attributs de l'archive.
     c. Choisissez Suivant.
     d. Choisissez le modèle d'événements à appliquer aux événements de l'archive.
     e. Choisissez Créer une archive.
   - Pour supprimer une archive :
     a. Pour le tag que vous souhaitez supprimer, choisissez Supprimer.
     b. Entrez le nom de l'archive, puis choisissez Supprimer.

        L'archive est définitivement supprimée. Vous ne pouvez pas annuler cette opération.

Pour créer ou supprimer une archive pour un bus d'événements à l'aide du AWS CLI

- Pour créer une archive, utilisez create-archive.

  Pour supprimer définitivement une archive, utilisez delete-archive.

Démarrage ou arrêt de la découverte de schémas sur les bus d'événements

Pour plus d'informations sur la découverte de schémas, consultez la section EventBridge Schémas.

> **Note**
>
> Les archives et la découverte de schémas ne sont pas prises en charge pour les bus d'événements chiffrés à l'aide d'un clé gérée par le client. Pour activer la découverte

d'archives ou de schémas sur un bus d'événements, choisissez d'utiliser un Clé détenue par AWS. Pour plus d'informations, consultez ???.

Pour démarrer ou arrêter la découverte de schémas sur un bus d'événements à l'aide de la EventBridge console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Choisissez le bus d'événements que vous souhaitez mettre à jour.
4. Effectuez l'une des actions suivantes :
   - Pour démarrer la découverte du schéma, choisissez Démarrer la découverte.
   - Pour arrêter la découverte du schéma, choisissez Supprimer la découverte.

Pour démarrer ou arrêter la découverte de schémas sur un bus d'événements à l'aide du AWS CLI

- Pour démarrer la découverte du schéma, utilisez create-discoverer.

  Pour arrêter la découverte du schéma, utilisez delete-discoverer.

Ajouter ou supprimer des tags sur les bus d'événements

Une balise est une étiquette d'attribut personnalisée que vous attribuez ou AWS assignez à une AWS ressource. Utilisez des balises pour identifier et organiser vos AWS ressources. Pour plus d'informations, consultez la section EventBridge Tags.

Pour ajouter ou supprimer des balises d'un bus d'événements à l'aide de la EventBridge console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Choisissez le bus d'événements que vous souhaitez mettre à jour.
4. Sur la page des détails du bus d'événements, choisissez l'onglet Tags, puis sélectionnez Gérer les tags.
5. Effectuez l'une des actions suivantes :
   - Pour ajouter un tag, procédez comme suit :

a. Sélectionnez Ajouter une nouvelle balise.
b. Spécifiez la clé et la valeur de la balise
c. Choisissez Mettre à jour.

- Pour supprimer un tag, procédez comme suit :

a. Pour le tag que vous souhaitez supprimer, choisissez Supprimer.
b. Choisissez Mettre à jour.

Pour ajouter ou supprimer des balises dans un bus d'événements à l'aide du AWS CLI

- Pour ajouter des balises, utilisez tag-resource.

  Pour supprimer des balises, utilisez untag-resource.

Mise à jour du bus d'événements par défaut à l'aide de AWS CloudFormation

AWS CloudFormation vous permet de configurer et de gérer vos AWS ressources sur l'ensemble des comptes et des régions de manière centralisée et reproductible en traitant l'infrastructure comme du code. CloudFormation pour ce faire, vous pouvez créer des modèles qui définissent les ressources que vous souhaitez approvisionner et gérer.

Étant donné EventBridge que le bus d'événements par défaut est automatiquement intégré à votre compte, vous ne pouvez pas le créer à l'aide d'un CloudFormation modèle, comme vous le feriez normalement pour toute ressource que vous souhaitez inclure dans une CloudFormation pile. Pour inclure le bus d'événements par défaut dans une CloudFormation pile, vous devez d'abord l'importer dans une pile. Une fois que vous avez importé le bus d'événements par défaut dans une pile, vous pouvez mettre à jour les propriétés du bus d'événements comme vous le souhaitez.

Pour importer une ressource existante dans une CloudFormation pile nouvelle ou existante, vous avez besoin des informations suivantes :

- Identifiant unique de la ressource à importer.

  Pour les bus d'événements par défaut, l'identifiant est `Name` puis la valeur de l'identifiant est`default`.
- Modèle qui décrit avec précision les propriétés actuelles de la ressource existante.

L'extrait de modèle ci-dessous contient une `AWS::Events::EventBus` ressource qui décrit les propriétés actuelles d'un bus d'événements par défaut. Dans cet exemple, le bus d'événements a été configuré pour utiliser a clé gérée par le client et DLQ pour le chiffrement au repos.

En outre, la `AWS::Events::EventBus` ressource qui décrit le bus d'événements par défaut que vous souhaitez importer doit inclure une `DeletionPolicy` propriété définie sur`Retain`.

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Description": "Default event bus import example",
    "Resources": {
        "defaultEventBus": {
            "Type" : "AWS::Events::EventBus",
            "DeletionPolicy": "Retain",
            "Properties" : {
                "Name" : "default",
                "KmsKeyIdentifier" : "KmsKeyArn",
                "DeadLetterConfig" : {
                    "Arn" : "DLQ_ARN"
                }
            }
        }
    }
}
```

Pour plus d'informations, consultez la section Intégration des ressources existantes dans CloudFormation la gestion dans le Guide de CloudFormation l'utilisateur.

Supprimer un bus d' EventBridge événements Amazon

Vous pouvez supprimer un bus d'événements personnalisé ou partenaire. Vous ne pouvez pas supprimer le bus d'événements par défaut. La suppression d'un bus d'événements entraîne la suppression des règles associées à ce bus d'événements.

Pour supprimer un bus d'événements à l'aide de la EventBridge console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Choisissez le bus d'événements que vous souhaitez supprimer.

4. Effectuez l’une des actions suivantes :
   - Sélectionnez Delete (Supprimer).
   - Choisissez le nom du bus d'événements.

     Sur la page des détails du bus d'événements, choisissez Supprimer.

Autorisations pour les bus d'événements Amazon EventBridge

Le bus d'événements par défaut de votre compte AWS n'autorise que les événements provenant d'un seul compte. Vous pouvez accorder des autorisations supplémentaires à un bus d'événements en y attachant une politique basée sur les ressources. Avec une politique basée sur les ressources, vous pouvez autoriser les appels d'API `PutEvents`, `PutRule` et `PutTargets` à partir d'un autre compte. Vous pouvez également utiliser les conditions IAM de la politique pour accorder des autorisations à une organisation, appliquer des balises ou filtrer les événements uniquement sur ceux provenant d'une règle ou d'un compte spécifique. Vous pouvez définir une politique basée sur les ressources pour un bus d'événements lors de sa création ou ultérieurement.

Les API EventBridge qui acceptent le paramètre `Name` d'un bus d'événements tel que `PutRule` ,`PutTargets`, `DeleteRule`, `RemoveTargets`, `DisableRule` et `EnableRule` acceptent également l'ARN du bus d'événements. Utilisez ces paramètres pour référencer les bus d'événements entre comptes ou entre régions via les API. Par exemple, vous pouvez appeler `PutRule` pour créer une règle sur un bus d'événements dans un autre compte sans avoir à assumer un rôle.

Vous pouvez attacher les exemples de politiques de la présente rubrique à un rôle IAM afin d'autoriser l'envoi d'événements vers un autre compte ou une autre région. Utilisez les rôles IAM pour définir les politiques de contrôle de l'organisation et les limites quant aux personnes autorisées à envoyer des événements depuis votre compte vers d'autres comptes. Nous recommandons de toujours utiliser des rôles IAM lorsque la cible d'une règle est un bus d'événements. Vous pouvez attacher des rôles IAM en utilisant des appels `PutTarget`. Pour en savoir plus sur la création d'une règle permettant d'envoyer des événements vers un autre compte ou une autre région, consultez Envoyer et recevoir des EventBridge événements Amazon entre AWS comptes.

Rubriques

- Exemple de politique : envoi d'événements à un bus d'événements dans le même compte
- Exemple de politique : envoi d'événements au même compte et restriction des mises à jour
- Exemple de politique : envoi d'événements uniquement à partir d'une règle spécifique au bus d'une autre région
- Exemple de politique : envoi d'événements uniquement à partir d'une région spécifique vers une autre région
- Exemple de politique : refus de l'envoi d'événements à partir de régions spécifiques

Gestion des autorisations pour les bus d'événements

Procédez comme suit pour modifier les autorisations pour un bus d'événements existant. Pour en savoir plus sur l'utilisation de AWS CloudFormation pour créer une politique de bus d'événements, consultez AWS::Events::EventBusPolicy.

Pour gérer les autorisations pour un bus d'événements existant

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation de gauche, choisissez Bus d'événements.
3. Dans Nom, choisissez le nom du bus d'événements pour lequel gérer les autorisations.

   Si une politique de ressource est attachée au bus d'événements, elle s'affiche.
4. Choisissez Gérer les autorisations, puis effectuez l'une des opérations suivantes :
   - Entrez la politique qui inclut les autorisations à accorder pour le bus d'événements. Vous pouvez coller une politique provenant d'une autre source ou entrer le code JSON de la politique.
   - Pour utiliser un modèle pour la politique, choisissez Charger un modèle. Modifiez la politique en fonction de votre environnement et ajoutez des actions supplémentaires que vous autorisez le principal de la politique à utiliser.
5. Choisissez Mettre à jour.

Le modèle fournit des exemples d'instructions de politique que vous pouvez personnaliser pour votre compte et votre environnement. Le modèle n'est pas une politique valide. Vous pouvez modifier le modèle en fonction de votre cas d'utilisation, ou copier l'un des exemples de politiques et le personnaliser.

Le modèle charge les politiques qui incluent un exemple expliquant comment accorder des autorisations à un compte pour utiliser l'action `PutEvents`, comment accorder des autorisations à une organisation et comment accorder des autorisations au compte pour gérer les règles du compte. Vous pouvez personnaliser le modèle pour votre compte spécifique, puis supprimer les autres sections du modèle. D'autres exemples de politiques sont inclus plus loin dans cette rubrique.

Si vous essayez de mettre à jour les autorisations pour le bus, mais que la politique contient une erreur, un message d'erreur indique le problème spécifique de la politique.

```
 ### Choose which sections to include in the policy to match your use case. ###
 ### Be sure to remove all lines that start with ###, including the ### at the end of
the line. ###

 ### The policy must include the following: ###

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

     ### To grant permissions for an account to use the PutEvents action, include the
following, otherwise delete this section: ###

     {

       "Sid": "AllowAccountToPutEvents",
       "Effect": "Allow",
       "Principal": {
         "AWS": "<ACCOUNT_ID>"
       },
       "Action": "events:PutEvents",
       "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default"
     },

     ### Include the following section to grant permissions to all members of your AWS
Organizations to use the PutEvents action ###

     {
       "Sid": "AllowAllAccountsFromOrganizationToPutEvents",
       "Effect": "Allow",
       "Principal": "*",
       "Action": "events:PutEvents",
       "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default",
```

```
        "Condition": {
          "StringEquals": {
            "aws:PrincipalOrgID": "o-yourOrgID"
          }
        }
      },

      ### Include the following section to grant permissions to the account to manage
the rules created in the account ###

      {
        "Sid": "AllowAccountToManageRulesTheyCreated",
        "Effect": "Allow",
        "Principal": {
          "AWS": "<ACCOUNT_ID>"
        },
        "Action": [
          "events:PutRule",
          "events:PutTargets",
          "events:DeleteRule",
          "events:RemoveTargets",
          "events:DisableRule",
          "events:EnableRule",
          "events:TagResource",
          "events:UntagResource",
          "events:DescribeRule",
          "events:ListTargetsByRule",
          "events:ListTagsForResource"],
        "Resource": "arn:aws:events:us-east-1:123456789012:rule/default",
        "Condition": {
          "StringEqualsIfExists": {
            "events:creatorAccount": "<ACCOUNT_ID>"
          }
        }
    }]
  }
```

Exemple de politique : envoi d'événements au bus par défaut dans un autre compte

L'exemple de politique suivant accorde au compte 111122223333 l'autorisation de publier des événements sur le bus d'événements par défaut du compte 123456789012.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
        "Sid": "sid1",
        "Effect": "Allow",
        "Principal": {"AWS":"arn:aws:iam::111112222333:root"},
        "Action": "events:PutEvents",
        "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/default"
        }
    ]
  }
```

Exemple de politique : envoi d'événements à un bus personnalisé dans un autre compte

L'exemple de politique suivant accorde au compte 111122223333 l'autorisation de publier des événements sur `central-event-bus` dans le compte 123456789012, mais uniquement pour les événements dont la valeur source est définie sur `com.exampleCorp.webStore` et `detail-type` défini sur `newOrderCreated`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WebStoreCrossAccountPublish",
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/central-event-bus",
      "Condition": {
        "StringEquals": {
          "events:detail-type": "newOrderCreated",
          "events:source": "com.exampleCorp.webStore"
        }
      }
    }
```

```
  ]
}
```

Exemple de politique : envoi d'événements à un bus d'événements dans le même compte

L'exemple de politique suivant attaché à un bus d'événements nommé `CustomBus1` autorise le bus d'événements à recevoir des événements provenant du même compte et de la même région.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutEvents"
      ],
      "Resource": [
        "arn:aws:events:us-east-1:123456789:event-bus/CustomBus1"
      ]
    }
  ]
}
```

Exemple de politique : envoi d'événements au même compte et restriction des mises à jour

L'exemple de politique suivant accorde au compte 123456789012 l'autorisation de créer, de supprimer, de mettre à jour, de désactiver et d'activer des règles, ainsi que d'ajouter ou de supprimer des cibles. Il limite les règles qui correspondent aux événements dont la source est `com.exampleCorp.webStore`, et il utilise `"events:creatorAccount": "${aws:PrincipalAccount}"` pour garantir que seul le compte 123456789012 peut modifier ces règles et ces cibles une fois qu'elles ont été créées.

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

```
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": [
        "events:PutRule",
        "events:DeleteRule",
        "events:DescribeRule",
        "events:DisableRule",
        "events:EnableRule",
        "events:PutTargets",
        "events:RemoveTargets"
      ],
      "Resource": "arn:aws:events:us-east-1:123456789012:rule/central-event-bus/*",
      "Condition": {
        "StringEqualsIfExists": {
          "events:creatorAccount": "${aws:PrincipalAccount}",
          "events:source": "com.exampleCorp.webStore"
        }
      }
    }
  ]
}
```

Exemple de politique : envoi d'événements uniquement à partir d'une règle spécifique au bus d'une autre région

L'exemple de politique suivant accorde au compte 111122223333 l'autorisation d'envoyer des événements correspondant à une règle nommée `SendToUSE1AnotherAccount` dans les régions Moyen-Orient (Bahreïn) et USA Ouest (Oregon) à un bus d'événements nommé `CrossRegionBus` dans la région USA Est (Virginie du Nord) dans le compte 123456789012. L'exemple de politique est ajouté au bus d'événements nommé `CrossRegionBus` dans le compte 123456789012. La politique autorise les événements uniquement s'ils correspondent à une règle spécifiée pour le bus d'événements dans le compte 111122223333. L'instruction `Condition` limite les événements aux seuls événements qui correspondent aux règles avec l'ARN de règle spécifié.

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

```
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:111112222333:rule/CrossRegionBus/
SendToUSE1AnotherAccount",
            "arn:aws:events:me-south-1:111112222333:rule/CrossRegionBus/
SendToUSE1AnotherAccount"
          ]
        }
      }
    }
  ]
}
```

Exemple de politique : envoi d'événements uniquement à partir d'une région spécifique vers une autre région

L'exemple de politique suivant accorde au compte 111122223333 l'autorisation d'envoyer tous les événements générés dans les régions Moyen-Orient (Bahreïn) et USA Ouest (Oregon) au bus d'événements nommé `CrossRegionBus` dans le compte 123456789012 de la région USA Est (Virginie du Nord). Le compte 111122223333 n'est pas autorisé à envoyer les événements générés dans une autre région.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossRegionEventsFromUSWest2AndMESouth1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
```

```
            "arn:aws:events:us-west-2:*:*",
            "arn:aws:events:me-south-1:*:*"
          ]
        }
      }
    }
  ]
}
```

Exemple de politique : refus de l'envoi d'événements à partir de régions spécifiques

L'exemple de politique suivant attaché à un bus d'événements nommé `CrossRegionBus` dans le compte 123456789012 accorde l'autorisation au bus d'événements de recevoir des événements du compte 111122223333, mais pas les événements générés dans la région USA Ouest (Oregon).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1AllowAnyEventsFromAccount111112222333",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111112222333:root"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus"
    },
    {
      "Sid": "2DenyAllCrossRegionUSWest2Events",
      "Effect": "Deny",
      "Principal": {
        "AWS": "*"
      },
      "Action": "events:PutEvents",
      "Resource": "arn:aws:events:us-east-1:123456789012:event-bus/CrossRegionBus",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:events:us-west-2:*:*"
          ]
        }
```

```
      }
    }
  ]
}
```

Génération d'un modèle AWS CloudFormation à partir d'un bus d'événements Amazon EventBridge

AWS CloudFormation vous permet de configurer et de gérer vos ressources AWS sur l'ensemble des comptes et des régions de manière centralisée et reproductible en traitant l'infrastructure comme du code. Pour ce faire, CloudFormation vous permet de créer des modèles qui définissent les ressources que vous souhaitez provisionner et gérer.

EventBridge vous permet de générer des modèles à partir des bus d'événements existants dans votre compte, afin de vous aider à démarrer rapidement le développement de modèles CloudFormation. En outre, EventBridge offre la possibilité d'inclure les règles associées à ce bus d'événements dans votre modèle. Vous pouvez ensuite utiliser ces modèles comme base pour créer des piles de ressources sous la gestion de CloudFormation.

Pour plus d'informations sur CloudFormation, consultez le Guide de l'utilisateur AWS CloudFormation.

> **Note**
>
> EventBridge n'inclut pas de règles gérées dans le modèle généré.

Vous pouvez également générer un modèle à partir d'une ou de plusieurs règles contenues dans un bus d'événements sélectionné.

Pour générer un modèle CloudFormation à partir d'un bus d'événements

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Choisissez le bus d'événements à partir duquel vous souhaitez générer un modèle CloudFormation.
4. Dans le menu Actions, choisissez Modèle CloudFormation, puis choisissez le format dans lequel vous souhaitez qu'EventBridge génère le modèle : JSON ou YAML.

EventBridge affiche le modèle, généré dans le format sélectionné. Par défaut, toutes les règles associées au bus d'événements sont incluses dans le modèle.

- Pour générer le modèle sans inclure de règles, désélectionnez Inclure des règles sur cet EventBus.

5. EventBridge vous donne la possibilité de télécharger le fichier modèle ou de copier le modèle dans le presse-papiers.

   - Pour télécharger le fichier modèle, choisissez Télécharger.
   - Pour copier le modèle dans le presse-papiers, choisissez Copier.

6. Pour quitter le modèle, choisissez Annuler.

Une fois que vous avez personnalisé votre modèle AWS CloudFormation en fonction de votre cas d'utilisation, vous pouvez l'utiliser pour créer des piles dans CloudFormation.

Considérations lors de l'utilisation de modèles CloudFormation générés à partir d'Amazon EventBridge

Tenez compte des facteurs suivants lorsque vous utilisez un modèle CloudFormation que vous avez généré à partir d'un bus d'événements :

- EventBridge n'inclut pas de mots de passe dans le modèle généré.

  Vous pouvez modifier le modèle pour y inclure des paramètres de modèle qui permettent aux utilisateurs de spécifier des mots de passe ou d'autres informations sensibles lorsqu'ils utilisent le modèle pour créer ou mettre à jour une pile CloudFormation.

  En outre, les utilisateurs peuvent utiliser Secrets Manager pour créer un secret dans la région souhaitée, puis modifier le modèle généré pour utiliser des paramètres dynamiques.

- Les cibles du modèle généré restent exactement telles qu'elles ont été spécifiées dans le bus d'événements d'origine. Cela peut entraîner des problèmes entre régions si vous ne modifiez pas correctement le modèle avant de l'utiliser pour créer des piles dans d'autres régions.

  De plus, le modèle généré ne créera pas automatiquement les cibles en aval.

EventBridge Événements Amazon

Un événement indique un changement dans un environnement, comme un environnement AWS , un service ou une application partenaire SaaS, l'une de vos applications ou l'un de vos services. Voici des exemples d'événements :

- Amazon EC2 génère un événement lorsqu'une instance passe de l'état d'attente à l'état en cours d'exécution.
- Amazon EC2 Auto Scaling génère des événements lorsqu'il lance des instances ou les résilie.
- AWS CloudTrail publie des événements lorsque vous effectuez des appels d'API.

Vous pouvez également configurer des événements planifiés qui sont générés de façon périodique.

Pour obtenir la liste des services qui génèrent des événements, y compris des exemples d'événements de chaque service, consultez Événements organisés par AWS les services et cliquez sur les liens dans le tableau.

Les événements sont représentés sous forme d'objets JSON. Leur structure et leurs champs de niveau supérieur sont tous similaires.

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 des exemples d'événements générés par AWS les services, voirÉvénements organisés par AWS les services.

Rubriques

- Référence sur la structure des événements
- Ajouter des EventBridge événements Amazon avec PutEvents
- Événements organisés par AWS les services
- Réception d'événements d'un partenaire SaaS avec Amazon EventBridge
- Débogage de la livraison d'événements

La vidéo suivante explique les principes de base des événements : Qu'est-ce qu'un événement ?

La vidéo suivante explique comment les événements se déroulent EventBridge : D'où viennent les événements

Référence sur la structure des événements

Les champs suivants apparaissent dans tous les événements transmis à un bus d'événements et comprennent les métadonnées de l'événement :

```
{
  "???": "0",
  "???": "UUID",
  "???": "event name",
  "???": "event source",
  "???": "ARN",
  "???": "timestamp",
  "???": "region",
  "???": [
    "ARN"
  ],
  "???": {
    JSON object
  }
}
```

version

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

id

Un UUID version 4 généré pour chaque événement. Vous pouvez utiliser `id` pour suivre les événements au fur et à mesure qu'ils passent par les règles pour atteindre les cibles.

detail-type

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

Les événements fournis par CloudTrail ont `AWS API Call via CloudTrail` pour valeur de`detail-type`.

source

Identifie le service qui a généré l'événement. Tous les événements provenant des services AWS commencent par « aws ». Les événements générés par un client peuvent avoir n'importe quelle valeur ici tant qu'elle ne commence pas par « aws ». Nous recommandons l'utilisation de chaînes Java domaine-nom inversées de style nom de package.

Pour trouver la valeur correcte `source` pour un AWS service, consultez le tableau des clés de condition, sélectionnez un service dans la liste et recherchez le préfixe du service. Par exemple, la `source` valeur pour Amazon CloudFront est`aws.cloudfront`.

compte

Le numéro à 12 chiffres identifiant un AWS compte.

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 signaler l'heure de début, donc cette valeur peut être antérieure à l'heure de réception de l'événement.

region

Identifie la AWS région d'origine de l'événement.

resources

Un tableau JSON qui contient des ARN qui identifient les ressources liées à l'événement. Le service qui génère l'événement détermine s'il faut inclure ces ARN. 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 qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ. Il peut être défini sur `"{}"`.

AWS Les événements d'appel d'API comportent des objets détaillés contenant environ 50 champs imbriqués à plusieurs niveaux de profondeur.

> **Note**
>
> PutEventsaccepte les données au format JSON. Pour le type de données numérique (entier) JSON, les contraintes sont les suivantes : une valeur minimale de -9 223 372 036 854 775 808 et une valeur maximale de 9 223 372 036 854 775 807.

Example Exemple : Notification de changement d'état de l'instance Amazon EC2

L'événement suivant sur Amazon EventBridge indique la résiliation d'une instance Amazon EC2.

```
{
  "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"
  }
}
```

Informations minimales nécessaires pour un événement personnalisé valide

Lorsque vous créez des événements personnalisés, ceux-ci doivent inclure les champs suivants :

- `detail`
- `detail-type`
- `source`

```
{
  "detail-type": "event name",
  "source": "event source",
  "detail": {
```

```
  }
}
```

Ajouter des EventBridge événements Amazon avec **`PutEvents`**

L'`PutEvents`action envoie plusieurs événements EventBridge dans une seule demande. Pour plus d'informations, consultez PutEventsla référence des EventBridge API Amazon et put-events dans la référence des AWS CLI commandes.

Chaque demande `PutEvents` peut prendre en charge un nombre d'entrées limité. Pour plus d'informations, consultez Quotas Amazon EventBridge. L'opération `PutEvents` tente de traiter toutes les entrées dans l'ordre naturel de la demande. Après avoir appelé`PutEvents`, EventBridge attribue un identifiant unique à chaque événement.

Rubriques

- Traitement des échecs avec PutEvents
- Envoi d'événements à l'aide du AWS CLI
- Calcul de la taille de EventBridge PutEvents l'entrée d'un événement Amazon

L'exemple de code Java suivant envoie deux événements identiques à EventBridge.

AWS SDK for Java Version 2.x

```
EventBridgeClient eventBridgeClient =
    EventBridgeClient.builder().build();

PutEventsRequestEntry requestEntry = PutEventsRequestEntry.builder()
    .resources("resource1", "resource2")
    .source("com.mycompany.myapp")
    .detailType("myDetailType")
    .detail("{ \"key1\": \"value1\", \"key2\": \"value2\" }")
    .build();

List <
PutEventsRequestEntry > requestEntries = new ArrayList <
PutEventsRequestEntry > ();
requestEntries.add(requestEntry);

PutEventsRequest eventsRequest = PutEventsRequest.builder()
    .entries(requestEntries)
```

```
    .build();

PutEventsResponse result = eventBridgeClient.putEvents(eventsRequest);

for (PutEventsResultEntry resultEntry: result.entries()) {
    if (resultEntry.eventId() != null) {
        System.out.println("Event Id: " + resultEntry.eventId());
    } else {
        System.out.println("PutEvents failed with Error Code: " +
 resultEntry.errorCode());
    }
}
```

AWS SDK for Java Version 1.0

```
EventBridgeClient eventBridgeClient =
    EventBridgeClient.builder().build();

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());
    }
}
```

Après avoir exécuté ce code, le résultat de `PutEvents` inclut un tableau d'entrées de réponse. Chaque entrée de ce tableau correspond à une entrée du tableau de demandes dans l'ordre, du

début à la fin de la demande et de la réponse. Le tableau de réponse `Entries` comprend toujours le même nombre d'entrées que le tableau de demande.

Traitement des échecs avec **`PutEvents`**

Par défaut, en cas d'échec d'une entrée individuelle dans une demande, EventBridge le traitement du reste des entrées de la demande est poursuivi. Un tableau `Entries` de réponses peut inclure à la fois des entrées réussies et infructueuses. Vous devez détecter les entrées infructueuses et les inclure dans un appel ultérieur.

Les entrées réussies incluent une valeur `Id` et les entrées infructueuses incluent des valeurs `ErrorCode` et `ErrorMessage`. `ErrorCode` décrit le type d'erreur. `ErrorMessage` fournit plus d'informations sur l'erreur. L'exemple suivant comporte trois entrées de résultat pour une demande `PutEvents`. La deuxième entrée est infructueuse.

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

> **Note**
>
> Si vous `PutEvents` publiez un événement dans un bus d'événements qui n'existe pas, la mise en correspondance des EventBridge événements ne trouvera pas de règle correspondante et supprimera l'événement. Bien qu' EventBridge il envoie une `200` réponse, il n'échouera pas à la demande et n'inclura pas l'événement dans la `FailedEntryCount` valeur de la réponse à la demande.

Vous pouvez inclure les entrées qui sont infructueuses dans les demandes `PutEvents` ultérieures. Tout d'abord, pour savoir si la demande comporte des entrées ayant échoué, vérifiez le paramètre `FailedRecordCount` dans `PutEventsResult`. Si sa valeur est différente de zéro, vous pouvez ajouter chaque `Entry` qui comporte un `ErrorCode` non nul à une demande ultérieure. L'exemple suivant représente un gestionnaire d'échec.

```
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 du AWS CLI

Vous pouvez utiliser le AWS CLI pour envoyer des événements personnalisés EventBridge afin qu'ils puissent être traités. L'exemple suivant place un événement personnalisé dans EventBridge :

```
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 JSON qui contient des événements personnalisés.

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

Ensuite, pour utiliser le AWS CLI pour lire les entrées de ce fichier et envoyer des événements, à l'invite de commande, tapez :

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

Calcul de la taille de EventBridge PutEvents l'entrée d'un événement Amazon

Vous pouvez envoyer des événements personnalisés à EventBridge en utilisant l'`PutEvents`action. Vous pouvez regrouper plusieurs entrées d'événement en une seule demande pour plus d'efficacité. La taille totale d'une entrée doit être inférieure à 256 Ko. Vous pouvez calculer la taille d'une entrée avant d'envoyer les événements.

> **Note**
>
> La limite de taille est imposée pour l'entrée. Même si la taille de l'entrée est inférieure à la limite de taille, l'événement EventBridge est toujours supérieur à la taille de l'entrée en raison des caractères et des clés nécessaires à la représentation JSON de l'événement. Pour plus d'informations, consultez EventBridge Événements Amazon.

EventBridge calcule la `PutEventsRequestEntry` taille comme suit :

- Si le paramètre `Time` est spécifié, sa taille est de 14 octets.
- La taille des paramètres `Source` et `DetailType` correspond au nombre d'octets de leur forme codée en UTF-8.
- Si le paramètre `Detail` est spécifié, sa taille correspond au nombre d'octets de sa forme codée en UTF-8.
- Si le paramètre `Resources` est spécifié, chacune de ses entrées correspond au nombre d'octets de sa forme codée en UTF-8.

L'exemple de code Java 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;
}
```

> **Note**
>
> Si la taille d'une entrée est supérieure à 256 Ko, nous vous recommandons de charger l'événement dans un compartiment Amazon S3 et d'inclure `Object URL` dans l'entrée `PutEvents`.

Événements organisés par AWS les services

De nombreux AWS services génèrent des événements qui EventBridge reçoivent. Lorsqu'un AWS service de votre compte émet un événement, celui-ci est redirigé vers le bus d'événements par défaut de votre compte.

Livraison d'événements par le biais des AWS services

Chaque AWS service qui génère des événements les envoie soit en EventBridge tant que meilleur effort, soit en tant que tentative de livraison durable.

- La fourniture de tous les efforts signifie que le service tente d'envoyer tous les événements à EventBridge, mais dans de rares cas, un événement peut ne pas être livré.
- Une livraison durable signifie que le service tentera avec succès d'organiser des événements EventBridge au moins une fois.

  EventBridge acceptera tous les événements valides dans des conditions normales. Dans les cas où les événements ne peuvent pas être livrés en raison d'une interruption de EventBridge service, ils seront réessayés ultérieurement par le AWS service pendant 24 heures au maximum.

Une fois qu'un événement est diffusé EventBridge, il le EventBridge met en conformité avec les règles, puis suit la politique de nouvelles tentatives et toute file d'attente de lettres mortes spécifiée pour la ou les cibles de l'événement.

Pour obtenir la liste des AWS services qui génèrent des événements, consultez???.

Accès aux événements AWS de service via AWS CloudTrail

AWS CloudTrail est un service qui enregistre automatiquement des événements tels que les appels AWS d'API. Vous pouvez créer des EventBridge règles qui utilisent les informations provenant de CloudTrail. Pour plus d'informations CloudTrail, voir Qu'est-ce que c'est AWS CloudTrail ? .

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

Pour enregistrer des événements d'une `detail-type` valeur égale à`AWS API Call via CloudTrail`, un CloudTrail parcours avec journalisation activé est requis.

Lorsque vous utilisez CloudTrail Amazon S3, vous devez configurer pour CloudTrail consigner les événements de données. Pour plus d'informations, consultez Activation de la journalisation des CloudTrail événements pour les compartiments et objets S3.

Certains événements survenus dans AWS les services peuvent être signalés à la EventBridge fois par le service lui-même et par CloudTrail. Par exemple, un appel d'API Amazon EC2 qui démarre ou arrête une instance génère EventBridge des événements ainsi que des événements via. CloudTrail

CloudTrail aide les appelants d'API et les propriétaires de ressources à recevoir des événements dans leurs compartiments Amazon S3 en créant des traces, et diffuse des événements aux appelants d'API via. EventBridge Les propriétaires des ressources, en plus des appelants d'API, peuvent surveiller les appels d'API entre comptes via. EventBridge CloudTraill'intégration avec EventBridge fournit un moyen pratique de définir des flux de travail automatisés basés sur des règles en réponse aux événements.

Vous ne pouvez pas utiliser les AWS événements d'appel de l'API Put*Events dont la taille est supérieure à 256 Ko comme modèles d'événements, car la taille maximale de toute requête Put*Events est de 256 Ko. Pour plus d'informations sur les appels d'API que vous pouvez utiliser, consultez la section Services et intégrations CloudTrail pris en charge.

Réception d'événements de gestion en lecture seule de la part des services AWS

Vous pouvez configurer des règles sur votre bus d'événements par défaut ou personnalisé pour recevoir les événements de gestion en lecture seule des AWS services via. CloudTrail Les événements de gestion fournissent une visibilité sur les opérations de gestion effectuées sur les ressources de votre AWS compte. Ils sont également connus sous le nom opérations de plan de contrôle. Pour plus d'informations, consultez Journalisation des événements de gestion dans le Guide de l'utilisateur CloudTrail .

Pour chaque règle sur les bus d'événements par défaut ou personnalisés, vous pouvez définir l'état de la règle afin de contrôler les types d'événements à recevoir :

- Désactivez la règle afin que les événements EventBridge ne correspondent pas à la règle.
- Activez la règle de manière à ce que les événements EventBridge correspondent à la règle, à l'exception des événements de AWS gestion en lecture seule transmis via. CloudTrail
- Activez la règle de manière à ce que tous les événements EventBridge correspondent à la règle, y compris les événements de gestion en lecture seule transmis via. CloudTrail

Les bus dédiés aux événements partenaires ne reçoivent pas AWS d'événements.

Voici quelques éléments à prendre en compte lorsque vous décidez de recevoir ou non des événements de gestion en lecture seule :

- Certains événements de gestion en lecture seule, tels que AWS Key Management Service `GetKeyPolicy` et`DescribeKey`, ou IAM `GetPolicy` et les `GetRole` événements, se produisent à un volume beaucoup plus élevé que les événements de changement classiques.
- Vous recevez peut-être déjà des événements de gestion en lecture seule, s'ils ne commencent pas par `Describe`, `Get` ou `List`. Par exemple, les événements issus des AWS STS API suivantes sont des événements de changement, même s'ils commencent par le verbe `Get` :
  - `GetFederationToken`
  - `GetSessionToken`

  Pour obtenir la liste des événements de gestion en lecture seule qui ne respectent pas le `Describe` ou la convention de `List` dénomination, par AWS service, voir. `Get` ???

Pour créer une règle qui reçoit des événements de gestion en lecture seule à l'aide de la CLI AWS

- Utilisez la commande `put-rule` pour créer ou mettre à jour la règle, en utilisant des paramètres pour :
  - Spécifier que la règle appartient au bus d'événements par défaut ou à un bus d'événements personnalisé spécifique
  - Définir l'état de la règle sur `ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS`

```
aws events put-rule --name "ruleForManagementEvents" --event-bus-name
"default" --state "ENABLED_WITH_ALL_CLOUDTRAIL_MANAGEMENT_EVENTS"
```

> **Note**
>
> L'activation d'une règle pour les événements CloudWatch de gestion est prise en charge via la AWS CLI et les AWS CloudFormation modèles uniquement.

Example

L'exemple suivant montre comment comparer des événements spécifiques. La bonne pratique consiste à définir une règle dédiée pour faire correspondre des événements spécifiques, dans un souci de clarté et de facilité de modification.

Dans ce cas, la règle dédiée correspond à l'événement de `AssumeRole` gestion de AWS Security Token Service.

```
{
    "source" : [ "aws.sts" ],
    "detail-type": ["AWS API Call via CloudTrail"],
    "detail" : {
        "eventName" : ["AssumeRole"]
    }
}
```

AWS services qui génèrent des événements

Le tableau suivant indique les AWS services qui génèrent des événements. Choisissez le nom du service pour obtenir plus d'informations sur la façon dont ce service et sa EventBridge collaboration sont combinés.

Chaque AWS service qui génère des événements les envoie soit en EventBridge tant que meilleur effort, soit en tant que tentative de livraison durable. Pour plus d'informations, consultez ???.

Ce tableau inclut une représentation des AWS services auxquels des événements sont envoyés EventBridge, mais il n'inclut pas tous les services. Pour les services non répertoriés qui envoient des événements à EventBridge, supposez que vous déployez tous les efforts nécessaires.

Service	Type de tentative
Alexa for Business	Dans la mesure du possible
AWS Account Management	Dans la mesure du possible
Amazon API Gateway	Dans la mesure du possible
AWS AppConfig	Dans la mesure du possible
Amazon AppFlow	Dans la mesure du possible
Application Auto Scaling	Dans la mesure du possible
AWS Profileur des coûts d'application	Dans la mesure du possible
AWS Application Migration Service	Dans la mesure du possible
Amazon Athena	Dans la mesure du possible
AWS Backup	Dans la mesure du possible
AWS Batch	Durable
Amazon Braket	Durable
AWS Certificate Manager	Dans la mesure du possible

Service	Type de tentative
Amazon Chime	Dans la mesure du possible
Amazon Cloud Directory	Dans la mesure du possible
AWS CloudFormation	Durable
Amazon CloudFront	Dans la mesure du possible
AWS CloudHSM	Dans la mesure du possible
Amazon CloudSearch	Dans la mesure du possible
AWS CloudShell	Dans la mesure du possible
Evénements de AWS CloudTrail	Dans la mesure du possible
Amazon CloudWatch	Durable
Informations sur les CloudWatch applications Amazon	Dans la mesure du possible
Amazon CloudWatch Internet Monitor	Dans la mesure du possible
Amazon CloudWatch Logs	Dans la mesure du possible
Amazon CloudWatch Synthetics	Dans la mesure du possible
AWS CodeArtifact	Durable
AWS CodeBuild	Dans la mesure du possible
AWS CodeCommit	Dans la mesure du possible
AWS CodeDeploy	Dans la mesure du possible
Amazon CodeGuru Profiler	Dans la mesure du possible
AWS CodePipeline	Dans la mesure du possible
AWS CodeStar	Dans la mesure du possible

Service	Type de tentative
CodeConnections	Dans la mesure du possible
Amazon Cognito Identity	Dans la mesure du possible
Groupes d’utilisateurs Amazon Cognito	Dans la mesure du possible
Amazon Cognito Sync	Dans la mesure du possible
AWS Config	Dans la mesure du possible
Amazon Connect	Dans la mesure du possible
Amazon Connect Voice ID	Dans la mesure du possible
AWS Control Tower	Dans la mesure du possible
AWS Database Migration Service	Dans la mesure du possible
AWS Data Exchange	Dans la mesure du possible
Amazon Data Lifecycle Manager	Dans la mesure du possible
AWS Data Pipeline	Dans la mesure du possible
AWS DataSync	Dans la mesure du possible
AWS Device Farm	Dans la mesure du possible
Amazon DevOps Guru	Dans la mesure du possible
AWS Direct Connect	Dans la mesure du possible
AWS Directory Service	Dans la mesure du possible
Amazon DynamoDB	Dans la mesure du possible
AWS Elastic Beanstalk	Dans la mesure du possible
Amazon Elastic Block Store	Dans la mesure du possible

Service	Type de tentative
Modifications du volume Amazon Elastic Block Store	Dans la mesure du possible
Amazon ElastiCache	Dans la mesure du possible
Amazon Elastic Compute Cloud (Amazon EC2)	Dans la mesure du possible
Amazon EC2 Auto Scaling	Dans la mesure du possible
Flottes d'Amazon EC2	Dans la mesure du possible
Interruption d'instance Spot Amazon EC2	Dans la mesure du possible
Amazon Elastic Container Registry	Dans la mesure du possible
Amazon Elastic Container Service	Durable
AWS Elastic Disaster Recovery	Dans la mesure du possible
Amazon Elastic File System	Dans la mesure du possible
Amazon Elastic Kubernetes Service	Dans la mesure du possible
Elastic Load Balancing	Dans la mesure du possible
Amazon Elastic MapReduce	Dans la mesure du possible
Amazon Elastic Transcoder	Dans la mesure du possible
AWS Elemental MediaConnect	Dans la mesure du possible
AWS Elemental MediaConvert	Durable
AWS Elemental MediaLive	Dans la mesure du possible
AWS Elemental MediaPackage	Dans la mesure du possible
AWS Elemental MediaStore	Durable
Amazon EMR	Dans la mesure du possible

Service	Type de tentative
Amazon EMR on EKS	Dans la mesure du possible
Amazon EMR sans serveur	Dans la mesure du possible
Règles de EventBridge planification d'Amazon	Durable
EventBridge Schémas Amazon	Dans la mesure du possible
AWS Fault Injection Service	Dans la mesure du possible
Forecast	Dans la mesure du possible
Amazon GameLift	Dans la mesure du possible
AWS Glue	Dans la mesure du possible
AWS Glue DataBrew	Dans la mesure du possible
AWS Ground Station	Dans la mesure du possible
Amazon GuardDuty	Dans la mesure du possible
AWS Health	Dans la mesure du possible
AWS HealthLake	Durable
AWS Identity and Access Management (JE SUIS)	Dans la mesure du possible
IAM Access Analyzer	Dans la mesure du possible
Amazon Inspector Classic	Dans la mesure du possible
Amazon Inspector	Dans la mesure du possible
AWS IoT	Dans la mesure du possible
AWS IoT Analytics	Durable
AWS IoT Greengrass V1	Dans la mesure du possible

Service	Type de tentative
AWS IoT Greengrass V2	Dans la mesure du possible
Amazon Interactive Video Service	Dans la mesure du possible
Amazon Kinesis	Dans la mesure du possible
Amazon Data Firehose	Dans la mesure du possible
AWS Key Management Service Suppression de CMK	Durable
AWS Key Management Service Rotation de la CMK	Dans la mesure du possible
AWS Key Management Service expiration du matériel clé importé	Dans la mesure du possible
AWS Lambda	Dans la mesure du possible
Amazon Location Service	Durable
Amazon Machine Learning	Dans la mesure du possible
Amazon Macie	Dans la mesure du possible
Amazon Managed Blockchain	Dans la mesure du possible
AWS Managed Services	Dans la mesure du possible
AWS Management Console Connectez-vous	Dans la mesure du possible
AWS Metering Marketplace	Dans la mesure du possible
AWS Migration Hub	Dans la mesure du possible
AWS Migration Hub Refactor Spaces	Dans la mesure du possible
AWS Surveillance	Dans la mesure du possible
AWS Network Manager	Dans la mesure du possible

Service	Type de tentative
Amazon OpenSearch Service	Dans la mesure du possible
AWS OpsWorks	Durable
AWS OpsWorks CM	Dans la mesure du possible
AWS Organizations	Dans la mesure du possible
Amazon Polly	Dans la mesure du possible
AWS Private Certificate Authority	Dans la mesure du possible
AWS Proton	Dans la mesure du possible
Amazon QLDB	Durable
Amazon QuickSight	Dans la mesure du possible
Amazon RDS	Dans la mesure du possible
AWS Corbeille	Dans la mesure du possible
Amazon Redshift	Durable
API de données Amazon Redshift	Dans la mesure du possible
Amazon Redshift sans serveur	Dans la mesure du possible
AWS Resource Access Manager	Dans la mesure du possible
AWS Resource Groups	Dans la mesure du possible
AWS Resource Groups Tagging API	Dans la mesure du possible
Amazon Route 53	Dans la mesure du possible
Amazon Route 53 Recovery Readiness	Dans la mesure du possible
Amazon SageMaker	Dans la mesure du possible

Service	Type de tentative
Savings Plans	Dans la mesure du possible
AWS Secrets Manager	Dans la mesure du possible
AWS Security Hub	Durable
AWS Security Token Service	Dans la mesure du possible
AWS Server Migration Service	Dans la mesure du possible
AWS Service Catalog	Dans la mesure du possible
AWS Signer	Durable
Amazon Simple Email Service	Dans la mesure du possible
Amazon Simple Storage Service (Amazon S3)	Durable
Amazon S3 Glacier	Dans la mesure du possible
Amazon S3 on Outposts	Dans la mesure du possible
Amazon Simple Queue Service	Dans la mesure du possible
Amazon Simple Notification Service	Dans la mesure du possible
Amazon Simple Workflow Service	Dans la mesure du possible
AWS Step Functions	Dans la mesure du possible
AWS Storage Gateway	Durable
AWS Support	Dans la mesure du possible
AWS Systems Manager	Dans la mesure du possible
Amazon Transcribe	Dans la mesure du possible
AWS Transfer Family	Dans la mesure du possible

Service	Type de tentative
AWS Transit Gateway	Dans la mesure du possible
Amazon Translate	Durable
AWS Trusted Advisor	Dans la mesure du possible
AWS WAF	Dans la mesure du possible
AWS WAF Régional	Dans la mesure du possible
AWS Well-Architected Tool	Dans la mesure du possible
Amazon WorkDocs	Dans la mesure du possible
Amazon WorkSpaces	Dans la mesure du possible
AWS X-Ray	Dans la mesure du possible

Événements de gestion générés par les AWS services

En général, les API qui génèrent des événements de gestion (ou en lecture seule) commencent par les instructions `Describe`, `Get` ou `List`. Le tableau ci-dessous répertorie les AWS services et les événements de gestion qu'ils génèrent qui ne respectent pas cette convention de dénomination. Pour plus d'informations sur les événements de gestion, consultez ???.

Événements de gestion qui ne commencent pas par **`Describe`**, **`Get`** ou **`List`**

Le tableau suivant répertorie les AWS services et les événements de gestion qu'ils génèrent qui ne respectent pas les conventions de dénomination habituelles consistant à commencer par `DescribeGet`, ou`List`.

Service	Nom de l'événement	Type d'événement
Alexa for Business	ResolveRoom	Appel d'API
Alexa for Business	SearchAddressBooks	Appel d'API

Service	Nom de l’événement	Type d’événement
Alexa for Business	SearchContacts	Appel d'API
Alexa for Business	SearchDevices	Appel d'API
Alexa for Business	SearchProfiles	Appel d'API
Alexa for Business	SearchRooms	Appel d'API
Alexa for Business	SearchSkillGroups	Appel d'API
Alexa for Business	SearchUsers	Appel d'API
IAM Access Analyzer	ValidatePolicy	Appel d'API
AWS AdSpace Chambres propres	BatchGetSchema	Appel d'API
AWS Amplify Générateur d'interface utilisateur	ExportComponents	Appel d'API
AWS Amplify Générateur d'interface utilisateur	ExportForms	Appel d'API
AWS Amplify Générateur d'interface utilisateur	ExportThemes	Appel d'API
Amazon OpenSearch Service	BatchGetCollection	Appel d'API
Amazon API Gateway	ExportApi	Appel d'API
AWS AppConfig	ValidateConfiguration	Appel d'API
Amazon AppFlow	RetrieveConnectorData	Appel d'API
Informations sur les CloudWatch applications Amazon	UpdateApplicationDashboardConfiguration	Appel d'API
Amazon Athena	BatchGetNamedQuery	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Athena	BatchGetPreparedStatement	Appel d'API
Amazon Athena	BatchGetQueryExecution	Appel d'API
Amazon Athena	CheckQueryCompatibility	Appel d'API
Amazon Athena	ExportNotebook	Appel d'API
AWS Auto Scaling	AreScalableTargetsRegistered	Appel d'API
AWS Auto Scaling	Test	Appel d'API
AWS Marketplace	SearchAgreements	Appel d'API
AWS Backup	CreateLegalHold	Appel d'API
AWS Backup	ExportBackupPlanTemplate	Appel d'API
AWS Backup gateway	TestHypervisorConfiguration	Appel d'API
AWS Billing and Cost Management	AWSPaymentInstrumentGateway.Obtenez	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.DescribeMakePaymentPage	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.DescribePaymentsDashboard	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAccountPreferences	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetAdvancePaySummary	Action de console

Service	Nom de l’événement	Type d’événement
AWS Billing and Cost Management	AWSPaymentPortalService.GetAsoBulkDownload	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetBillingContactAddress	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetDocuments	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetEligiblePaymentInstruments	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetEntitiesByIds	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetFundingDocuments	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetKybcValidationStatus	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetOneTimePasswordStatus	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentHistory	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileByArn	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileCurrencies	Action de console

Service	Nom de l’événement	Type d’événement
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfiles	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentProfileServiceProviders	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetPaymentsDue	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetRemittanceInformation	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTaxInvoiceMetadata	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTermsAndConditionsForProgramGroup	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetTransactionsHistory	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnappliedFunds	Action de console
AWS Billing and Cost Management	AWSPaymentPortalService.GetUnpaidInvoices	Action de console
AWS Billing and Cost Management	AWSPaymentPreferenceGateway.Obtenez	Action de console
AWS Billing and Cost Management	CancelBulkDownload	Action de console

Service	Nom de l’événement	Type d’événement
AWS Billing and Cost Management	DownloadCommercialInvoice	Action de console
AWS Billing and Cost Management	DownloadCsv	Action de console
AWS Billing and Cost Management	DownloadDoc	Action de console
AWS Billing and Cost Management	Télécharger ECSV ForBillin gPeriod	Action de console
AWS Billing and Cost Management	DownloadPaymentHistory	Action de console
AWS Billing and Cost Management	DownloadRegistrati onDocument	Action de console
AWS Billing and Cost Management	DownloadTaxInvoice	Action de console
AWS Billing and Cost Management	FindBankRedirectPaymentInst ruments	Action de console
AWS Billing and Cost Management	Trouvez un CSV ForBillin gPeriod	Action de console
AWS Billing and Cost Management	ValidateReportDestination	Action de console
AWS Billing and Cost Management	VerifyChinaPaymentEligibility	Action de console
Amazon Braket	SearchCompilations	Appel d'API
Amazon Braket	SearchDevices	Appel d'API
Amazon Braket	SearchQuantumTasks	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Connect Cases	BatchGetField	Appel d'API
Amazon Connect Cases	SearchCases	Appel d'API
Amazon Connect Cases	SearchRelatedItems	Appel d'API
Amazon Chime	RetrieveDataExports	Appel d'API
Amazon Chime	SearchChannels	Appel d'API
Amazon Chime SDK Identity	DeleteProfile	Evénement de service
Amazon Chime SDK Identity	DeleteWorkTalkAccount	Evénement de service
AWS Chambres propres	BatchGetSchema	Appel d'API
Amazon Cloud Directory	BatchRead	Appel d'API
Amazon Cloud Directory	LookupPolicy	Appel d'API
AWS CloudFormation	DetectStackDrift	Appel d'API
AWS CloudFormation	DetectStackResourceDrift	Appel d'API
AWS CloudFormation	DetectStackSetDrift	Appel d'API
AWS CloudFormation	EstimateTemplateCost	Appel d'API
AWS CloudFormation	ValidateTemplate	Appel d'API
AWS CloudShell	RedeemCode	Appel d'API
AWS CloudTrail	LookupEvents	Appel d'API
AWS CodeArtifact	ReadFromRepository	Appel d'API
AWS CodeArtifact	SearchPackages	Appel d'API
AWS CodeArtifact	VerifyResourcesExistForTagris	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS CodeBuild	BatchGetBuildBatches	Appel d'API
AWS CodeBuild	BatchGetBuilds	Appel d'API
AWS CodeBuild	BatchGetProjects	Appel d'API
AWS CodeBuild	BatchGetReportGroups	Appel d'API
AWS CodeBuild	BatchGetReports	Appel d'API
AWS CodeBuild	BatchPutCodeCoverages	Appel d'API
AWS CodeBuild	BatchPutTestCases	Appel d'API
AWS CodeBuild	RequestBadge	Evénement de service
AWS CodeCommit	BatchDescribeMergeConflicts	Appel d'API
AWS CodeCommit	BatchGetCommits	Appel d'API
AWS CodeCommit	BatchGetPullRequests	Appel d'API
AWS CodeCommit	BatchGetRepositories	Appel d'API
AWS CodeCommit	EvaluatePullRequestApprovalRules	Appel d'API
AWS CodeCommit	GitPull	Appel d'API
AWS CodeDeploy	BatchGetApplicationRevisions	Appel d'API
AWS CodeDeploy	BatchGetApplications	Appel d'API
AWS CodeDeploy	BatchGetDeploymentGroups	Appel d'API
AWS CodeDeploy	BatchGetDeploymentInstances	Appel d'API
AWS CodeDeploy	BatchGetDeployments	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS CodeDeploy	BatchGetDeploymentTargets	Appel d'API
AWS CodeDeploy	BatchGetOnPremises Instances	Appel d'API
Amazon CodeGuru Profiler	BatchGetFrameMetricData	Appel d'API
Amazon CodeGuru Profiler	SubmitFeedback	Appel d'API
AWS CodePipeline	PollForJobs	Appel d'API
AWS CodePipeline	PollForThirdPartyJobs	Appel d'API
CodeConnections	StartAppRegistrationHandsha ke	Appel d'API
CodeConnections	Commencer à AuthHandshake	Appel d'API
CodeConnections	ValidateHostWebhook	Appel d'API
Amazon CodeWhisperer	CreateCodeScan	Appel d'API
Amazon CodeWhisperer	CreateProfile	Appel d'API
Amazon CodeWhisperer	CreateUploadUrl	Appel d'API
Amazon CodeWhisperer	GenerateRecommendations	Appel d'API
Amazon CodeWhisperer	UpdateProfile	Appel d'API
Amazon Cognito Identity	LookupDeveloperIdentity	Appel d'API
Groupes d’utilisateurs Amazon Cognito	AdminGetDevice	Appel d'API
Groupes d’utilisateurs Amazon Cognito	AdminGetUser	Appel d'API

Service	Nom de l’événement	Type d’événement
Groupes d’utilisateurs Amazon Cognito	AdminListDevices	Appel d'API
Groupes d’utilisateurs Amazon Cognito	AdminListGroupsForUser	Appel d'API
Groupes d’utilisateurs Amazon Cognito	AdminListUserAuthEvents	Appel d'API
Groupes d’utilisateurs Amazon Cognito	Beta_Authorize_GET	Evénement de service
Groupes d’utilisateurs Amazon Cognito	Confirm_GET	Evénement de service
Groupes d’utilisateurs Amazon Cognito	ConfirmForgotPassword_OBTENIR	Evénement de service
Groupes d’utilisateurs Amazon Cognito	Error_GET	Evénement de service
Groupes d’utilisateurs Amazon Cognito	ForgotPassword_OBTENIR	Evénement de service
Groupes d’utilisateurs Amazon Cognito	IntrospectToken	Appel d'API
Groupes d’utilisateurs Amazon Cognito	Login_Error_POST	Evénement de service
Groupes d’utilisateurs Amazon Cognito	Login_GET	Evénement de service
Groupes d’utilisateurs Amazon Cognito	Mfa_GET	Evénement de service
Groupes d’utilisateurs Amazon Cognito	MfaOption_OBTENIR	Evénement de service

Service	Nom de l’événement	Type d’événement
Groupes d’utilisateurs Amazon Cognito	ResetPassword_OBTENIR	Evénement de service
Groupes d’utilisateurs Amazon Cognito	Signup_GET	Evénement de service
Groupes d’utilisateurs Amazon Cognito	UserInfo_OBTENIR	Evénement de service
Groupes d’utilisateurs Amazon Cognito	UserInfo_PUBLIER	Evénement de service
Amazon Cognito Sync	BulkPublish	Appel d'API
Amazon Comprehend	BatchContainsPiiEntities	Appel d'API
Amazon Comprehend	BatchDetectDominantLanguage	Appel d'API
Amazon Comprehend	BatchDetectEntities	Appel d'API
Amazon Comprehend	BatchDetectKeyPhrases	Appel d'API
Amazon Comprehend	BatchDetectPiiEntities	Appel d'API
Amazon Comprehend	BatchDetectSentiment	Appel d'API
Amazon Comprehend	BatchDetectSyntax	Appel d'API
Amazon Comprehend	BatchDetectTargetedSentiment	Appel d'API
Amazon Comprehend	ClassifyDocument	Appel d'API
Amazon Comprehend	ContainsPiiEntities	Appel d'API
Amazon Comprehend	DetectDominantLanguage	Appel d'API
Amazon Comprehend	DetectEntities	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Comprehend	DetectKeyPhrases	Appel d'API
Amazon Comprehend	DetectPiiEntities	Appel d'API
Amazon Comprehend	DetectSentiment	Appel d'API
Amazon Comprehend	DetectSyntax	Appel d'API
Amazon Comprehend	DetectTargetedSentiment	Appel d'API
Amazon Comprehend	DetectToxicContent	Appel d'API
AWS Compute Optimizer	ExportAutoScalingGroupRecommendations	Appel d'API
AWS Compute Optimizer	Exporter BS VolumeRecommendations	Appel d'API
AWS Compute Optimizer	Exporter C InstanceRecommendations	Appel d'API
AWS Compute Optimizer	Exporter des EC ServiceRecommendations	Appel d'API
AWS Compute Optimizer	ExportLambdaFunctionRecommendations	Appel d'API
AWS Compute Optimizer	Exporter des RDS InstanceRecommendations	Appel d'API
AWS Config	BatchGetAggregateResourceConfig	Appel d'API
AWS Config	BatchGetResourceConfig	Appel d'API
AWS Config	SelectAggregateResourceConfig	Appel d'API
AWS Config	SelectResourceConfig	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Connect	AdminGetEmergencyAccessToken	Appel d'API
Amazon Connect	SearchQueues	Appel d'API
Amazon Connect	SearchRoutingProfiles	Appel d'API
Amazon Connect	SearchSecurityProfiles	Appel d'API
Amazon Connect	SearchUsers	Appel d'API
AWS Glue DataBrew	SendProjectSessionAction	Appel d'API
AWS Data Pipeline	EvaluateExpression	Appel d'API
AWS Data Pipeline	QueryObjects	Appel d'API
AWS Data Pipeline	ValidatePipelineDefinition	Appel d'API
AWS DataSync	VerifyResourcesExistForTagris	Appel d'API
AWS DeepLens	BatchGetDevice	Appel d'API
AWS DeepLens	BatchGetModel	Appel d'API
AWS DeepLens	BatchGetProject	Appel d'API
AWS DeepLens	CreateDeviceCertificates	Appel d'API
AWS DeepRacer	AdminGetAccountConfig	Appel d'API
AWS DeepRacer	AdminListAssociatedUsers	Appel d'API
AWS DeepRacer	TestRewardFunction	Appel d'API
AWS DeepRacer	VerifyResourcesExistForTagris	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Detective	BatchGetGraphMemberDatasources	Appel d'API
Amazon Detective	BatchGetMembershipDatasources	Appel d'API
Amazon Detective	SearchGraph	Appel d'API
Amazon DevOps Guru	SearchInsights	Appel d'API
Amazon DevOps Guru	SearchOrganizationInsights	Appel d'API
AWS Database Migration Service	BatchStartRecommendations	Appel d'API
AWS Database Migration Service	ModifyRecommendation	Appel d'API
AWS Database Migration Service	StartRecommendations	Appel d'API
AWS Database Migration Service	VerifyResourcesExistForTagris	Appel d'API
AWS Directory Service	VerifyTrust	Appel d'API
Amazon Elastic Compute Cloud	ConfirmProductInstance	Appel d'API
Amazon Elastic Compute Cloud	ReportInstanceStatus	Appel d'API
Amazon Elastic Container Registry	BatchCheckLayerAvailability	Appel d'API
Amazon Elastic Container Registry	BatchGetImage	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Elastic Container Registry	BatchGetImageReferrer	Appel d'API
Amazon Elastic Container Registry	BatchGetRepository ScanningConfiguration	Appel d'API
Amazon Elastic Container Registry	DryRunEvent	Evénement de service
Amazon Elastic Container Registry	PolicyExecutionEvent	Evénement de service
Amazon Elastic Container Registry Public	BatchCheckLayerAvailability	Appel d'API
Amazon Elastic Container Service	DiscoverPollEndpoint	Appel d'API
Amazon Elastic Container Service	FindSubfleetRoute	Appel d'API
Amazon Elastic Container Service	ValidateResources	Appel d'API
Amazon Elastic Container Service	VerifyTaskSetsExist	Appel d'API
Amazon Elastic Kubernetes Service	AccessKubernetesApi	Appel d'API
AWS Elastic Beanstalk	CheckDNSAvailability	Appel d'API
AWS Elastic Beanstalk	RequestEnvironmentInfo	Appel d'API
AWS Elastic Beanstalk	RetrieveEnvironmentInfo	Appel d'API
AWS Elastic Beanstalk	ValidateConfigurationSettings	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Elastic File System	NewClientConnection	Evénement de service
Amazon Elastic File System	UpdateClientConnection	Evénement de service
Amazon Elastic Transcoder	ReadJob	Appel d'API
Amazon Elastic Transcoder	ReadPipeline	Appel d'API
Amazon Elastic Transcoder	ReadPreset	Appel d'API
Amazon EventBridge	TestEventPattern	Appel d'API
Amazon EventBridge	TestScheduleExpression	Appel d'API
Amazon FinSpace API	BatchListCatalogNodesByDataset	Appel d'API
Amazon FinSpace API	BatchListNodesByDataset	Appel d'API
Amazon FinSpace API	BatchValidateAccess	Appel d'API
Amazon FinSpace API	CreateAuditRecordsQuery	Appel d'API
Amazon FinSpace API	SearchDatasets	Appel d'API
Amazon FinSpace API	SearchDatasetsV	Appel d'API
Amazon FinSpace API	ValidateIdToken	Appel d'API
AWS Firewall Manager	DisassociateAdminAccount	Appel d'API
Amazon Forecast	InvokeForecastEndpoint	Appel d'API
Amazon Forecast	QueryFeature	Appel d'API
Amazon Forecast	QueryForecast	Appel d'API
Amazon Forecast	QueryWhatIfForecast	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Forecast	VerifyResourcesExistForTagris	Appel d'API
Amazon Fraud Detector	BatchGetVariable	Appel d'API
Amazon Fraud Detector	VerifyResourcesExistForTagris	Appel d'API
FreeRTOS	VerifyEmailAddress	Appel d'API
Amazon GameLift	RequestUploadCredentials	Appel d'API
Amazon GameLift	ResolveAlias	Appel d'API
Amazon GameLift	SearchGameSessions	Appel d'API
Amazon GameLift	ValidateMatchmakingRuleSet	Appel d'API
Amazon GameSparks	ExportSnapshot	Appel d'API
Amazon Location Service	BatchGetDevicePosition	Appel d'API
Amazon Location Service	CalculateRoute	Appel d'API
Amazon Location Service	CalculateRouteMatrix	Appel d'API
Amazon Location Service	SearchPlaceIndexForPosition	Appel d'API
Amazon Location Service	SearchPlaceIndexForSuggestions	Appel d'API
Amazon Location Service	SearchPlaceIndexForText	Appel d'API
Amazon S3 Glacier	InitiateJob	Appel d'API
AWS Glue	BatchGetBlueprints	Appel d'API
AWS Glue	BatchGetColumnStatisticsForTable	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS Glue	BatchGetCrawlers	Appel d'API
AWS Glue	BatchGetCustomEntityTypes	Appel d'API
AWS Glue	BatchGetDataQualityResult	Appel d'API
AWS Glue	BatchGetDevEndpoints	Appel d'API
AWS Glue	BatchGetJobs	Appel d'API
AWS Glue	BatchGetTransformation ML	Appel d'API
AWS Glue	BatchGetPartition	Appel d'API
AWS Glue	BatchGetTriggers	Appel d'API
AWS Glue	BatchGetWorkflows	Appel d'API
AWS Glue	QueryJobRuns	Appel d'API
AWS Glue	QueryJobRunsAggregated	Appel d'API
AWS Glue	QueryJobs	Appel d'API
AWS Glue	QuerySchemaVersion Metadata	Appel d'API
AWS Glue	SearchTables	Appel d'API
AWS HealthLake	ReadResource	Appel d'API
AWS HealthLake	SearchWithGet	Appel d'API
AWS HealthLake	SearchWithPost	Appel d'API
AWS Identity and Access Management	GenerateCredentialReport	Appel d'API
AWS Identity and Access Management	GenerateOrganizationsAccess Report	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS Identity and Access Management	GenerateServiceLastAccessedDetails	Appel d'API
AWS Identity and Access Management	SimulateCustomPolicy	Appel d'API
AWS Identity and Access Management	SimulatePrincipalPolicy	Appel d'API
AWS Boutique d'identités	IsMemberInGroups	Appel d'API
AWS Authentification du magasin d'identités	BatchGetSession	Appel d'API
Amazon Inspector Classic	PreviewAgents	Appel d'API
Amazon Inspector Classic	BatchGetAccountStatus	Appel d'API
Amazon Inspector Classic	BatchGetFreeTrialInfo	Appel d'API
Amazon Inspector Classic	BatchGetMember	Appel d'API
Facturation AWS	ValidateDocumentDeliveryS3LocationInfo	Appel d'API
AWS IoT	SearchIndex	Appel d'API
AWS IoT	TestAuthorization	Appel d'API
AWS IoT	TestInvokeAuthorizer	Appel d'API
AWS IoT	ValidateSecurityProfileBehaviors	Appel d'API
AWS IoT Analytics	SampleChannelData	Appel d'API
AWS IoT SiteWise	GatewaysVerifyResourcesExistForTagrisInternal	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS IoT Things Graph	SearchEntities	Appel d'API
AWS IoT Things Graph	SearchFlowExecutions	Appel d'API
AWS IoT Things Graph	SearchFlowTemplates	Appel d'API
AWS IoT Things Graph	SearchSystemInstances	Appel d'API
AWS IoT Things Graph	SearchSystemTemplates	Appel d'API
AWS IoT Things Graph	SearchThings	Appel d'API
AWS IoT TwinMaker	ExecuteQuery	Appel d'API
AWS IoT Wireless	CreateNetworkAnalyzerConfiguration	Appel d'API
AWS IoT Wireless	DeleteNetworkAnalyzerConfiguration	Appel d'API
AWS IoT Wireless	DeregisterWirelessDevice	Appel d'API
Amazon Interactive Video Service	BatchGetChannel	Appel d'API
Amazon Interactive Video Service	BatchGetStreamKey	Appel d'API
Amazon Kendra	BatchGetDocumentStatus	Appel d'API
Amazon Kendra	Requête	Appel d'API
Service géré Amazon pour Apache Flink	DiscoverInputSchema	Appel d'API
AWS Key Management Service	Decrypt	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS Key Management Service	Encrypt	Appel d'API
AWS Key Management Service	GenerateDataKey	Appel d'API
AWS Key Management Service	GenerateDataKeyPair	Appel d'API
AWS Key Management Service	GenerateDataKeyPairWithoutPlaintext	Appel d'API
AWS Key Management Service	GenerateDataKeyWithoutPlaintext	Appel d'API
AWS Key Management Service	GenerateMac	Appel d'API
AWS Key Management Service	GenerateRandom	Appel d'API
AWS Key Management Service	ReEncrypt	Appel d'API
AWS Key Management Service	Sign (Signer)	Appel d'API
AWS Key Management Service	Vérification	Appel d'API
AWS Key Management Service	VerifyMac	Appel d'API
AWS Lake Formation	SearchDatabasesByMots-clés LF	Appel d'API
AWS Lake Formation	SearchTablesByMots-clés LF	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS Lake Formation	StartQueryPlanning	Appel d'API
Amazon Lex	BatchCreateCustomVocabularyItem	Appel d'API
Amazon Lex	BatchDeleteCustomVocabularyItem	Appel d'API
Amazon Lex	BatchUpdateCustomVocabularyItem	Appel d'API
Amazon Lex	DeleteCustomVocabulary	Appel d'API
Amazon Lex	SearchAssociatedTranscripts	Appel d'API
Amazon Lightsail	Créer une interface graphique SessionAccessDetails	Appel d'API
Amazon Lightsail	DownloadDefaultKeyPair	Appel d'API
Amazon Lightsail	IsVpcPeered	Appel d'API
Amazon CloudWatch Logs	FilterLogEvents	Appel d'API
Amazon Macie	BatchGetCustomDataIdentifiers	Appel d'API
Amazon Macie	UpdateFindingsFilter	Appel d'API
AWS Elemental MediaConnect	ManagedDescribeFlow	Appel d'API
AWS Elemental MediaConnect	PrivateDescribeFlowMeta	Appel d'API
AWS Application Migration Service	OperationalDescribeJobLogItems	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS Application Migration Service	OperationalDescribeJobs	Appel d'API
AWS Application Migration Service	OperationalDescribeReplicationConfigurationTemplates	Appel d'API
AWS Application Migration Service	OperationalDescribeSourceServer	Appel d'API
AWS Application Migration Service	OperationalGetLaunchConfiguration	Appel d'API
AWS Application Migration Service	OperationalListSourceServers	Appel d'API
AWS Application Migration Service	VerifyClientRoleForMgn	Appel d'API
AWS HealthOmics	VerifyResourceExists	Appel d'API
AWS HealthOmics	VerifyResourcesExistForTagris	Appel d'API
Amazon Polly	SynthesizeLongSpeech	Appel d'API
Amazon Polly	SynthesizeSpeech	Appel d'API
Amazon Polly	SynthesizeSpeechGet	Appel d'API
AWS service fournissant des réseaux privés gérés	Ping	Appel d'API
AWS Proton	DeleteEnvironmentTemplateVersion	Appel d'API
AWS Proton	DeleteServiceTemplateVersion	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon QLDB	ShowCatalog	Appel d'API
Amazon QuickSight	GenerateEmbedUrlFo rAnonymousUser	Appel d'API
Amazon QuickSight	GenerateEmbedUrlFo rRegisteredUser	Appel d'API
Amazon QuickSight	QueryDatabase	Evénement de service
Amazon QuickSight	SearchAnalyses	Appel d'API
Amazon QuickSight	SearchDashboards	Appel d'API
Amazon QuickSight	SearchDataSets	Appel d'API
Amazon QuickSight	SearchDataSources	Appel d'API
Amazon QuickSight	SearchFolders	Appel d'API
Amazon QuickSight	SearchGroups	Appel d'API
Amazon QuickSight	SearchUsers	Appel d'API
Amazon Relational Database Service	DownloadCompleteDB LogFile	Appel d'API
Amazon Relational Database Service	Télécharger DB LogFilePo rtion	Appel d'API
Amazon Rekognition	CompareFaces	Appel d'API
Amazon Rekognition	DetectCustomLabels	Appel d'API
Amazon Rekognition	DetectFaces	Appel d'API
Amazon Rekognition	DetectLabels	Appel d'API
Amazon Rekognition	DetectModerationLabels	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Rekognition	DetectProtectiveEquipment	Appel d'API
Amazon Rekognition	DetectText	Appel d'API
Amazon Rekognition	RecognizeCelebrities	Appel d'API
Amazon Rekognition	SearchFaces	Appel d'API
Amazon Rekognition	SearchFacesByImage	Appel d'API
Amazon Rekognition	SearchUsers	Appel d'API
Amazon Rekognition	SearchUsersByImage	Appel d'API
Explorateur de ressources AWS	BatchGetView	Appel d'API
Explorateur de ressources AWS	Search	Appel d'API
AWS Resource Groups	SearchResources	Appel d'API
AWS Resource Groups	ValidateResourceSharing	Appel d'API
AWS RoboMaker	BatchDescribeSimulationJob	Appel d'API
Amazon Route 53	TestDNSAnswer	Appel d'API
Amazon Route 53 Domaines	checkAvailabilities	Appel d'API
Amazon Route 53 Domaines	CheckDomainAvailability	Appel d'API
Amazon Route 53 Domaines	checkDomainTransferability	Appel d'API
Amazon Route 53 Domaines	CheckDomainTransferability	Appel d'API
Amazon Route 53 Domaines	isEmailReachable	Appel d'API
Amazon Route 53 Domaines	searchDomains	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon Route 53 Domaines	sendVerificationMessage	Appel d'API
Amazon Route 53 Domaines	ViewBilling	Appel d'API
Amazon Route 53 Domaines	viewBilling	Appel d'API
Amazon CloudWatch RUM	BatchGetRumMetricDefinitions	Appel d'API
Amazon Simple Storage Service	echo	Appel d'API
Amazon Simple Storage Service	GenerateInventory	Evénement de service
Amazon SageMaker	BatchDescribeModelPackage	Appel d'API
Amazon SageMaker	DeleteModelCard	Appel d'API
Amazon SageMaker	QueryLineage	Appel d'API
Amazon SageMaker	RenderUiTemplate	Appel d'API
Amazon SageMaker	Search	Appel d'API
EventBridge Schémas Amazon	ExportSchema	Appel d'API
EventBridge Schémas Amazon	SearchSchemas	Appel d'API
Amazon SimpleDB	DomainMetadata	Appel d'API
AWS Secrets Manager	ValidateResourcePolicy	Appel d'API
AWS Service Catalog	ScanProvisionedProducts	Appel d'API
AWS Service Catalog	SearchProducts	Appel d'API
AWS Service Catalog	SearchProductsAsAdmin	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS Service Catalog	SearchProvisionedProducts	Appel d'API
Amazon SES	BatchGetMetricData	Appel d'API
Amazon SES	TestRenderEmailTemplate	Appel d'API
Amazon SES	TestRenderTemplate	Appel d'API
Amazon Simple Notification Service	CheckIfPhoneNumber IsOptedOut	Appel d'API
AWS SQL Workbench	BatchGetNotebookCell	Appel d'API
AWS SQL Workbench	ExportNotebook	Appel d'API
Amazon EC2 Systems Manager	ExecuteApi	Appel d'API
AWS Systems Manager Incident Manager	DeleteContactChannel	Appel d'API
AWS IAM Identity Center	IsMemberInGroup	Appel d'API
AWS IAM Identity Center	SearchGroups	Appel d'API
AWS IAM Identity Center	SearchUsers	Appel d'API
AWS STS	AssumeRole	Appel d'API
AWS STS	AssumeRoleWithSAML	Appel d'API
AWS STS	AssumeRoleWithWebIdentity	Appel d'API
AWS STS	DecodeAuthorizationMessage	Appel d'API
AWS Réglages fiscaux	BatchGetTaxExemptions	Appel d'API
AWS WAFV2	CheckCapacity	Appel d'API

Service	Nom de l’événement	Type d’événement
AWS WAFV2	GenerateMobileSdkReleaseUrl	Appel d'API
AWS Well-Architected Tool	ExportLens	Appel d'API
AWS Well-Architected Tool	TagResource	Appel d'API
AWS Well-Architected Tool	UntagResource	Appel d'API
AWS Well-Architected Tool	UpdateGlobalSettings	Appel d'API
Amazon Connect Wisdom	QueryAssistant	Appel d'API
Amazon Connect Wisdom	SearchContent	Appel d'API
Amazon Connect Wisdom	SearchSessions	Appel d'API
Amazon WorkDocs	AbortDocumentVersionUpload	Appel d'API
Amazon WorkDocs	AddUsersToGroup	Appel d'API
Amazon WorkDocs	BatchGetUsers	Appel d'API
Amazon WorkDocs	CheckAlias	Appel d'API
Amazon WorkDocs	CompleteDocumentVersionUpload	Appel d'API
Amazon WorkDocs	CreateAnnotation	Appel d'API
Amazon WorkDocs	CreateComment	Appel d'API
Amazon WorkDocs	CreateFeedbackRequest	Appel d'API
Amazon WorkDocs	CreateFolder	Appel d'API
Amazon WorkDocs	CreateGroup	Appel d'API
Amazon WorkDocs	CreateShare	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon WorkDocs	CreateUser	Appel d'API
Amazon WorkDocs	DeleteAnnotation	Appel d'API
Amazon WorkDocs	DeleteComment	Appel d'API
Amazon WorkDocs	DeleteDocument	Appel d'API
Amazon WorkDocs	DeleteFeedbackRequest	Appel d'API
Amazon WorkDocs	DeleteFolder	Appel d'API
Amazon WorkDocs	DeleteFolderContents	Appel d'API
Amazon WorkDocs	DeleteGroup	Appel d'API
Amazon WorkDocs	DeleteOrganizationShare	Appel d'API
Amazon WorkDocs	DeleteUser	Appel d'API
Amazon WorkDocs	DownloadDocumentVersion	Appel d'API
Amazon WorkDocs	DownloadDocumentVersionUnderlays	Appel d'API
Amazon WorkDocs	InitiateDocumentVersionUpload	Appel d'API
Amazon WorkDocs	LogoutUser	Appel d'API
Amazon WorkDocs	PaginatedOrganizationActivity	Appel d'API
Amazon WorkDocs	PublishAnnotations	Appel d'API
Amazon WorkDocs	PublishComments	Appel d'API
Amazon WorkDocs	RestoreDocument	Appel d'API
Amazon WorkDocs	RestoreFolder	Appel d'API

Service	Nom de l’événement	Type d’événement
Amazon WorkDocs	SearchGroups	Appel d'API
Amazon WorkDocs	SearchOrganizationUsers	Appel d'API
Amazon WorkDocs	TransferUserResources	Appel d'API
Amazon WorkDocs	UpdateAnnotation	Appel d'API
Amazon WorkDocs	UpdateComment	Appel d'API
Amazon WorkDocs	UpdateDocument	Appel d'API
Amazon WorkDocs	UpdateDocumentVersion	Appel d'API
Amazon WorkDocs	UpdateFolder	Appel d'API
Amazon WorkDocs	UpdateGroup	Appel d'API
Amazon WorkDocs	UpdateOrganization	Appel d'API
Amazon WorkDocs	UpdateUser	Appel d'API
Amazon WorkMail	AssumeImpersonationRole	Appel d'API
Amazon WorkMail	QueryDnsRecords	Appel d'API
Amazon WorkMail	SearchMembers	Appel d'API
Amazon WorkMail	TestAvailabilityConfiguration	Appel d'API
Amazon WorkMail	TestInboundMailFlowRules	Appel d'API
Amazon WorkMail	TestOutboundMailFlowRules	Appel d'API

EventBridge référence détaillée des événements

EventBridge émet lui-même les événements suivants. Ces événements sont automatiquement envoyés au bus d'événements par défaut, comme pour tout autre AWS service.

Pour les définitions des champs de métadonnées inclus dans tous les événements, voirthe section called “Référence sur la structure des événements”.

Rubriques

Événement programmé

Vous trouverez ci-dessous les champs détaillés de l'`Scheduled Event`événement.

Les `detail-type` champs `source` et sont inclus car ils contiennent des valeurs spécifiques pour les EventBridge événements. Pour les définitions des autres champs de métadonnées inclus dans tous les événements, voirthe section called “Référence sur la structure des événements”.

```
{
  . . .,
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  . . .,
  "detail": {}
}
```

`detail-type`

Identifie le type d'événement.

Pour cet événement, cette valeur est`Scheduled Event`.

Obligatoire : oui

`source`

Identifie le service qui a généré l'événement. Pour les EventBridge événements, cette valeur est`aws.events`.

Obligatoire : oui

`detail`

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Obligatoire : oui

Il n'y a aucun champ obligatoire dans cet objet pour les `Scheduled Event` événements.

Example Exemple d'événement planifié

```
{
  "version": "0",
  "id": "89d1a02d-5ec7-412e-82f5-13505f849b41",
  "detail-type": "Scheduled Event",
  "source": "aws.events",
  "account": "123456789012",
  "time": "2016-12-30T18:44:49Z",
  "region": "us-east-1",
  "resources": ["arn:aws:events:us-east-1:123456789012:rule/SampleRule"],
  "detail": {}
}
```

Schéma créé

Vous trouverez ci-dessous les champs détaillés de l'`Schema Created`dévénement.

Lorsqu'un schéma est créé, il EventBridge envoie à la fois un `Schema Created` et un `Schema Version Created` événement.

Les `detail-type` champs `source` et sont inclus car ils contiennent des valeurs spécifiques pour les EventBridge événements. Pour les définitions des autres champs de métadonnées inclus dans tous les événements, voirthe section called “Référence sur la structure des événements”.

```
{
  . . .,
  "detail-type": "Schema Created",
  "source": "aws.schemas",
  . . .,
  "detail": {
    "SchemaName" : "String",
```

```
    "SchemaType" : "String",
    "RegistryName" : "String",
    "CreationDate" : "DateTime",
    "Version" : "Number"
  }
}
```

`detail-type`

Identifie le type d'événement.

Pour cet événement, cette valeur est`Schema Created`.

Obligatoire : oui

`source`

Identifie le service qui a généré l'événement. Pour les EventBridge événements, cette valeur est`aws.schemas`.

Obligatoire : oui

`detail`

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Obligatoire : oui

Pour cet événement, ces données incluent :

`SchemaName`

Nom du schéma.

Obligatoire : oui

`SchemaType`

Type de schéma.

Valeurs valides : `OpenApi3` | `JSONSchemaDraft4`

Obligatoire : oui

`RegistryName`

Nom du registre contenant le schéma.

Obligatoire : oui

`CreationDate`

Date à laquelle le schéma a été créé.

Obligatoire : oui

`Version`

Version du schéma.

Pour `Schema Created` les événements, cette valeur sera toujours`1`.

Obligatoire : oui

Example Exemple d'événement créé par un schéma

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Schema Created",
  "source": "aws.schemas",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
  "resources": ["arn:aws:schemas:us-east-1::schema/myRegistry/mySchema"],
  "detail": {
    "SchemaName": "mySchema",
    "SchemaType": "OpenApi3",
    "RegistryName": "myRegistry",
    "CreationDate": "2019-11-29T20:08:55Z",
    "Version": "1"
  }
}
```

Version du schéma créée

Vous trouverez ci-dessous les champs détaillés de l'`Schema Version Created`dévénement.

Lorsqu'un schéma est créé, il EventBridge envoie à la fois un `Schema Created` et un `Schema Version Created` événement.

Les `detail-type` champs `source` et sont inclus car ils contiennent des valeurs spécifiques pour les EventBridge événements. Pour les définitions des autres champs de métadonnées inclus dans tous les événements, voirthe section called “Référence sur la structure des événements”.

```
{
  . . .,
  "detail-type": "Schema Version Created",
  "source": "aws.schemas",
  . . .,
  "detail": {
    "SchemaName" : "String",
    "SchemaType" : "String",
    "RegistryName" : "String",
    "CreationDate" : "DateTime",
    "Version" : "Number"
  }
}
```

`detail-type`

Identifie le type d'événement.

Pour cet événement, cette valeur est`Schema Version Created`.

Obligatoire : oui

`source`

Identifie le service qui a généré l'événement. Pour les EventBridge événements, cette valeur est`aws.schemas`.

Obligatoire : oui

`detail`

Un objet JSON qui contient des informations sur l'événement. Le service qui génère l'événement détermine le contenu de ce champ.

Obligatoire : oui

Pour cet événement, ces données incluent :

`SchemaName`

Nom du schéma.

Obligatoire : oui

`SchemaType`

Type de schéma.

Valeurs valides : `OpenApi3` | `JSONSchemaDraft4`

Obligatoire : oui

`RegistryName`

Nom du registre contenant le schéma.

Obligatoire : oui

`CreationDate`

Date de création de la version du schéma.

Obligatoire : oui

`Version`

Version du schéma.

Obligatoire : oui

Example Exemple d'événement créé sous forme de version de schéma

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Schema Version Created",
  "source": "aws.schemas",
  "account": "123456789012",
  "time": "2019-05-31T21:49:54Z",
  "region": "us-east-1",
```

```
  "resources": ["arn:aws:schemas:us-east-1::schema/myRegistry/mySchema"],
  "detail": {
    "SchemaName": "mySchema",
    "SchemaType": "OpenApi3",
    "RegistryName": "myRegistry",
    "CreationDate": "2019-11-29T20:08:55Z",
    "Version": "5"
  }
}
```

Réception d'événements d'un partenaire SaaS avec Amazon EventBridge

Pour recevoir des événements des applications et services partenaires SaaS, vous avez besoin d'une source d'événement partenaire provenant du partenaire. Vous pouvez ensuite créer un bus d'événements partenaire et le faire correspondre à la source d'événement partenaire.

La vidéo suivante couvre les intégrations SaaS avec EventBridge : les partenaires du logiciel en tant que service (SaaS)

Rubriques

- Intégrations de partenaires SaaS prises en charge
- Configuration d'Amazon EventBridge pour recevoir des événements à partir d'une intégration SaaS
- Création d'une règle qui correspond à des événements partenaires SaaS
- Réception d'événements à l'aide d'URL de AWS Lambda fonctions
- Réception d'événements de Salesforce

Intégrations de partenaires SaaS prises en charge

EventBridge prend en charge les intégrations de partenaires SaaS suivantes :

- Adobe
- Auth0
- Blitline
- BUIDLHub

- Buildkite
- CleverTap
- Datadog
- Epsagon
- Freshworks
- Genesys
- GS2
- Karte
- Kloudless
- Mackerel
- MongoDB
- New Relic
- OneLogin
- Opsgenie
- PagerDuty
- Payshield
- SaaSus Platform
- SailPoint
- Saviynt
- Segment
- Shopify
- SignalFx
- Site24x7
- Stax
- Stripe
- SugarCRM
- SugarCRM
- Symantec
- Thundra

- TriggerMesh
- Whispir
- Zendesk
- API partenaire Amazon Seller

Les sources d'événements partenaires sont disponibles dans les régions suivantes.

Code	Nom
us-east-1	USA Est (Virginie du Nord)
us-east-2	USA Est (Ohio)
us-west-1	USA Ouest (Californie du Nord)
us-west-2	US West (Oregon)
ca-central-1	Canada (Centre)
eu-central-1	Europe (Francfort)
eu-central-2	Europe (Zurich)
eu-west-1	Europe (Irlande)
eu-west-2	Europe (Londres)
eu-west-3	Europe (Paris)
eu-north-1	Europe (Stockholm)
eu-south-1	Europe (Milan)
eu-south-2	Europe (Espagne)
af-south-1	Afrique (Le Cap)
ap-south-1	Asie-Pacifique (Mumbai)
ap-south-2	Asie-Pacifique (Hyderabad)

Code	Nom
ap-east-1	Asie-Pacifique (Hong Kong)
ap-northeast-1	Asie-Pacifique (Tokyo)
ap-northeast-2	Asie-Pacifique (Séoul)
ap-northeast-3	Asie-Pacifique (Osaka)
ap-southeast-1	Asie-Pacifique (Singapour)
ap-southeast-2	Asie-Pacifique (Sydney)
ap-southeast-3	Asie-Pacifique (Jakarta)
ap-southeast-4	Asie-Pacifique (Melbourne)
cn-north-1	Chine (Beijing)
cn-northwest-1	Chine (Ningxia)
me-central-1	Moyen-Orient (EAU)
me-south-1	Moyen-Orient (Bahreïn)
sa-east-1	Amérique du Sud (Sao Paulo)
il-central-1	Israël (Tel Aviv)

Configuration d'Amazon EventBridge pour recevoir des événements à partir d'une intégration SaaS

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Partner event sources (Sources d'événements partenaires).
3. Recherchez le partenaire de votre choix, puis choisissez Configurer pour ce partenaire.
4. Pour copier votre ID de compte dans le presse-papiers, choisissez Copier.

5. Dans le volet de navigation, choisissez Partner event sources (Sources d'événements partenaires).
6. Accédez au site web du partenaire et suivez les instructions pour créer une source d'événement partenaire à l'aide de votre ID de compte. La source d'événement que vous créez est disponible uniquement pour votre compte.
7. Revenez à la EventBridge console et choisissez Partner event sources dans le volet de navigation.
8. Sélectionnez le bouton en regard de la source d'événement partenaire, puis choisissez Associer au bus d'événements.

   Le statut de la source d'événement passe de `Pending` à `Active` et le nom du bus d'événements est mis à jour pour correspondre au nom de la source d'événement partenaire. Vous pouvez désormais commencer à créer des règles qui correspondent à des événements de la source d'événement partenaire. Pour plus d'informations, consultez Création d'une règle qui correspond à des événements partenaires SaaS.

   > **Note**
   >
   > Tout événement publié par un partenaire sur une source d'événements partenaire qui n'a pas été associé à un bus d'événements sera immédiatement supprimé. Ces événements ne se poursuivront pas pendant la pause. EventBridge

Création d'une règle qui correspond à des événements partenaires SaaS

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle.

   Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle corresponde aux événements provenant de votre compte, sélectionnez Bus d'événements par défaut AWS . Lorsqu'un service AWS de

votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.

6. Pour Type de règle, choisissez Règle avec un modèle d’événement.
7. Choisissez Suivant.
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. (Facultatif) Pour Exemples d’événements, choisissez le type d’événement.
10. Pour Modèle d’événement, entrez un modèle d’événement JSON.
11. Choisissez Suivant.
12. Pour Types de cibles, choisissez service AWS .
13. Pour Sélectionner une cible, choisissez le AWS service auquel vous souhaitez envoyer des informations en cas de EventBridge détection d'un événement correspondant au modèle d'événement.
14. Les champs affichés varient en fonction du service que vous choisissez. Entrez les informations spécifiques requises pour ce type de cible.
15. Pour de nombreux types de cibles, EventBridge nécessite des autorisations pour envoyer des événements à la cible. Dans ces cas, EventBridge vous pouvez créer le rôle IAM nécessaire à l'exécution de votre règle. Effectuez l’une des actions suivantes :

    - Pour créer un rôle IAM automatiquement, sélectionnez Create a new role for this specific resource.
    - Pour utiliser un rôle IAM que vous avez créé précédemment, choisissez Utiliser le rôle existant et sélectionnez le rôle existant dans la liste déroulante.

16. (Facultatif) Pour Additional settings (Paramètres supplémentaires), procédez comme suit :

    a. Pour Maximum age of event (Âge maximal de l'événement), saisissez une valeur comprise entre une minute (00:01) et 24 heures (24:00).
    b. Pour Retry attempts (Nouvelles tentatives), saisissez un nombre compris entre 0 et 185.
    c. Pour la file d'attente de lettres mortes, choisissez si vous souhaitez utiliser une file d'attente Amazon SQS standard comme file d'attente de lettres mortes. EventBridge envoie les événements correspondant à cette règle à la file d'attente des lettres mortes s'ils ne sont pas correctement transmis à la cible. Effectuez l’une des actions suivantes :
        - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
        - Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel

à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.

- Choisissez Sélectionnez une file d'attente Amazon SQS dans un autre AWS compte en tant que file d'attente de lettres mortes, puis entrez l'ARN de la file d'attente à utiliser. Vous devez associer à la file d'attente une politique basée sur les ressources qui EventBridge autorise l'envoi de messages. Pour plus d’informations, consultez Octroi d’autorisations à la file d’attente de lettres mortes.

17. (Facultatif) Sélectionnez Add another target (Ajouter une autre cible) pour ajouter une nouvelle cible pour cette règle.
18. Choisissez Suivant.
19. (Facultatif) Saisissez une ou plusieurs balises pour la règle. Pour plus d’informations, consultez EventBridge Balises Amazon.
20. Choisissez Suivant.
21. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Réception d'événements à l'aide d'URL de AWS Lambda fonctions

> **Note**
>
> Pour que le webhook entrant soit accessible à nos partenaires, nous créons un Open Lambda dans votre AWS compte qui est sécurisé au niveau de l'application Lambda en vérifiant la signature d'authentification envoyée par le partenaire tiers. Vérifiez cette configuration avec votre équipe de sécurité. Pour plus d'informations, consultez Modèle de sécurité et d'authentification pour les URL de fonctions Lambda.

Votre bus d' EventBridge événements Amazon peut utiliser une URL de AWS Lambda fonction créée par un AWS CloudFormation modèle pour recevoir des événements provenant de fournisseurs SaaS pris en charge. Avec les URL de fonction, les données d'événement sont envoyées à une fonction Lambda. La fonction convertit ensuite ces données en un événement qui peut être ingéré EventBridge et envoyé à un bus d'événements pour traitement. Une fois que l'événement est sur un bus d'événements, vous pouvez utiliser des règles pour filtrer les événements, appliquer les éventuelles transformations d'entrée configurées, puis les router vers la cible appropriée.

> **Note**
>
> La création d'URL de fonction Lambda augmentera vos coûts mensuels. Pour en savoir plus, consultez AWS Lambda Tarification.

Pour configurer une connexion EventBridge, vous devez d'abord sélectionner le fournisseur SaaS avec lequel vous souhaitez établir une connexion. Ensuite, vous fournissez un secret de signature que vous avez créé avec ce fournisseur et vous sélectionnez le bus d' EventBridge événements auquel envoyer les événements. Enfin, vous utilisez un AWS CloudFormation modèle et créez les ressources nécessaires pour terminer la connexion.

Les fournisseurs SaaS suivants sont actuellement disponibles pour utiliser les URL des EventBridge fonctions Lambda :

- GitHub
- Twilio

Rubriques

Configuration d’une connexion à GitHub

Étape 1 : Création de la AWS CloudFormation pile

Tout d'abord, utilisez la EventBridge console Amazon pour créer une CloudFormation pile :

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Quick Starts.
3. Sous Webhooks entrants utilisant des fURL Lambda, choisissez Démarrer.
4. Sous GitHub, choisissez Configurer.
5. Sous Étape 1 : Sélectionner un bus d’événements, sélectionnez un bus d’événements dans la liste déroulante. Ce bus d’événements reçoit les données de l’URL de fonction Lambda que vous fournissez à GitHub. Vous pouvez également créer un bus d’événements en sélectionnant Nouveau bus d’événements.
6. Dans Étape 2 : Configuration à l'aide de CloudFormation, choisissez Nouveau GitHub webhook.
7. Sélectionnez Je reconnais que le webhook entrant que je crée sera accessible au public et choisissez Confirmer.
8. Entrez un nom pour la pile.
9. Sous les paramètres, vérifiez que le bus d’événements correct est répertorié, puis spécifiez un jeton sécurisé pour GitHubWebhookSecret. Pour plus d’informations sur la création d’un jeton sécurisé, consultez Définition de votre jeton secret (langue française non garantie) dans la documentation de GitHub.
10. Sous Fonctionnalités et transformations, sélectionnez chacune des options suivantes :

    - Je reconnais que cela AWS CloudFormation peut créer des ressources IAM.

- Je reconnais que cela AWS CloudFormation peut créer des ressources IAM avec des noms personnalisés.
- Je reconnais que AWS CloudFormation cela peut nécessiter les capacités suivantes : **`CAPABILITY_AUTO_EXPAND`**

11. Sélectionnez Créer la pile.

Étape 2 : Créer un webhook GitHub

Créez ensuite le webhook sur GitHub. Vous aurez besoin du jeton sécurisé et de l'URL de fonction Lambda que vous avez créés à l'étape 2 pour terminer cette étape. Pour plus d'informations, consultez Création de webhooks (langue française non garantie) dans la documentation de GitHub.

Configuration d'une connexion à Twilio

Étape 1 : Rechercher votre jeton d'authentification Twilio

Pour établir une connexion entre Twilio et EventBridge, configurez d'abord la connexion Twilio avec le jeton d'authentification, ou secret, de votre Twilio compte. Pour plus d'informations, consultez Jetons d'authentification et comment les modifier (langue française non garantie) dans la documentation de Twilio.

Étape 2 : Création de la AWS CloudFormation pile

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Quick Starts.
3. Sous Webhooks entrants utilisant des fURL Lambda, choisissez Démarrer.
4. Sous Twilio, choisissez Configurer.
5. Sous Étape 1 : Sélectionner un bus d'événements, sélectionnez un bus d'événements dans la liste déroulante. Ce bus d'événements reçoit les données de l'URL de fonction Lambda que vous fournissez à Twilio. Vous pouvez également créer un bus d'événements en sélectionnant Nouveau bus d'événements.
6. Dans Étape 2 : Configuration à l'aide de CloudFormation, choisissez Nouveau Twilio webhook.
7. Sélectionnez Je reconnais que le webhook entrant que je crée sera accessible au public et choisissez Confirmer.
8. Entrez un nom pour la pile.

9. Sous les paramètres, vérifiez que le bus d'événements correct est répertorié, puis entrez le secret TwilioWebhookSecret que vous avez créé à l'étape 1.
10. Sous Fonctionnalités et transformations, sélectionnez chacune des options suivantes :
    - Je reconnais que cela AWS CloudFormation peut créer des ressources IAM.
    - Je reconnais que cela AWS CloudFormation peut créer des ressources IAM avec des noms personnalisés.
    - Je reconnais que cela AWS CloudFormation peut nécessiter la capacité suivante : CAPABILITY_AUTO_EXPAND
11. Sélectionnez Créer la pile.

Étape 3 : Créer un webhook Twilio

Après avoir configuré l'URL de fonction Lambda, vous devez la communiquer à Twilio afin que les données d'événements puissent être envoyées. Pour plus d'informations, consultez Configuration de votre URL publique avec Twilio (langue française non garantie) dans la documentation de Twilio.

Mise à jour du secret ou du jeton d'authentification d'un webhook

Mise à jour d'un secret GitHub

> **Note**
>
> GitHub ne permet pas d'avoir deux secrets en même temps. Il est possible que les ressources soient indisponibles lorsque le GitHub secret et le secret de la AWS CloudFormation pile ne sont pas synchronisés. GitHubles messages envoyés alors que les secrets ne sont pas synchronisés échoueront en raison de signatures incorrectes. Attendez que les CloudFormation secrets GitHub et soient synchronisés, puis réessayez.

1. Créez un secret GitHub. Pour plus d'informations, consultez Secrets chiffrés (langue française non garantie) dans la documentation de GitHub.
2. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
3. Choisissez Piles dans le volet de navigation.
4. Choisissez la pile du webhook qui inclut le secret que vous souhaitez mettre à jour.
5. Choisissez Mettre à jour.

6. Assurez-vous que l'option Utiliser modèle en cours est sélectionnée, puis choisissez Suivant.
7. Sous GitHubWebhookSecret, décochez Utiliser la valeur existante, entrez le nouveau GitHub secret que vous avez créé à l'étape 1, puis choisissez Suivant.
8. Choisissez Suivant.
9. Choisissez Mettre à jour la pile.

La propagation du secret peut prendre jusqu'à une heure. Pour réduire ce temps d'arrêt, vous pouvez actualiser le contexte d'exécution Lambda.

Mise à jour d'un secret Twilio

> **Note**
>
> Twilio ne permet pas d'avoir deux secrets en même temps. Il est possible que les ressources soient indisponibles lorsque le Twilio secret et le secret de la AWS CloudFormation pile ne sont pas synchronisés. Twilioles messages envoyés alors que les secrets ne sont pas synchronisés échoueront en raison de signatures incorrectes. Attendez que les CloudFormation secrets Twilio et soient synchronisés, puis réessayez.

1. Créez un secret Twilio. Pour plus d'informations, consultez Jetons d'authentification et comment les modifier (langue française non garantie) dans la documentation de Twilio.
2. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
3. Choisissez Piles dans le volet de navigation.
4. Choisissez la pile du webhook qui inclut le secret que vous souhaitez mettre à jour.
5. Choisissez Mettre à jour.
6. Assurez-vous que l'option Utiliser modèle en cours est sélectionnée, puis choisissez Suivant.
7. Sous TwilioWebhookSecret, décochez Utiliser la valeur existante, entrez le nouveau Twilio secret que vous avez créé à l'étape 1, puis choisissez Suivant.
8. Choisissez Suivant.
9. Choisissez Mettre à jour la pile.

La propagation du secret peut prendre jusqu'à une heure. Pour réduire ce temps d'arrêt, vous pouvez actualiser le contexte d'exécution Lambda.

Mise à jour d’une fonction Lambda

La fonction Lambda créée par la CloudFormation pile crée le webhook de base. Si vous souhaitez personnaliser la fonction Lambda pour un cas d'utilisation spécifique, tel que la journalisation personnalisée, utilisez la console pour accéder à la fonction, puis utilisez la CloudFormation console Lambda pour mettre à jour le code de la fonction Lambda.

Accès à la fonction Lambda

1. Ouvrez la AWS CloudFormation console à l'adresse https://console.aws.amazon.com/cloudformation.
2. Choisissez Piles dans le volet de navigation.
3. Choisissez la pile pour le webhook qui inclut la fonction Lambda que vous souhaitez mettre à jour.
4. Cliquez sur l’onglet Ressources.
5. Pour ouvrir la fonction Lambda dans la console Lambda, sous ID physique, choisissez l’ID de la fonction Lambda.

Maintenant que vous avez accédé à la fonction Lambda, utilisez la console Lambda pour mettre à jour le code de la fonction.

Mise à jour du code de la fonction Lambda

1. Sous Actions, choisissez Exporter la fonction.
2. Choisissez Télécharger un package de déploiement, puis enregistrez le fichier sur votre ordinateur.
3. Décompressez le fichier .zip du package de déploiement, mettez à jour le fichier `app.py`, puis compressez le package de déploiement à jour, en vous assurant que tous les fichiers du fichier .zip d’origine sont inclus.
4. Dans la console Lambda, cliquez sur l’onglet Code.
5. Sous Code source (Source du code), sélectionnez Upload from (Charger depuis).
6. Choisissez .zip file (fichier .zip), puis Upload (Charger).
   - Dans le sélecteur de fichiers, sélectionnez le fichier que vous avez mis à jour, choisissez Ouvrir, puis Enregistrer.
7. Sous Actions, choisissez Publier une nouvelle version.

Types d’événements disponibles

Les types d'événements suivants sont actuellement pris en charge par les bus CloudFormation d'événements :

- GitHub— Tous les types d'événements sont pris en charge.
- Twilio : les webhooks post-événement sont pris en charge.

Quotas, codes d’erreur et nouvelle tentative de livraison

Quotas

Le nombre de demandes entrantes adressées au webhook est plafonné par les services sous-jacents AWS . Le tableau suivant inclut les quotas correspondants.

Service	Quota
AWS Lambda	Par défaut : 10 exécutions simultanées Pour plus d’informations sur les quotas, notamment sur la demande d’augmentation des quotas, consultez Quotas Lambda.
AWS Secrets Manager	Par défaut : 5 000 demandes par seconde Pour plus d’informations sur les quotas, notamment sur la demande d’augmentation des quotas, consultez Quotas AWS Secrets Manager. **Note** Le nombre de demandes par seconde est réduit au maximum à l’aide du client de mise en cache AWS Secrets Manager Python.
Amazon EventBridge	Taille d'entrée maximale de 256 Ko pour les PutEvents actions.

Service	Quota
	EventBridge applique les quotas tarifaires basés sur les régions. Pour plus d'informations, consultez ???.

Codes d'erreur

Chaque AWS service renvoie des codes d'erreur spécifiques en cas d'erreur. Le tableau suivant inclut les codes d'erreur correspondants.

Service	Code d'erreur	Description
AWS Lambda	429 « TooManyRequestsException »	Le quota d'exécutions simultanées est dépassé.
AWS Secrets Manager	500 « Erreur de serveur interne »	Le quota de demandes par seconde est dépassé.
Amazon EventBridge	500 « Erreur de serveur interne »	Le quota de taux est dépassé pour la région.

Nouvelle livraison d'événements

En cas d'erreur, vous pouvez réessayer de livrer les événements concernés. Chaque fournisseur SaaS dispose de procédures de nouvelle tentative différentes.

GitHub

Utilisez l'API des webhooks GitHub pour le statut de livraison des appels webhooks et relivrer l'événement, si nécessaire. Pour plus d'informations, consultez la documentation de GitHub suivante :

- Organisation : Nouvelle livraison pour un webhook d'organisation (langue française non garantie)
- Référentiel : Nouvelle livraison pour un webhook de référentiel (langue française non garantie)
- Application : Nouvelle livraison pour un webhook d'application (langue française non garantie)

Twilio

Les utilisateurs de Twilio peuvent personnaliser les options de nouvelle tentative de livraison d'événement en utilisant des remplacements de connexion. Pour plus d'informations, consultez Webhooks (rappels HTTP) : Remplacements de connexion (langue française non garantie) dans la documentation de Twilio.

Réception d’événements de Salesforce

Vous pouvez utiliser Amazon EventBridge pour recevoir des événements Salesforce de différentes manières :

- En utilisant la fonction Salesforce's Event Bus Relay pour recevoir des événements directement sur le bus d'événements d'un EventBridge partenaire.
- En configurant un flux dans Amazon AppFlow qui est utilisé Salesforce comme source de données. Amazon envoie AppFlow ensuite Salesforce les événements à EventBridge l'aide d'un bus d'événements partenaire.

Vous pouvez envoyer des informations sur les événements à Salesforce l’aide de destinations d’API. Une fois que l’événement est envoyé à Salesforce, il peut être traité par des flux ou des déclencheurs Apex. Pour plus d’informations sur la configuration d’une destination d’API Salesforce, consultez ???.

Rubriques

Réception d’événements de Salesforce à l’aide d’Event Bus Relay

Étape 1 : configurer Salesforce Event Bus Relay et une source d'événements EventBridge partenaire

Lorsque vous créez une configuration de relais d'événements surSalesforce, Salesforce crée une source d'événements partenaire EventBridge dans l'état en attente.

Pour configurer Salesforce Event Bus Relay

1. Configuration d’un outil d’API REST (langue française non garantie)
2. (Facultatif) Définition d’un événement de plateforme (langue française non garantie)
3. Création d’une chaîne pour un événement de plateforme personnalisé (langue française non garantie)
4. Création d’un membre de chaîne pour associer l’événement de plateforme personnalisé (langue française non garantie)
5. Création d’informations d’identification nommées (langue française non garantie)
6. Création d’une configuration de relais d’événements (langue française non garantie)

Étape 2 : activer la source d'événements du Salesforce partenaire dans la EventBridge console et démarrer le relais d'événements

1. Ouvrez la page des sources d'événements partenaires dans la EventBridge console.
2. Sélectionnez la source d’événement partenaire Salesforce que vous avez créée à l’étape 1.
3. Choisissez Associer au bus d’événements.
4. Validez le nom du bus d’événements partenaire.
5. Choisissez Associer.
6. Démarrez le relais d’événements.

Maintenant que vous avez configuré et démarré le relais Event Bus et configuré la source d'événements partenaire, vous pouvez créer une EventBridge règle qui réagit aux événements pour filtrer et envoyer les données à une cible.

Réception d'événements liés à Salesforce l'utilisation d'Amazon AppFlow

Amazon AppFlow encapsule les événements depuis Salesforce une enveloppe d' EventBridge événements. L'exemple suivant montre un Salesforce événement reçu par un bus d'événements EventBridge partenaire.


```
{
    "version": "0",
    "id": "5c42b99e-e005-43b3-c744-07990c50d2cc",
    "detail-type": "AccountChangeEvent",
    "source": "aws.partner/appflow.test/salesforce.com/364228160620/CustomSF-Source-
Final",
    "account": "000000000",
    "time": "2020-08-20T18:25:51Z",
    "region": "us-west-2",
    "resources": [],
    "detail": {
        "ChangeEventHeader": {
            "commitNumber": 248197218874,
            "commitUser": "0056g000003XW7AAAW",
            "sequenceNumber": 1,
            "entityName": "Account",
            "changeType": "UPDATE",
            "changedFields": [
                "LastModifiedDate",
                "Region__c"
```

```
            ],
            "changeOrigin": "com/salesforce/api/soap/49.0;client=SfdcInternalAPI/",
            "transactionKey": "000035af-b239-0581-9f14-461e4187de11",
            "commitTimestamp": 1597947935000,
            "recordIds": [
                "0016g00000MLhLeAAL"
            ]
        },
        "LastModifiedDate": "2020-08-20T18:25:35.000Z",
        "Region__c": "America"
    }
}
```

Étape 1 : configurer Amazon AppFlow pour l'utiliser Salesforce comme source d'événements partenaire

Pour envoyer des événements à EventBridge, vous devez d'abord configurer Amazon AppFlow pour l'utiliser Salesforce comme source d'événements partenaire.

1. Dans la AppFlowconsole Amazon, choisissez Create flow.
2. Dans la section Détails du flux, dans Nom du flux, entrez le nom de votre flux.
3. (Facultatif) Entrez une description du flux, puis choisissez Suivant.
4. Sous Détails de la source, choisissez Salesforce dans le menu déroulant Nom de la source, puis choisissez Connexion pour créer une nouvelle connexion.
5. Dans la boîte de dialogue Connexion à Salesforce, choisissez Production ou Environnement de test (sandbox) pour l'environnement Salesforce.
6. Dans le champ Nom de la connexion, entrez un nom unique pour la connexion, puis choisissez Continuer.
7. Dans la boîte de dialogue Salesforce, procédez comme suit :

    a. Entrez vos informations d'identification de connexion Salesforce pour vous connecter à Salesforce.

    b. Sélectionnez Salesforce des événements pour les types de données AppFlow à traiter par Amazon.
8. Dans le menu déroulant Choisir un Salesforce événement, sélectionnez le type d'événement auquel envoyer EventBridge.
9. Pour une destination, sélectionnez Amazon EventBridge.
10. Sélectionnez Créer une nouvelle source d'événement partenaire.

11. (Facultatif) Spécifiez un suffixe unique pour la source d’événement partenaire.
12. Choisissez Générer une source d’événement partenaire.
13. Choisissez un compartiment Amazon S3 pour stocker les fichiers de charge utile d’événement dont la taille est supérieure à 256 Ko.
14. Dans la section Déclencheur de flux, assurez-vous que le paramètre Exécuter le flux en cas d’événement est sélectionné. Ce paramètre garantit que le flux est exécuté lorsqu’un nouvel événement Salesforce se produit.
15. Choisissez Suivant.
16. Pour le mappage de champs, sélectionnez Mapper directement tous les champs. Vous pouvez également sélectionner les champs qui vous intéressent dans la liste Nom de champ source.

    Pour plus d’informations sur le mappage de champs, consultez Mappage des champs de données.
17. Choisissez Suivant.
18. (Facultatif) Configurez les filtres pour les champs de données sur Amazon AppFlow.
19. Choisissez Suivant.
20. Passez en revue les paramètres, puis choisissez Créer le flux.

Une fois le flux configuré, Amazon AppFlow crée une nouvelle source d'événements partenaires que vous devez ensuite associer à un bus d'événements partenaire dans votre compte.

Étape 2 : Configuration EventBridge pour recevoir des Salesforce événements

Assurez-vous que le AppFlow flux Amazon déclenché par des Salesforce événements ayant EventBridge pour destination est configuré avant de suivre les instructions de cette section.

Pour configurer EventBridge la réception d'Salesforceévénements

1. Ouvrez la page des sources d'événements partenaires dans la EventBridge console.
2. Sélectionnez la source d’événement partenaire Salesforce que vous avez créée à l’étape 1.
3. Choisissez Associer au bus d’événements.
4. Validez le nom du bus d’événements partenaire.
5. Choisissez Associer.
6. Dans la AppFlow console Amazon, ouvrez le flux que vous avez créé et choisissez Activer le flux.

7. Ouvrez la page Règles dans la EventBridge console.
8. Choisissez Créer une règle.
9. Entrez un nom unique pour la règle.
10. Choisissez Modèle d'événement dans la section Définir un modèle.
11. Pour Modèle de correspondance d'événement, sélectionnez Modèle prédéfini par un service.
12. Dans la section Fournisseur de service, sélectionnez Tous les événements.
13. Pour Sélectionnez un bus d'événements, choisissez Bus d'événements personnalisé ou partenaire.
14. Sélectionnez le bus d'événements que vous avez associé à la source d'événements du AppFlow partenaire Amazon.
15. Pour Select targets, choisissez le AWS service qui doit agir lors de l'exécution de la règle. Une règle peut avoir jusqu'à cinq cibles.
16. Choisissez Créer.

Le service cible reçoit tous les événements Salesforce configurés pour votre compte. Pour filtrer les événements ou envoyer certains événements à différentes cibles, vous pouvez utiliser le filtrage basé sur le contenu avec des modèles d'événements.

> **Note**
>
> Pour les événements supérieurs à 256 Ko, Amazon AppFlow n'envoie pas l'événement complet à EventBridge. Amazon AppFlow place plutôt l'événement dans un compartiment S3 de votre compte, puis envoie un événement à l' EventBridge aide d'un pointeur vers le compartiment Amazon S3. Vous pouvez utiliser le pointeur pour obtenir l'événement complet depuis le compartiment.

Débogage de la livraison d'événements

Les problèmes de diffusion d'événements peuvent être difficiles à identifier. EventBridge Il existe plusieurs moyens de déboguer et de récupérer en cas d'échec de livraison d'événements.

Comment EventBridge réessaie d'organiser des événements

Parfois, un événement n'est pas correctement livré à la cible spécifiée dans une règle. Cela peut se produire, par exemple :

- Si la ressource cible n'est pas disponible
- En raison de l'état du réseau

Lorsqu'un événement n'est pas correctement transmis à une cible en raison d'erreurs récupérables, EventBridge réessaie d'envoyer l'événement. Vous définissez la durée des tentatives et le nombre de nouvelles tentatives dans les paramètres Politique de nouvelles tentatives de la cible. Par défaut, EventBridge réessaie d'envoyer l'événement pendant 24 heures et jusqu'à 185 fois avec un retard exponentiel ou un délai aléatoire.

Si un événement n'est pas délivré une fois toutes les tentatives épuisées, l'événement est abandonné et son traitement EventBridge ne se poursuit pas.

Utilisation de files d'attente de lettres mortes pour traiter les événements non livrés

Pour éviter de perdre des événements qui ne parviennent pas à être livrés à une cible, vous pouvez configurer une file d’attente de lettres mortes (DLQ) et lui envoyer tous les événements ayant échoué pour qu’ils soient traités ultérieurement.

EventBridge Les DLQ sont des files d'attente EventBridge Amazon SQS standard utilisées pour stocker des événements qui n'ont pas pu être transmis à une cible. Lorsque vous créez une règle et ajoutez une cible, vous pouvez choisir d’utiliser ou non une DLQ. Lorsque vous configurez une DLQ, vous pouvez conserver tous les événements qui n’ont pas été correctement livrés. Vous pouvez ensuite résoudre le problème ayant provoqué l’échec de la livraison des événements et traiter les événements ultérieurement.

Lorsque vous configurez une DLQ pour la cible d'une règle, EventBridge envoie les événements ayant échoué à la file d'attente Amazon SQS sélectionnée.

Les erreurs d’événement sont traitées de différentes façons. Certains événements sont supprimés ou envoyés à une DLQ sans effectuer de nouvelle tentative. Par exemple, pour les erreurs provoquées par l’absence d’autorisations sur une cible ou par une ressource cible qui n’existe plus, toutes les nouvelles tentatives échouent tant qu’une mesure n’est pas prise pour résoudre le problème sous-jacent. Plutôt que de réessayer, EventBridge envoie ces événements directement au DLQ, si vous en avez un.

En cas d'échec de la diffusion d'un événement, EventBridge publie un événement sur Amazon CloudWatch Metrics indiquant qu'un objectif `invocation` a échoué. Si vous utilisez un DLQ,

des métriques supplémentaires sont envoyées aux adresses suivantes : CloudWatch y compris `InvocationsSentToDLQ` et`InvocationsFailedToBeSentToDLQ`.

Vous pouvez également spécifier des DLQ pour les bus d'événements, si vous les utilisez AWS KMS clés gérées par le client pour chiffrer des événements au repos. Pour plus d'informations, consultez ???.

Chaque message de votre DLQ inclura les attributs personnalisés suivants :

- `RULE_ARN`
- `TARGET_ARN`
- `ERROR_CODE`

  Voici des exemples de code d'erreur qu'une DLQ peut renvoyer :

  - `CONNECTION_FAILURE`
  - `CROSS_ACCOUNT_INGESTION_FAILED`
  - `CROSS_REGION_INGESTION_FAILED`
  - `ERROR_FROM_TARGET`
  - `EVENTS_IN_BATCH_REQUEST_REJECTED`
  - `EVENTS_IN_BATCH_REQUEST_REJECTED`
  - `FAILED_TO_ASSUME_ROLE`
  - `INTERNAL_ERROR`
  - `INVALID_JSON`
  - `INVALID_PARAMETER`
  - `NO_PERMISSIONS`
  - `NO_RESOURCE`
  - `RESOURCE_ALREADY_EXISTS`
  - `RESOURCE_LIMIT_EXCEEDED`
  - `RESOURCE_MODIFICATION_COLLISION`
  - `SDK_CLIENT_ERROR`
  - `THIRD_ACCOUNT_HOP_DETECTED`
  - `THIRD_REGION_HOP_DETECTED`
  - `THROTTLING`

  - `TIMEOUT`
  - `TRANSIENT_ASSUME_ROLE`
  - `UNKNOWN`
- `ERROR_MESSAGE`
- `EXHAUSTED_RETRY_CONDITION`

  Les conditions suivantes peuvent être renvoyées :

  - `MaximumRetryAttempts`
  - `MaximumEventAgeInSeconds`
- `RETRY_ATTEMPTS`

La vidéo suivante décrit la configuration des DLQ : Utilisation des files d’attente de lettres mortes (DLQ)

Rubriques

- Considérations relatives à l’utilisation d’une file d’attente de lettres mortes
- Octroi d’autorisations à la file d’attente de lettres mortes
- Comment renvoyer des événements à partir d’une file d’attente de lettres mortes

Considérations relatives à l’utilisation d’une file d’attente de lettres mortes

Tenez compte des points suivants lors de la configuration d'un DLQ pour EventBridge.

- Seules les files d’attente standard sont prises en charge. Vous ne pouvez pas utiliser une file d'attente FIFO pour un DLQ dans. EventBridge
- EventBridge inclut les métadonnées de l'événement et les attributs du message dans le message, notamment : le code d'erreur, le message d'erreur, la condition de nouvelle tentative épuisée, l'ARN de la règle, les tentatives de nouvelle tentative et l'ARN cible. Vous pouvez utiliser ces valeurs pour identifier un événement et la cause de l’échec.
- Autorisations pour les DLQ dans le même compte :
  - Si vous ajoutez une cible à une règle à l'aide de la console et que vous choisissez une file d'attente Amazon SQS dans le même compte, une politique basée sur les ressources qui accorde l' EventBridge accès à la file d'attente est attachée à la file d'attente pour vous.

  - Si vous utilisez l' EventBridge API `PutTargets` pour ajouter ou mettre à jour une cible pour une règle, et que vous choisissez une file d'attente Amazon SQS dans le même compte, vous devez accorder manuellement des autorisations à la file d'attente sélectionnée. Pour en savoir plus, veuillez consulter la section Octroi d’autorisations à la file d’attente de lettres mortes.
- Autorisations d'utilisation des files d'attente Amazon SQS à partir d'un autre compte. AWS
  - Si vous créez une règle à partir de la console, les files d’attente des autres comptes ne sont pas affichées pour que vous puissiez les sélectionner. Vous devez fournir l’ARN de la file d’attente dans l’autre compte, puis attacher manuellement une politique basée sur les ressources pour accorder l’autorisation à la file d’attente. Pour en savoir plus, veuillez consulter la section Octroi d’autorisations à la file d’attente de lettres mortes.
  - Si vous créez une règle à l’aide de l’API, vous devez attacher manuellement une politique basée sur les ressources aux files d’attente SQS d’un autre compte utilisé comme file d’attente de lettres mortes. Pour en savoir plus, veuillez consulter la section Octroi d’autorisations à la file d’attente de lettres mortes.
- La file d’attente Amazon SQS que vous utilisez doit se trouver dans la même région que celle dans laquelle vous créez la règle.

Octroi d’autorisations à la file d’attente de lettres mortes

Pour transmettre correctement les événements à la file d'attente, vous EventBridge devez être autorisé à le faire. Lorsque vous spécifiez un DLQ à l'aide de la EventBridge console, les autorisations sont automatiquement ajoutées. Cela consiste notamment à :

- Lorsque vous configurez une DLQ pour la cible d'une règle.
- Lorsque vous configurez un DLQ pour un bus d'événements lorsque vous avez spécifié l' EventBridge utilisation d'un AWS KMS clé gérée par le client pour chiffrer les événements au repos.

  Pour plus d’informations, consultez ???.

Si vous spécifiez un DLQ à l'aide de l'API, ou si vous utilisez une file d'attente qui se trouve dans un autre AWS compte, vous devez créer manuellement une politique basée sur les ressources qui accorde les autorisations requises, puis l'associer à la file d'attente.

Exemple d'autorisations de file d'attente pour les lettres mortes

La politique basée sur les ressources suivante explique comment accorder les autorisations requises pour envoyer des messages d'événements EventBridge à une file d'attente Amazon SQS. L'exemple de politique accorde au EventBridge service l'autorisation d'utiliser l'`SendMessage`opération pour envoyer des messages à une file d'attente nommée « MyEvent DLQ ». La file d'attente doit se trouver dans la région us-west-2 sur le compte 123456789012. AWS La `Condition` déclaration autorise uniquement les demandes provenant d'une règle nommée « MyTestRule » créée dans la région us-west-2 sur le compte 123456789012. AWS

```
{
  "Sid": "Dead-letter queue permissions",
  "Effect": "Allow",
  "Principal": {
     "Service": "events.amazonaws.com"
  },
  "Action": "sqs:SendMessage",
  "Resource": "arn:aws:sqs:us-west-2:123456789012:MyEventDLQ",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:events:us-west-2:123456789012:rule/MyTestRule"
    }
  }
}
```

Exemple d'autorisations de file d'attente de lettres mortes dans un bus d'événements

La politique basée sur les ressources suivante montre comment accorder les autorisations requises lors de la spécification d'un DLQ pour un bus d'événements. Dans ce cas, `aws:SourceArn` spécifie l'ARN du bus d'événements qui envoie les événements au DLQ. Ici encore, dans cet exemple, la file d'attente doit se trouver dans la même région que le bus d'événements.

```
{
  "Sid": "Dead-letter queue permissions",
  "Effect": "Allow",
  "Principal": {
     "Service": "events.amazonaws.com"
  },
  "Action": "sqs:SendMessage",
  "Resource": "arn:aws:sqs:region:account-id:queue-name",
  "Condition": {
    "ArnEquals": {
      "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-arn"
```

```
    }
  }
}
```

Pour attacher la politique à la file d'attente, utilisez la console Amazon SQS, ouvrez la file d'attente, puis choisissez la stratégie d'accès et modifiez-la. Vous pouvez également utiliser AWS CLI. Pour en savoir plus, veuillez consulter la section Autorisations Amazon SQS.

Comment renvoyer des événements à partir d'une file d'attente de lettres mortes

Vous pouvez déplacer les messages hors d'une DLQ de deux façons :

- Éviter d'écrire une logique de consommateur Amazon SQS : définissez votre DLQ en tant que source d'évènement pour la fonction Lambda afin de purger votre DLQ.
- Rédigez la logique client Amazon SQS : utilisez l'API Amazon SQS AWS , le SDK AWS CLI ou écrivez une logique client personnalisée pour interroger, traiter et supprimer les messages dans le DLQ.

Modèles d' EventBridge événements Amazon

Les modèles d’événements ont la même structure que les événements auxquels ils correspondent. Les règles utilisent des modèles d’événements pour sélectionner des événements et les envoyer vers des cibles. Soit un modèle d’événement correspond à un événement, soit il n’y correspond pas.

> **Important**
>
> Dans EventBridge, il est possible de créer des règles pouvant entraîner des higher-than-expected frais et des ralentissements. Par exemple, vous pouvez créer par inadvertance une règle qui entraîne une boucle infinie, dans laquelle une règle est déclenchée de manière récursive sans fin. Supposons que vous avez créé une règle permettant de détecter que les listes ACL ont été modifiées sur un compartiment Amazon S3 et de déclencher 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 obtenir des conseils sur la façon d’écrire des règles et des modèles d’événements précis afin de réduire au maximum ces résultats inattendus, consultez ??? et ???.

La vidéo suivante présente les principes de base des modèles d’événements : Comment filtrer les événements

Rubriques

L'événement suivant montre un AWS événement simple provenant d'Amazon EC2.

```
{
  "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"
  }
}
```

Le modèle d'événement suivant traite tous les événements `instance-termination` Amazon EC2.

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

Création de modèles d'événements

Pour créer un modèle d'événement, vous spécifiez les champs d'un événement auxquels vous souhaitez que le modèle d'événement corresponde. Spécifiez uniquement les champs que vous utilisez pour la correspondance. L'exemple de modèle d'événement précédent ne fournit des valeurs que pour trois champs : les champs de niveau supérieur `"source"` et`"detail-type"`, et le `"state"` champ situé à l'intérieur du champ `"detail"` d'objet. EventBridge ignore tous les autres champs de l'événement lors de l'application de la règle.

Pour qu'un modèle d'événement corresponde à un événement, cet événement doit contenir tous les noms de champs figurant dans le modèle d'événement. Les noms de champs doivent également apparaître dans l'événement avec la même structure imbriquée.

Lorsque vous écrivez des modèles d'événements pour correspondre à des événements, vous pouvez utiliser l'API `TestEventPattern` ou la commande d'interface de ligne de commande `test-event-pattern` pour vérifier que votre modèle correspond au événements appropriés. Pour plus d'informations, consultez TestEventPattern.

Correspondance de valeurs d'événements

Dans un modèle d'événement, la valeur à mettre en correspondance se trouve dans un tableau JSON, entourée de crochets (« [», «] ») afin que vous puissiez fournir plusieurs valeurs. Par exemple, pour faire correspondre des événements provenant d'Amazon EC2 ou AWS Fargate, vous pouvez utiliser le modèle suivant, qui correspond aux événements dont la valeur du `"source"` champ est soit ou`"aws.ec2"`. `"aws.fargate"`

```
{
    "source": ["aws.ec2", "aws.fargate"]
}
```

Considérations lors de la création de modèles d'événements

Voici certains éléments à prendre en compte lors de la construction de vos modèles d'événements :

- EventBridge ignore les champs de l'événement qui ne sont pas inclus dans le modèle d'événement. En conséquence, le caractère générique `"*": "*"` est ajouté pour les champs qui n'apparaissent pas dans le modèle d'événements.
- Les valeurs auxquelles les modèles d'événements correspondent suivent les règles JSON. Vous pouvez inclure des chaînes entre guillemets ("), des nombres et les mots-clés `true`, `false`, et `null`.
- Pour les chaînes, EventBridge utilise une character-by-character correspondance exacte sans pliage des boîtiers ni aucune autre normalisation des chaînes.
- Pour les nombres, EventBridge utilise une représentation sous forme de chaîne. Par exemple, 300, 300,0 et 3.0e2 ne sont pas considérés égaux.
- Si plusieurs modèles sont spécifiés pour le même champ JSON, utilise EventBridge uniquement le dernier.

- Sachez que lorsque vous EventBridge compilez des modèles d'événements à utiliser, il utilise un point (.) comme caractère de jointure.

  Cela signifie que les modèles d'événements suivants EventBridge seront traités comme identiques :

```
## has no dots in keys
{ "detail" : { "state": { "status": [ "running" ] } } }

## has dots in keys
{ "detail" : { "state.status": [ "running" ] } }
```

  Et que les deux modèles d’événements correspondront aux deux événements suivants :

```
## has no dots in keys
{ "detail" : { "state": { "status": "running" } } }

## has dots in keys
{ "detail" : { "state.status": "running"  } }
```

  > **Note**
  >
  > Cela décrit EventBridge le comportement actuel et ne doit pas être invoqué pour ne pas le modifier.

- Les modèles d’événements contenant des champs en double ne sont pas valides. Si un modèle contient des champs dupliqués, EventBridge seule la valeur finale du champ est prise en compte.

  Par exemple, les modèles d’événements suivants correspondront au même événement :

```
## has duplicate keys
{
  "source": ["aws.s3"],
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail":  {
      "eventSource": ["s3.amazonaws.com"],
      "eventSource": ["sns.amazonaws.com"]
  }
}
```

```
## has unique keys
{
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": { "eventSource": ["sns.amazonaws.com"] }
}
```

Et EventBridge traite les deux événements suivants comme identiques :

```
## has duplicate keys
{
  "source": ["aws.s3"],
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail":  [
    {
      "eventSource": ["s3.amazonaws.com"],
      "eventSource": ["sns.amazonaws.com"]
    }
  ]
}

## has unique keys
{
  "source": ["aws.sns"],
  "detail-type": ["AWS API Call via CloudTrail"],
  "detail": [
    { "eventSource": ["sns.amazonaws.com"] }
  ]
}
```

> **Note**
>
> Cela décrit EventBridge le comportement actuel et ne doit pas être invoqué pour ne pas le modifier.

Opérations de comparaison à utiliser dans les modèles d'événements

Vous trouverez ci-dessous un résumé de tous les opérateurs de comparaison disponibles dans EventBridge.

Les opérateurs de comparaison ne fonctionnent que sur les nœuds terminaux, à l'exception de `$or` et `anything-but`.

Comparison (Comparaison)	Exemple	Syntaxe des règles
And	Le lieu est « New York » et le jour est « Monday »	`"Location": [ "New York" ], "Day": ["Monday"]`
N'importe quoi, sauf	L'état est n'importe quelle valeur autre que « initialisation ».	`"state": [ { "anything-but": "initializing" } ]`
Tout sauf (commence par)	La région n'est pas située aux États-Unis.	`"Region": [ { "anything-but": {"prefix": "us-" } } ]`
Tout sauf (se termine par)	FileName ne se termine pas par une extension .png.	`"FileName": [ { "anything-but": { "suffix": ".png" } } ]`
Tout sauf (ignorer le cas)	L'état est une valeur autre que « initialisation » ou toute autre variation du boîtier, telle que « INITIALISATION ».	`"state": : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}`
N'importe quoi, sauf en utilisant un joker	FileName n'est pas un chemin de fichier qui inclut `/lib/`.	`"FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]`
Commence par	La région se trouve aux États-Unis.	`"Region": [ {"prefix": "us-" } ]`

Comparison (Comparaison)	Exemple	Syntaxe des règles
Commence par (ignorer le cas)	Le nom du service commence par les lettres « eventb », quel que soit le cas.	`{"service" : [{ "prefix": { "equals-ignore-case": "eventb" }}]}`
Vide	LastName est vide.	`"LastName": [""]`
Égal à	Le nom est « Alice »	`"Name": [ "Alice" ]`
Est égal à (ignorer la casse)	Le nom est « Alice »	`"Name": [ { "equals-ignore-case": "alice" } ]`
Se termine par	FileName se termine par une extension .png	`"FileName": [ { "suffix": ".png" } ]`
Se termine par (ignorer les majuscules)	Le nom du service se termine par les lettres « tbridge » ou par toute autre variante du boîtier, telle que « TBRIDGE ».	`{"service" : [{ "suffix": { "equals-ignore-case": "tBridge" }}]}`
Existe	ProductName existe	`"ProductName": [ { "exists": true } ]`
N'existe pas	ProductName n'existe pas	`"ProductName": [ { "exists": false } ]`
Pas	La météo est tout sauf « Raining »	`"Weather": [ { "anything-but": [ "Raining" ] } ]`
Null	UserID est null	`"UserID": [ null ]`
Numérique (égal à)	Le prix est de 100	`"Price": [ { "numeric": [ "=", 100 ] } ]`
Numérique (plage)	Le prix est supérieur à 10 et inférieur ou égal à 20	`"Price": [ { "numeric": [ ">", 10, "<=", 20 ] } ]`

Comparison (Comparaison)	Exemple	Syntaxe des règles
Ou	PaymentType est « Crédit » ou « Débit »	`"PaymentType": [ "Credit", "Debit"]`
Ou (plusieurs champs)	Location est « New York », ou Day est « Monday ».	`"$or": [ { "Location": [ "New York" ] }, { "Day": [ "Monday" ] } ]`
Caractère générique	Tout fichier portant l'extension .png, situé dans le dossier « dir »	`"FileName": [ { "wildcard": "dir/*.png" } ]`

Exemples d'événements et de modèles d'événements

Vous pouvez utiliser tous les types de données et toutes les valeurs JSON pour mettre des événements en correspondance. Les exemples suivants illustrent des évènements et les modèles d'évènements qui leur correspondent.

Correspondance de champs

Vous pouvez mettre en correspondance la valeur d'un champ. Tenez compte de l'événement Amazon EC2 Auto Scaling 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 l’événement précédent, vous pouvez utiliser le champ `"responseElements"` pour établir une correspondance.

```
{
  "source": ["aws.autoscaling"],
  "detail-type": ["EC2 Instance Launch Successful"],
  "detail": {
   "responseElements": [null]
  }
}
```

Correspondance de valeurs

Prenons l’événement Amazon Macie suivant, qui est tronqué.

```
{
  "version": "0",
  "id": "0948ba87-d3b8-c6d4-f2da-732a1example",
  "detail-type": "Macie Finding",
  "source": "aws.macie",
  "account": "123456789012",
  "time": "2021-04-29T23:12:15Z",
  "region":"us-east-1",
  "resources": [

  ],
  "detail": {
    "schemaVersion": "1.0",
    "id": "64b917aa-3843-014c-91d8-937ffexample",
    "accountId": "123456789012",
    "partition": "aws",
    "region": "us-east-1",
    "type": "Policy:IAMUser/S3BucketEncryptionDisabled",
    "title": "Encryption is disabled for the S3 bucket",
    "description": "Encryption is disabled for the Amazon S3 bucket. The data in the
 bucket isn’t encrypted
        using server-side encryption.",
    "severity": {
        "score": 1,
        "description": "Low"
    },
    "createdAt": "2021-04-29T15:46:02Z",
    "updatedAt": "2021-04-29T23:12:15Z",
```

```
    "count": 2,
.
.
.
```

Le modèle d'événement suivant correspond à tout événement dont le score de gravité est de 1 et le nombre de 2.

```
{
  "source": ["aws.macie"],
  "detail-type": ["Macie Finding"],
  "detail": {
    "severity": {
      "score": [1]
    },
    "count":[2]
  }
}
```

Faire correspondre des valeurs nulles et des chaînes vides dans les modèles EventBridge d'événements Amazon

> **Important**
>
> Dans EventBridge, il est possible de créer des règles pouvant entraîner des higher-than-expected frais et des ralentissements. Par exemple, vous pouvez créer par inadvertance une règle qui entraîne une boucle infinie, dans laquelle une règle est déclenchée de manière récursive sans fin. Supposons que vous avez créé une règle permettant de détecter que les listes ACL ont été modifiées sur un compartiment Amazon S3 et de déclencher 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 obtenir des conseils sur la façon d'écrire des règles et des modèles d'événements précis afin de réduire au maximum ces résultats inattendus, consultez ??? et ???.

Vous pouvez créer un modèle d'événement qui correspond à un champ d'un événement comportant une valeur nulle ou une chaîne vide. Prenez l'exemple d'événement suivant :

Découvrez les bonnes pratiques pour éviter des frais plus élevés que prévu et les limitations.

```
{
  "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 d'événement suivant, qui correspond à l'événement précédent.

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

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

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

> **Note**
>
> Les valeurs nulles et les chaînes vides ne sont pas interchangeables dans une correspondance de modèle. Un modèle d'événement qui correspond à des chaînes vides ne correspond pas aux valeurs de `null`.

Tableaux dans les modèles d' EventBridge événements Amazon

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

> **Important**
>
> Dans EventBridge, il est possible de créer des règles pouvant entraîner des higher-than-expected frais et des ralentissements. Par exemple, vous pouvez créer par inadvertance une règle qui entraîne une boucle infinie, dans laquelle une règle est déclenchée de manière récursive sans fin. Supposons que vous avez créé une règle permettant de détecter que les listes ACL ont été modifiées sur un compartiment Amazon S3 et de déclencher 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 obtenir des conseils sur la façon d'écrire des règles et des modèles d'événements précis afin de réduire au maximum ces résultats inattendus, consultez ??? et ???.

Prenons l'exemple d'un modèle d'événement qui inclut le champ 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",
]
```

Le modèle d'événement précédent correspond à un événement qui inclut le champ suivant, car le premier élément du tableau du modèle d'événement 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"
]
```

Filtrage du contenu dans les modèles EventBridge d'événements Amazon

Amazon EventBridge prend en charge le filtrage déclaratif du contenu à l'aide de modèles d'événements. Grâce au filtrage de contenu, vous pouvez écrire des modèles d'événements complexes qui ne correspondent à des événements que sous des conditions très spécifiques. Par exemple, vous pouvez créer un modèle d'événement correspondant à un événement lorsque :

- Un champ de l'événement se situe dans une plage numérique spécifique.
- L'événement provient d'une adresse IP spécifique.
- Un champ spécifique n'existe pas dans le code JSON de l'événement.

> **Important**
>
> Dans EventBridge, il est possible de créer des règles pouvant entraîner des higher-than-expected frais et des ralentissements. Par exemple, vous pouvez créer par inadvertance une règle qui entraîne une boucle infinie, dans laquelle une règle est déclenchée de manière récursive sans fin. Supposons que vous avez créé une règle permettant de détecter que les listes ACL ont été modifiées sur un compartiment Amazon S3 et de déclencher 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 obtenir des conseils sur la façon d'écrire des règles et des modèles d'événements précis afin de réduire au maximum ces résultats inattendus, consultez ??? et ???.

Types de filtres

- Correspondance de préfixe
- Correspondance de suffixes
- Correspondance de type « anything-but » (tout-sauf)
- Correspondance numérique
- Correspondance d'adresses IP
- Correspondance exists
- quals-ignore-caseCorrespondance E

- Correspondance à l’aide de caractères génériques
- Exemple complexe avec correspondance multiple
- Exemple complexe avec correspondance $or

Correspondance de préfixe

Vous pouvez mettre en correspondance un événement en fonction du préfixe d’une valeur dans la source de l’événement. Vous pouvez utiliser la correspondance de préfixes pour les valeurs de chaîne.

Par exemple, le modèle d’événement suivant correspond à tout événement où le champ `"time"` commence par `"2017-10-02"`, tel que `"time": "2017-10-02T18:43:48Z"`.

```
{
  "time": [ { "prefix": "2017-10-02" } ]
}
```

Correspondance des préfixes sans tenir compte des majuscules

Vous pouvez également faire correspondre une valeur de préfixe quelle que soit la majuscule des caractères par lesquels une valeur commence, `equals-ignore-case` en conjonction avec `prefix`.

Par exemple, le modèle d'événement suivant correspond à tout événement où le `service` champ commence par la chaîne de caractères`EventB`, mais également `EVENTBeventb`, ou à toute autre majuscule de ces caractères.

```
{
  "detail": {"service" : [{ "prefix": { "equals-ignore-case": "EventB" }}]}
}
```

Correspondance de suffixes

Vous pouvez mettre en correspondance un événement en fonction du suffixe d’une valeur dans la source de l’événement. Vous pouvez utiliser la correspondance de suffixes pour les valeurs de chaîne.

Par exemple, le modèle d’événement suivant correspond à tout événement où le champ `"FileName"` se termine par l’extension de fichier `.png`.

```
{
  "FileName": [ { "suffix": ".png" } ]
}
```

Suffixe correspondant tout en ignorant les majuscules

Vous pouvez également faire correspondre la valeur d'un suffixe quelle que soit la majuscule des caractères par lesquels une valeur se termine, `equals-ignore-case` en utilisant conjointement avec `suffix`.

Par exemple, le modèle d'événement suivant correspondrait à tout événement où le `FileName` champ se terminait par la chaîne de caractères `.png`, mais également à `.PNG` toute autre mise en majuscules de ces caractères.

```
{
  "detail": {"FileName" : [{ "suffix": { "equals-ignore-case": ".png" }}]}
}
```

Correspondance de type « anything-but » (tout-sauf)

Tout sauf la correspondance correspond à tout sauf à ce qui est spécifié dans la règle.

Vous pouvez utiliser la correspondance anything-but avec des chaînes et des valeurs numériques, y compris des listes contenant uniquement des chaînes ou des nombres.

Le modèle d'événement suivant montre la correspondance anything-but avec des chaînes et des nombres.

```
{
  "detail": {
    "state": [ { "anything-but": "initializing" } ]
  }
}

{
  "detail": {
    "x-limit": [ { "anything-but": 123 } ]
  }
}
```

Le modèle d'événement suivant montre la correspondance anything-but avec une liste de chaînes.

```
{
  "detail": {
    "state": [ { "anything-but": [ "stopped", "overloaded" ] } ]
  }
}
```

Le modèle d'événement suivant montre la correspondance anything-but avec une liste de nombres.

```
{
  "detail": {
    "x-limit": [ { "anything-but": [ 100, 200, 300 ] } ]
  }
}
```

Tout sauf correspondre tout en ignorant le cas

Vous pouvez également l'utiliser `equals-ignore-case` en conjonction avec`anything-but`, pour faire correspondre les valeurs des chaînes indépendamment de la structure des caractères.

Le modèle d'événement suivant correspond aux `state` champs qui ne contiennent pas la chaîne « initialization », « INITIALIZING », « Initializing » ou toute autre majuscule de ces caractères.

```
{
  "detail": {"state" : [{ "anything-but": { "equals-ignore-case": "initializing" }}]}
}
```

Vous pouvez également utiliser `equals-ignore-case` en conjonction avec `anything-but` pour comparer à une liste de valeurs :

```
{
  "detail": {"state" : [{ "anything-but": { "equals-ignore-case": ["initializing",
 "stopped"] }}]}
}
```

Tout sauf la correspondance sur les préfixes

Vous pouvez l'utiliser `prefix` conjointement avec `anything-but` pour faire correspondre des valeurs de chaîne qui ne commencent pas par la valeur spécifiée. Cela inclut des valeurs uniques ou une liste de valeurs.

Le modèle d'événement suivant montre tout sauf une correspondance qui correspond à tout événement dont le champ ne contient pas le préfixe`"init"`. `"state"`

```
{
  "detail": {
    "state": [ { "anything-but": { "prefix": "init" } } ]
  }
}
```

Le modèle d'événement suivant montre tout sauf la correspondance utilisée avec une liste de valeurs de préfixes. Ce modèle d'événement correspond à tout événement qui ne possède ni le préfixe `"init"` ni `"stop"` le `"state"` champ.

```
{
"detail": {
  "state" : [{ "anything-but": { "prefix": ["init", "stop"] } } ] }
  }
}
```

Tout sauf la correspondance sur les suffixes

Vous pouvez l'utiliser `suffix` conjointement avec `anything-but` pour faire correspondre des valeurs de chaîne qui ne se terminent pas par la valeur spécifiée. Cela inclut des valeurs uniques ou une liste de valeurs.

Le modèle d'événement suivant correspond à toutes les valeurs du `FileName` champ qui ne se terminent pas par`.txt`.

```
{
  "detail": {
    "FileName": [ { "anything-but": { "suffix": ".txt" } } ]
  }
}
```

Le modèle d'événement suivant montre tout sauf la correspondance utilisée avec une liste de valeurs de suffixe. Ce modèle d'événement correspond à toutes les valeurs du `FileName` champ qui ne se terminent pas par `.txt` ou`.rtf`.

```
{
  "detail": {
```

```
    "FileName": [ { "anything-but": { "suffix": [".txt", ".rtf"] } } ]
  }
}
```

Tout sauf la correspondance à l'aide de caractères génériques

Vous pouvez utiliser le caractère générique (*) dans les valeurs que vous spécifiez pour tout sauf pour la correspondance. Cela inclut des valeurs uniques ou une liste de valeurs.

Le modèle d'événement suivant correspond à toutes les valeurs du `FileName` champ qui ne contiennent pas`/lib/`.

```
{
"detail": {
  "FilePath" : [{ "anything-but": { "wildcard": "*/lib/*" }}]
  }
}
```

Le modèle d'événement suivant montre tout sauf la correspondance utilisée avec une liste de valeurs, y compris des caractères génériques. Ce modèle d'événement correspond à toutes les valeurs du `FileName` champ qui ne contiennent aucun `/lib/` ou`/bin/`.

```
{
"detail": {
  "FilePath" : [{ "anything-but": { "wildcard": ["*/lib/*", "*/bin/*"] }}]
  }
}
```

Pour plus d’informations, consultez ???.

Correspondance numérique

La correspondance numérique fonctionne avec des valeurs qui sont des nombres JSON. Elle est limitée aux valeurs comprises entre -5.0e9 et +5.0e9 inclus, avec une précision à 15 chiffres (six chiffres à droite de la virgule décimale).

L’exemple suivant illustre la correspondance numérique pour un modèle d’événement qui ne correspond qu’aux événements dont tous les champs sont vrais.

```
{
```

```
  "detail": {
    "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ],
    "d-count": [ { "numeric": [ "<", 10 ] } ],
    "x-limit": [ { "numeric": [ "=", 3.018e2 ] } ]
  }
}
```

Correspondance d'adresses IP

Vous pouvez utiliser la correspondance d’adresses IP pour les adresses IPv4 et IPv6. Le modèle d’événement suivant illustre la correspondance d’adresses IP à des adresses IP qui commencent par 10.0.0 et se terminent par un nombre compris entre 0 et 255.

```
{
  "detail": {
    "sourceIPAddress": [ { "cidr": "10.0.0.0/24" } ]
  }
}
```

Correspondance exists

La correspondance exists fonctionne sur la présence ou l’absence d’un champ dans le code JSON de l’événement.

La correspondance de type « exists » ne fonctionne que sur des nœuds terminaux. Elle ne fonctionne pas sur des nœuds intermédiaires.

Le modèle d’événement suivant correspond à n’importe quel événement comportant un champ `detail.state`.

```
{
  "detail": {
    "state": [ { "exists": true  } ]
  }
}
```

Le modèle d’événement précédent correspond à l’événement suivant.

```
{
  "version": "0",
  "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-abcd1111"],
  "detail": {
    "instance-id": "i-abcd1111",
    "state": "pending"
  }
}
```

Le modèle d'événement précédent ne correspond PAS à l'événement suivant, car il ne comporte aucun champ `detail.state`.

```
{
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1:123456789012:instance/i-02ebd4584a2ebd341" ],
  "detail": {
    "c-count" : {
       "c1" : 100
    }
  }
}
```

quals-ignore-caseCorrespondance E

La quals-ignore-case correspondance E fonctionne sur les valeurs de chaîne, quel que soit le cas.

Le modèle d'événement suivant correspond à n'importe quel événement comportant un champ `detail-type` qui correspond à la chaîne spécifiée, quelle que soit sa casse.

```
{
  "detail-type": [ { "equals-ignore-case": "ec2 instance state-change notification" } ]
}
```

Le modèle d'événement précédent correspond à l'événement suivant.

```
{
  "detail-type": [ "EC2 Instance State-change Notification" ],
  "resources": [ "arn:aws:ec2:us-east-1:123456789012:instance/i-02ebd4584a2ebd341" ],
```

```
  "detail": {
    "c-count" : {
       "c1" : 100
    }
  }
}
```

Correspondance à l’aide de caractères génériques

Vous pouvez utiliser le caractère générique (*) pour mettre en correspondance des valeurs de chaîne dans les modèles d’événements.

> **Note**
>
> Actuellement, le caractère générique n’est pris en charge que dans les règles de bus d’événements.

Considérations relatives à l’utilisation de caractères génériques dans vos modèles d’événements :

- Vous pouvez spécifier n’importe quel nombre de caractères génériques dans une valeur de chaîne donnée ; toutefois, les caractères génériques consécutifs ne sont pas pris en charge.
- EventBridge prend en charge l'utilisation de la barre oblique inverse (\) pour spécifier les caractères littéraux * et \ dans les filtres génériques :
  - La chaîne * représente le caractère littéral *
  - La chaîne \\ représente le caractère littéral \

  L’utilisation de la barre oblique inversée pour mettre en échappement d’autres caractères n’est pas prise en charge.

Complexité des caractères génériques et des modèles d’événements

La complexité d’une règle utilisant des caractères génériques est limitée. Si une règle est trop complexe, EventBridge renvoie un `InvalidEventPatternException` lorsque vous essayez de créer la règle. Si votre règle génère une telle erreur, envisagez de suivre les conseils ci-dessous pour réduire la complexité du modèle d’événement :

- Réduisez le nombre de caractères génériques utilisés

N’utilisez des caractères génériques que lorsque vous devez véritablement comparer plusieurs valeurs possibles. Prenons l’exemple de modèle d’événement suivant, où vous souhaitez comparer des bus d’événements dans la même région :

```
{
"EventBusArn": [ { "wildcard": "*:*:*:*:*:event-bus/*" } ]
}
```

Dans le cas ci-dessus, de nombreuses sections de l’ARN seront directement basées sur la région dans laquelle résident vos bus d’événements. Ainsi, si vous utilisez la région `us-east-1`, voici exemple de modèle moins complexe qui correspond toujours aux valeurs souhaitées :

```
{
"EventBusArn": [ { "wildcard": "arn:aws:events:us-east-1:*:event-bus/*" } ]
}
```

- Réduisez les séquences de caractères qui se répètent, après un caractère générique

  Le fait que la même séquence de caractères apparaisse plusieurs fois après l’utilisation d’un caractère générique augmente la complexité du traitement du modèle d’événement. Recréez votre modèle d’événement afin de réduire au maximum les séquences répétées. Prenons l’exemple suivant, qui met en correspondance le nom de fichier `doc.txt` pour n’importe quel utilisateur :

  ```
  {
  "FileName": [ { "wildcard": "/Users/*/dir/dir/dir/dir/dir/doc.txt" } ]
  }
  ```

  Si vous saviez que le fichier `doc.txt` n’apparaîtrait que dans le chemin spécifié, vous pourriez réduire la séquence de caractères répétée comme suit :

  ```
  {
  "FileName": [ { "wildcard": "/Users/*/doc.txt" } ]
  }
  ```

Exemple complexe avec correspondance multiple

Vous pouvez combiner plusieurs règles de correspondance dans un modèle d'événement plus complexe. Par exemple, le modèle d'événement suivant combine `anything-but` et `numeric`.

```
{
  "time": [ { "prefix": "2017-10-02" } ],
  "detail": {
    "state": [ { "anything-but": "initializing" } ],
    "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ],
    "d-count": [ { "numeric": [ "<", 10 ] } ],
    "x-limit": [ { "anything-but": [ 100, 200, 300 ] } ]
  }
}
```

Note

Lorsque vous créez des modèles d'événements, si vous incluez une clé plusieurs fois, la dernière référence sera celle utilisée pour évaluer les événements. Par exemple, pour le modèle suivant :

```
{
  "detail": {
    "location": [ { "prefix": "us-" } ],
    "location": [ { "anything-but": "us-east" } ]
  }
}
```

Seul `{ "anything-but": "us-east" }` sera pris en compte lors de l'évaluation de `location`.

Exemple complexe avec correspondance **`$or`**

Vous pouvez également créer des modèles d'événements complexes qui vérifient si l'une des valeurs de champs correspond, dans plusieurs champs. Utilisez `$or` pour créer un modèle d'événement qui correspond si l'une des valeurs de plusieurs champs correspond.

Notez que vous pouvez inclure d'autres types de filtres, tels que la correspondance numérique et les tableaux, dans votre correspondance de modèles pour les champs individuels de votre construction `$or`.

Le modèle d'événement suivant correspond si l'une des conditions suivantes est remplie :

- Le champ `c-count` est supérieur à 0 ou inférieur ou égal à 5.
- Le champ `d-count` est inférieur à 10.
- Le champ `x-limit` est égal à 3.018e2.

```
{
  "detail": {
    "$or": [
      { "c-count": [ { "numeric": [ ">", 0, "<=", 5 ] } ] },
      { "d-count": [ { "numeric": [ "<", 10 ] } ] },
      { "x-limit": [ { "numeric": [ "=", 3.018e2 ] } ] }
    ]
  }
}
```

> **Note**
>
> Les API qui acceptent un modèle d'événement (telles que `PutRule`, `CreateArchive`, `UpdateArchive` et `TestEventPattern`) lanceront `InvalidEventPatternException` si l'utilisation de `$or` génère plus de 1 000 combinaisons de règles.
> Pour déterminer le nombre de combinaisons de règles dans un modèle d'événement, multipliez le nombre total d'arguments de chaque tableau `$or` dans le modèle d'événement. Par exemple, le modèle ci-dessus contient un seul tableau `$or` avec trois arguments, de sorte que le nombre total de combinaisons de règles est également de trois. Si vous ajoutiez un autre tableau `$or` avec deux arguments, le nombre total de combinaisons de règles serait alors de six.

Tester un modèle d'événement à l'aide du EventBridge Sandbox

Les règles utilisent des modèles d'événements pour sélectionner des événements et les envoyer vers des cibles. Les modèles d'événement ont la même structure que les événements auxquels ils correspondent. Soit un modèle d'événement correspond à un événement, soit il n'y correspond pas.

La définition d’un modèle d’événement fait généralement partie du processus plus large de création d’une nouvelle règle ou de modification d’une règle existante. En utilisant le Sandbox EventBridge, vous pouvez toutefois définir rapidement un modèle d'événement et utiliser un exemple d'événement pour confirmer que le modèle correspond aux événements souhaités, sans avoir à créer ou à modifier de règle. Une fois que vous avez testé votre modèle d'événement, EventBridge offrez-vous la possibilité de créer une nouvelle règle en utilisant ce modèle d'événement directement depuis le sandbox.

Pour plus d’informations sur les modèles d’événements, consultez ???.

> **Important**
>
> Dans EventBridge, il est possible de créer des règles pouvant entraîner des higher-than-expected frais et des ralentissements. Par exemple, vous pouvez créer par inadvertance une règle qui entraîne une boucle infinie, dans laquelle une règle est déclenchée de manière récursive sans fin. Supposons que vous avez créé une règle permettant de détecter que les listes ACL ont été modifiées sur un compartiment Amazon S3 et de déclencher 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 obtenir des conseils sur la façon d’écrire des règles et des modèles d’événements précis afin de réduire au maximum ces résultats inattendus, consultez ??? et ???.

Pour tester un modèle d'événement à l'aide du EventBridge sandbox

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Ressources pour développeurs, puis sélectionnez Environnement de test (sandbox), et sur la page Environnement de test (sandbox), choisissez l'onglet Modèle d'événement.
3. Dans Source de l'événement, choisissez AWS des événements ou des événements EventBridge partenaires.
4. Dans la section Exemples d’événements, choisissez un Type d’exemple d’événement par rapport auquel vous souhaitez tester votre modèle d’événement.

   Les types d’exemples d’événements suivants sont disponibles :

   - AWS événements — Choisissez parmi les événements émis ou pris en charge Services AWS.

- EventBridge événements partenaires : sélectionnez parmi les événements émis par des services tiers qui prennent en charge EventBridge, tels que Salesforce.
- Saisir mon propre : entrez votre propre événement au format texte JSON.

  Vous pouvez également utiliser un événement AWS ou un événement partenaire comme point de départ pour créer votre propre événement personnalisé.

  1. Sélectionnez AWS des événements ou des événements EventBridge partenaires.
  2. Utilisez le menu déroulant Exemples d’événements pour sélectionner l’événement que vous souhaitez utiliser comme point de départ pour votre événement personnalisé.

     EventBridge affiche l'exemple d'événement.
  3. Sélectionnez Copier.
  4. Sélectionnez Saisir mon propre pour Type d’événement.
  5. Supprimez l'exemple de structure d'événement dans le volet d'édition JSON et collez l'événement AWS ou l'événement partenaire à sa place.
  6. Modifiez le code JSON de l’événement pour créer votre propre exemple d’événement.

5. Choisissez une Méthode de création. Vous pouvez créer un modèle d'événement à partir d'un EventBridge schéma ou d'un modèle, ou vous pouvez créer un modèle d'événement personnalisé.

   Existing schema

   Pour utiliser un EventBridge schéma existant afin de créer le modèle d'événement, procédez comme suit :

   1. Dans la section Méthode de création, pour Méthode, sélectionnez Utiliser le schéma.
   2. Dans la section Modèle d’événement, pour Type de schéma, sélectionnez Sélectionner un schéma dans le registre des schémas.
   3. Pour le Registre des schémas, cliquez sur la liste déroulante et entrez le nom d’un registre de schémas, tel que `aws.events`. Vous pouvez également sélectionner une option dans la liste déroulante qui s’affiche.
   4. Pour Schéma, cliquez sur la liste déroulante et entrez le nom du schéma à utiliser. Par exemple, `aws.s3@0bjectDeleted`. Vous pouvez également sélectionner une option dans la liste déroulante qui s’affiche.

5. Dans la section Modèles, cliquez sur le bouton Modifier en regard de n'importe quel attribut pour ouvrir ses propriétés. Définissez les champs Relation et Valeur selon vos besoins, puis choisissez Jeu pour enregistrer l'attribut.

   > **Note**
   >
   > Pour en savoir plus sur la définition d'un attribut, cliquez sur l'icône Infos en regard du nom de l'attribut. Pour savoir comment définir les propriétés des attributs dans votre événement, ouvrez la section Remarque de la boîte de dialogue des propriétés des attributs.
   > Pour supprimer les propriétés d'un attribut, cliquez sur le bouton Modifier correspondant à cet attribut, puis choisissez Effacer.

6. Choisissez Générer un modèle d'événement au format JSON pour générer et valider votre modèle d'événement sous forme de texte JSON.
7. Pour tester l'exemple d'événement par rapport à votre modèle de test, choisissez Modèle de test.

   EventBridge affiche une boîte de message indiquant si votre exemple d'événement correspond au modèle d'événement.

   Vous pouvez également choisir l'une des options suivantes :

   - Copier : copiez le modèle d'événement dans le presse-papiers de votre appareil.
   - Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.

Custom schema

Pour écrire un schéma personnalisé et le convertir en modèle d'événement, procédez comme suit :

1. Dans la section Méthode de création, pour Méthode, choisissez Utiliser le schéma.
2. Dans la section Modèle d'événement, pour Type de schéma, choisissez Saisir le schéma.
3. Entrez votre schéma dans la zone de texte. Vous devez mettre en forme le schéma au format texte JSON valide.

4. Dans la section Modèles, cliquez sur le bouton Modifier en regard de n'importe quel attribut pour ouvrir ses propriétés. Définissez les champs Relation et Valeur selon vos besoins, puis choisissez Jeu pour enregistrer l'attribut.

   > **Note**
   >
   > Pour en savoir plus sur la définition d'un attribut, cliquez sur l'icône Infos en regard du nom de l'attribut. Pour savoir comment définir les propriétés des attributs dans votre événement, ouvrez la section Remarque de la boîte de dialogue des propriétés des attributs.
   > Pour supprimer les propriétés d'un attribut, cliquez sur le bouton Modifier correspondant à cet attribut, puis choisissez Effacer.

5. Choisissez Générer un modèle d'événement au format JSON pour générer et valider votre modèle d'événement sous forme de texte JSON.
6. Pour tester l'exemple d'événement par rapport à votre modèle de test, choisissez Modèle de test.

   EventBridge affiche une boîte de message indiquant si votre exemple d'événement correspond au modèle d'événement.

   Vous pouvez également choisir l'une des options suivantes :

   - Copier : copiez le modèle d'événement dans le presse-papiers de votre appareil.
   - Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.

Event pattern

Pour écrire un modèle d'événement personnalisé au format JSON, procédez comme suit :

1. Dans la section Méthode de création, pour Méthode, choisissez Modèle personnalisé (éditeur JSON).
2. Pour Modèle d'événement, entrez votre modèle d'événement personnalisé au format texte JSON.
3. Pour tester l'exemple d'événement par rapport à votre modèle de test, choisissez Modèle de test.

EventBridge affiche une boîte de message indiquant si votre exemple d'événement correspond au modèle d'événement.

Vous pouvez également choisir l’une des options suivantes :

- Copier : copiez le modèle d’événement dans le presse-papiers de votre appareil.
- Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.
- Formulaire de modèle d’événement : ouvre le modèle d’événement dans le Générateur de modèle. Si le modèle ne peut pas être affiché tel quel dans Pattern Builder, vous EventBridge avertit avant qu'il n'ouvre Pattern Builder.

6. (Facultatif) Pour créer une règle avec ce modèle d’événement et l’affecter à un bus d’événements spécifique, choisissez Créer une règle avec un modèle.

   EventBridge vous amène à l'étape 1 de la section Créer une règle, que vous pouvez utiliser pour créer une règle et l'attribuer au bus d'événements de votre choix.

   Notez que l’Étape 2 : Générer un modèle d’événement contient les informations du modèle d’événement que vous avez déjà spécifiées et que vous pouvez accepter ou mettre à jour.

   Pour plus d’informations sur la création de règles, consultez ???.

Bonnes pratiques lors de la définition des modèles EventBridge d'événements Amazon

Vous trouverez ci-dessous quelques bonnes pratiques à prendre en compte lors de la définition de modèles d’événements dans les règles de votre bus d’événements.

Évitez d’écrire des boucles infinies

Dans EventBridge, il est possible de créer des règles qui mènent à des boucles infinies, où une règle est déclenchée à plusieurs reprises. 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 ces problèmes, écrivez les modèles d’événements de vos règles aussi précisément que possible, afin qu’ils correspondent uniquement aux événements que vous souhaitez réellement

envoyer à la cible. Dans l'exemple ci-dessus, vous créez un modèle d'événement pour mettre en correspondance des événements afin que les actions déclenchées ne déclenchent pas à nouveau la même règle. Par exemple, créez un modèle d'événement dans votre règle qui met en correspondance des événements uniquement s'il s'avère que les listes ACL sont dans un état incorrect, plutôt qu'après une modification. Pour plus d'informations, consultez ??? et ???.

Une boucle infinie peut rapidement entraîner des coûts plus importants que prévu. Elle peut également entraîner des limitations et un retard dans la livraison des événements. Vous pouvez surveiller la limite supérieure de vos taux d'invocation pour être averti de pics de volume inattendus.

Utilisez les budgets pour vous avertir lorsque les frais dépassent votre limite spécifiée. Pour plus d'informations, consultez Gestion des coûts avec les budgets.

Faites en sorte que les modèles d'événements soient aussi précis que possible

Plus votre modèle d'événement est précis, plus il est probable qu'il corresponde uniquement aux événements auxquels vous souhaitez réellement qu'il corresponde. De plus, vous évitez les correspondances inattendues lorsque de nouveaux événements sont ajoutés à une source d'événement ou que des événements existants sont mis à jour pour inclure de nouvelles propriétés.

Les modèles d'événements peuvent inclure des filtres qui mettent en correspondance les éléments suivants :

- Métadonnées de l'événement relatives à l'événement, telles que `source`, `detail-type`, `account` ou `region`.
- Données de l'événement, c'est-à-dire les champs à l'intérieur de l'objet `detail`.
- Contenu de l'événement ou valeurs réelles des champs à l'intérieur de l'objet `detail`.

La plupart des modèles sont simples, tels que la spécification des filtres `source` et `detail-type` uniquement. Cependant, EventBridge les modèles incluent la possibilité de filtrer sur n'importe quelle clé ou valeur de l'événement. En outre, vous pouvez appliquer des filtres de contenu tels que `prefix` et `suffix` pour améliorer la précision de vos modèles. Pour plus d'informations, consultez ???.

Spécifiez la source de l’événement et le type de détail sous forme de filtres

Vous pouvez réduire la génération de boucles infinies et la mise en correspondance d’événements indésirables en rendant vos modèles d’événements plus précis à l’aide des champs de métadonnées `source` et `detail-type`.

Lorsque vous devez mettre en correspondance des valeurs spécifiques dans deux champs ou plus, utilisez l’opérateur de comparaison `$or`, plutôt que de répertorier toutes les valeurs possibles dans un seul tableau de valeurs.

Pour les événements diffusés via AWS CloudTrail, nous vous recommandons d'utiliser le `eventName` champ comme filtre.

L'exemple de modèle d'événement suivant correspond `CreateQueue` ou `SetQueueAttributes` provient du service Amazon Simple Queue Service, `CreateKey` ou à `DisableKeyRotation` des événements du AWS Key Management Service service.

```
{
  "detail-type": ["AWS API Call via CloudTrail"],
  "$or": [{
      "source": [
        "aws.sqs"
        ],
      "detail": {
        "eventName": [
          "CreateQueue",
          "SetQueueAttributes"
        ]
      }
    },
    {
      "source": [
        "aws.kms"
        ],
      "detail": {
        "eventName": [
          "CreateKey",
          "DisableKeyRotation"
        ]
      }
    }
  ]
```

```
}
```

Spécifiez le compte et la région sous forme de filtres

L'inclusion des champs `account` et `region` dans votre modèle d'événement permet de limiter la mise en correspondance des événements entre comptes ou entre régions.

Spécifiez des filtres de contenu

Le filtrage basé sur le contenu peut contribuer à améliorer la précision des modèles d'événements, tout en maintenant leur longueur au minimum. Par exemple, la mise en correspondance basée sur une plage numérique peut être utile au lieu de répertorier toutes les valeurs numériques possibles.

Pour plus d'informations, consultez ???.

Définissez la portée de vos modèles d'événements pour prendre en compte les mises à jour de la source d'événement

Lorsque vous créez des modèles d'événements, vous devez tenir compte du fait que les schémas d'événements et les domaines d'événements peuvent évoluer et s'étendre au fil du temps. Là encore, le fait de rendre vos modèles d'événements aussi précis que possible contribue à limiter les correspondances inattendues en cas de modification ou d'extension de la source d'événement.

Supposons, par exemple, que vous compariez des événements provenant d'un nouveau microservice qui publie des événements liés au paiement. Dans un premier temps, le service utilise le domaine `acme.payments` et publie un seul événement, `Payment accepted` :

```
{
  "detail-type": "Payment accepted",
  "source": "acme.payments",
  "detail": {
    "type": "credit",
    "amount": "100",
    "date": "2023-06-10",
    "currency": "USD"
    }
  }
}
```

À ce stade, vous pouvez créer un modèle d'événement simple qui correspond aux événements Payment accepted :

```
{ “source” : “acme.payments” }
```

Supposons toutefois que le service introduise ultérieurement un nouvel événement pour les paiements rejetés :

```
{
  "detail-type": "Payment rejected",
  "source": "acme.payments",
  "detail": {
  }
}
```

Dans ce cas, le modèle d'événement simple que vous avez créé correspondra désormais à la fois aux événements `Payment accepted` et `Payment rejected`. EventBridge achemine les deux types d'événements vers la cible spécifiée pour traitement, ce qui peut entraîner des échecs de traitement et des coûts de traitement supplémentaires.

Pour définir la portée de votre modèle d'événements sur les événements `Payment accepted` uniquement, vous devez spécifier au moins `source` et `detail-type` :

```
{
  "detail-type": "Payment accepted",
  "source": "acme.payments"
  }
}
```

Vous pouvez également spécifier le compte et la région dans votre modèle d'événement, afin de limiter davantage les cas où les événements entre comptes ou entre régions correspondent à cette règle.

```
{
  "account": "012345678910",
  "source": "acme.payments",
  "region": "AWS-Region",
  "detail-type": "Payment accepted"
}
```

Validez les modèles d’événements

Pour garantir que les règles correspondent aux événements souhaités, nous vous recommandons vivement de valider vos modèles d’événements. Vous pouvez valider vos modèles d'événements à l'aide de la EventBridge console ou de l'API :

- Dans la EventBridge console, vous pouvez créer et tester des modèles d'événements dans le cadre de la création d'une règle, ou séparément à l'aide de la Sandbox.
- Vous pouvez tester vos modèles d'événements par programmation à l'aide de cette action. TestEventPattern

EventBridge Règles d'Amazon

Vous spécifiez ce qu' EventBridge il est fait des événements transmis à chaque bus d'événements. Pour ce faire, vous devez créer des règles. Une règle indique les événements à envoyer et à quelles cibles pour les traiter. Une seule règle peut envoyer un événement à plusieurs cibles, qui s'exécutent ensuite en parallèle.

Vous pouvez créer deux types de règle :

- Règles identiques en ce qui concerne les données des événements

  Vous pouvez créer des règles qui correspondent aux événements entrants en fonction de critères de données d'événements (appelés modèles d'événements). Un modèle d'événement définit la structure de l'événement et les champs auxquels une règle correspond. Si un événement correspond aux critères définis dans le modèle d'événement, il est EventBridge envoyé aux cibles que vous spécifiez.

  Pour de plus amples informations, veuillez consulter ???.
- Règles exécutées selon un calendrier

  Vous pouvez également créer des règles qui envoient des événements aux cibles spécifiées à des intervalles spécifiés. Par exemple, pour exécuter régulièrement une Lambda fonction, vous pouvez créer une règle à exécuter selon un calendrier.

  > **Note**
  >
  > EventBridge propose Amazon EventBridge Scheduler, un planificateur sans serveur qui vous permet de créer, d'exécuter et de gérer des tâches à partir d'un service géré centralisé. EventBridge Le planificateur est hautement personnalisable et offre une évolutivité améliorée par rapport aux règles EventBridge planifiées, avec un ensemble plus large d'opérations et de services d'API cibles. AWS
  > Nous vous recommandons d'utiliser le EventBridge planificateur pour appeler des cibles selon un calendrier. Pour de plus amples informations, veuillez consulter ???.

La vidéo suivante présente les principes de base des règles : Que sont les règles ?

Règles EventBridge gérées par Amazon

Outre les règles que vous créez, les AWS services peuvent créer et gérer des EventBridge règles dans votre AWS compte qui sont nécessaires à certaines fonctions de ces services. Celles-ci sont appelées des règles gérées.

Lorsqu'un service crée une règle gérée, il peut également créer une IAM politique qui autorise ce service à 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. Toutefois, avant de les supprimer, vérifiez 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.

Création de règles Amazon EventBridge qui réagissent aux événements

Pour effectuer une action lorsque des événements sont reçus par Amazon EventBridge, vous pouvez créer des règles. Lorsqu'un événement correspond au modèle d'événement défini dans votre règle, EventBridge envoie l'événement à la cible spécifiée et déclenche l'action définie dans la règle.

La vidéo suivante explique comment créer différents types de règles et comment les tester : En savoir plus sur les règles

Procédez comme suit pour créer une règle Amazon EventBridge qui répond aux événements.

Création d'une règle qui réagit aux événements

Les étapes suivantes expliquent comment créer une règle utilisée par EventBridge pour faire correspondre les événements lorsqu'ils sont envoyés au bus d'événements spécifié.

Étapes

- Définition de la règle
- Création du modèle d'événement
- Sélection des cibles
- Configuration des balises et vérification de la règle

Définition de la règle

Commencez par entrer un nom et une description pour identifier la règle. Vous devez également définir le bus d'événements dans lequel votre règle recherche les événements correspondant à un modèle d'événement.

Pour définir les détails de la règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule (Créer une règle).
4. Entrez un Nom et éventuellement une Description pour la règle.

Une règle ne peut pas avoir le même nom qu'une autre règle de la même Région AWS et sur le même bus d'événement.

5. Pour Bus d'événements, choisissez le bus d'événements à associer à cette règle. Si vous souhaitez que cette règle corresponde aux événements provenant de votre compte, sélectionnez Bus d'événements par défaut AWS. Lorsqu'un Service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.
6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).

Création du modèle d'événement

Créez ensuite le modèle d'événement. Pour ce faire, spécifiez la source de l'événement, choisissez la base du modèle d'événement et définissez les attributs et les valeurs sur lesquels vous souhaitez établir une correspondance. Vous pouvez également générer le modèle d'événement au format JSON et le tester par rapport à un exemple d'événement.

Pour créer le modèle d'événement

1. Pour Event source (Origine de l'événement), choisissez events or EventBridge partner events (Événements AWS ou événements partenaires EventBridge).
2. (Facultatif) Dans la section Exemples d'événements, choisissez un Type d'exemple d'événement par rapport auquel vous souhaitez tester votre modèle d'événement.

   Les types d'exemples d'événements suivants sont disponibles :

   - Événements AWS : faites votre choix parmi les événements émis par les Services AWS pris en charge.
   - Événements partenaires EventBridge : faites votre choix parmi les événements émis par des services tiers prenant en charge EventBridge, tels que Salesforce.
   - Saisir mon propre : entrez votre propre événement au format texte JSON.

     Vous pouvez également utiliser un événement AWS ou partenaire comme point de départ pour créer votre propre événement personnalisé.

     1. Sélectionnez Événements AWS ou Événements partenaires EventBridge.

2. Utilisez le menu déroulant Exemples d'événements pour sélectionner l'événement que vous souhaitez utiliser comme point de départ pour votre événement personnalisé.

   EventBridge affiche l'exemple d'événement.

3. Sélectionnez Copier.
4. Sélectionnez Saisir mon propre pour Type d'événement.
5. Supprimez la structure de l'exemple d'événement dans le volet d'édition JSON et collez l'événement AWS ou partenaire à sa place.
6. Modifiez le code JSON de l'événement pour créer votre propre exemple d'événement.

3. Choisissez une Méthode de création. Vous pouvez créer un modèle d'événement à partir d'un schéma ou d'un modèle EventBridge, ou créer un modèle d'événement personnalisé.

   Existing schema

   Pour utiliser un schéma EventBridge existant pour créer le modèle d'événement, procédez comme suit :

   1. Dans la section Méthode de création, pour Méthode, sélectionnez Utiliser le schéma.
   2. Dans la section Modèle d'événement, pour Type de schéma, sélectionnez Sélectionner un schéma dans le registre des schémas.
   3. Pour le Registre des schémas, cliquez sur la liste déroulante et entrez le nom d'un registre de schémas, tel que `aws.events`. Vous pouvez également sélectionner une option dans la liste déroulante qui s'affiche.
   4. Pour Schéma, cliquez sur la liste déroulante et entrez le nom du schéma à utiliser. Par exemple, `aws.s3@0bjectDeleted`. Vous pouvez également sélectionner une option dans la liste déroulante qui s'affiche.
   5. Dans la section Modèles, cliquez sur le bouton Modifier en regard de n'importe quel attribut pour ouvrir ses propriétés. Définissez les champs Relation et Valeur selon vos besoins, puis choisissez Jeu pour enregistrer l'attribut.

      > **Note**
      >
      > Pour en savoir plus sur la définition d'un attribut, cliquez sur l'icône Infos en regard du nom de l'attribut. Pour savoir comment définir les propriétés des attributs dans votre événement, ouvrez la section Remarque de la boîte de dialogue des propriétés des attributs.

Pour supprimer les propriétés d'un attribut, cliquez sur le bouton Modifier correspondant à cet attribut, puis choisissez Effacer.

6. Choisissez Générer un modèle d'événement au format JSON pour générer et valider votre modèle d'événement sous forme de texte JSON.
7. (Facultatif) Pour tester l'exemple d'événement par rapport à votre modèle de test, choisissez Modèle de test.

   EventBridge affiche un message indiquant si votre exemple d'événement correspond au modèle d'événement.

   Vous pouvez également choisir l'une des options suivantes :

   - Copier : copiez le modèle d'événement dans le presse-papiers de votre appareil.
   - Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.

Custom schema

Pour écrire un schéma personnalisé et le convertir en modèle d'événement, procédez comme suit :

1. Dans la section Méthode de création, pour Méthode, choisissez Utiliser le schéma.
2. Dans la section Modèle d'événement, pour Type de schéma, choisissez Saisir le schéma.
3. Entrez votre schéma dans la zone de texte. Vous devez mettre en forme le schéma au format texte JSON valide.
4. Dans la section Modèles, cliquez sur le bouton Modifier en regard de n'importe quel attribut pour ouvrir ses propriétés. Définissez les champs Relation et Valeur selon vos besoins, puis choisissez Jeu pour enregistrer l'attribut.

> **Note**
>
> Pour en savoir plus sur la définition d'un attribut, cliquez sur l'icône Infos en regard du nom de l'attribut. Pour savoir comment définir les propriétés des attributs dans votre événement, ouvrez la section Remarque de la boîte de dialogue des propriétés des attributs.

Pour supprimer les propriétés d'un attribut, cliquez sur le bouton Modifier correspondant à cet attribut, puis choisissez Effacer.

5. Choisissez Générer un modèle d'événement au format JSON pour générer et valider votre modèle d'événement sous forme de texte JSON.
6. (Facultatif) Pour tester l'exemple d'événement par rapport à votre modèle de test, choisissez Modèle de test.

   EventBridge affiche un message indiquant si votre exemple d'événement correspond au modèle d'événement.

   Vous pouvez également choisir l'une des options suivantes :

   - Copier : copiez le modèle d'événement dans le presse-papiers de votre appareil.
   - Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.

Event pattern

Pour écrire un modèle d'événement personnalisé au format JSON, procédez comme suit :

1. Dans la section Méthode de création, pour Méthode, choisissez Modèle personnalisé (éditeur JSON).
2. Pour Modèle d'événement, entrez votre modèle d'événement personnalisé au format texte JSON.
3. (Facultatif) Pour tester l'exemple d'événement par rapport à votre modèle de test, choisissez Modèle de test.

   EventBridge affiche un message indiquant si votre exemple d'événement correspond au modèle d'événement.

   Vous pouvez également choisir l'une des options suivantes :

   - Copier : copiez le modèle d'événement dans le presse-papiers de votre appareil.
   - Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.
   - Formulaire de modèle d'événement : ouvre le modèle d'événement dans le Générateur de modèle. Si le modèle ne peut pas être rendu tel quel dans le Générateur de modèle, EventBridge vous avertit avant d'ouvrir le Générateur de modèle.

4. Choisissez Next (Suivant).

Sélection des cibles

Choisissez une ou plusieurs cibles pour recevoir des événements correspondant au modèle spécifié. Les cibles peuvent inclure un bus d'événements EventBridge, des destinations d'API EventBridge, y compris des partenaires SaaS tels que Salesforce, ou un autre Service AWS.

Pour sélectionner des cibles

1. Pour Type de cible, choisissez l'un des types de cibles suivants :

   Event bus

   Pour sélectionner un bus d'événements EventBridge, sélectionnez Bus d'événements EventBridge, puis procédez comme suit :

   - Pour utiliser un bus d'événements dans la même Région AWS que cette règle :
     1. Sélectionnez Bus d'événements dans le même compte et la même région.
     2. Pour Bus d'événements pour la cible, cliquez sur la liste déroulante et entrez le nom du bus d'événements. Vous pouvez également sélectionner le bus d'événements dans la liste déroulante.

        Pour de plus amples informations, veuillez consulter ???.
   - Pour utiliser un bus d'événements dans une autre Région AWS ou un autre compte que cette règle :
     1. Sélectionnez Bus d'événements dans un compte ou une région différent.
     2. Pour Bus d'événements comme cible, entrez l'ARN du bus d'événements que vous souhaitez utiliser.

        Pour plus d'informations, consultez :

        - ???
        - ???

API destination

Pour utiliser une destination d'API EventBridge, sélectionnez Destination d'API EventBridge, puis effectuez l'une des opérations suivantes :

- Pour utiliser une destination d'API existante, sélectionnez Utiliser une destination d'API existante. Ensuite, sélectionnez une destination d'API dans la liste déroulante.
- Pour créer une nouvelle destination d'API, sélectionnez Créer une nouvelle destination API. Fournissez ensuite les informations suivantes pour la destination :
  - Nom : entrez un nom pour la destination.

    Les noms doivent être uniques dans votre Compte AWS. Les noms peuvent comporter jusqu'à 64 caractères. Les caractères valides sont A-Z, a-z, 0-9 et . _ - (tiret).
  - (Facultatif) Description : entrez une description pour la destination.

    Les descriptions peuvent comporter jusqu'à 512 caractères.
  - Point de terminaison de la destination d'API : point de terminaison d'URL de la cible.

    L'URL du point de terminaison doit commencer par **`https`**. Vous pouvez inclure le caractère générique **`*`** en tant que paramètre de chemin. Vous pouvez définir les paramètres du chemin à partir de l'attribut `HttpParameters` de la cible.
  - Méthode HTTP : sélectionnez la méthode HTTP utilisée lorsque vous invoquez le point de terminaison.
  - (Facultatif) Limite du taux d'appel par seconde : entrez le nombre maximal d'invocations acceptées par seconde pour cette destination.

    Cette valeur doit être supérieure à zéro. Par défaut, cette valeur est définie sur 300.
  - Connexion : choisissez d'utiliser une connexion nouvelle ou existante :
    - Pour utiliser une connexion existante, sélectionnez Utiliser une connexion existante et sélectionnez la connexion dans la liste déroulante.
    - Pour créer une nouvelle connexion pour cette destination, sélectionnez Créer une nouvelle connexion, puis définissez le Nom, le Type de destination et le Type d'autorisation de la connexion. Vous pouvez également ajouter une Description facultative pour cette connexion.

Pour de plus amples informations, veuillez consulter ???.

Service AWS

Pour utiliser un Service AWS, sélectionnez Service AWS, puis procédez comme suit :

1. Pour Sélectionner une cible, sélectionnez un Service AWS à utiliser comme cible. Fournissez les informations demandées pour le service que vous sélectionnez.

> **Note**
>
> Les champs affichés varient en fonction du service sélectionné. Pour plus d'informations sur les cibles disponibles, consultez Cibles disponibles dans la EventBridge console.

2. Pour de nombreux types de cibles, EventBridge a besoin d'autorisations pour envoyer des événements à la cible. Dans ce cas, EventBridge peut créer le rôle IAM nécessaire à l'exécution de votre règle.

   Pour Rôle d'exécution, effectuez l'une des opérations suivantes :

   - Pour créer un nouveau rôle d'exécution pour cette règle :
     a. Sélectionnez Créer un rôle pour cette ressource spécifique.
     b. Entrez un nom pour ce rôle d'exécution ou utilisez le nom généré par EventBridge.
   - Pour utiliser un rôle d'exécution existant pour cette règle :
     a. Sélectionnez Utiliser le rôle existant.
     b. Entrez ou sélectionnez le nom du rôle d'exécution à utiliser dans la liste déroulante.

3. (Facultatif) Pour Réglages supplémentaires, spécifiez l'un des paramètres facultatifs disponibles pour votre type de cible :

   Event bus

   (Facultatif) Pour File d'attente de lettres mortes, indiquez s'il convient d'utiliser une file d'attente Amazon SQS standard en tant que file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente de lettres mortes s'ils ne sont pas correctement remis à la cible. Effectuez l'une des actions suivantes :

- Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
- Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.
- Choisissez Sélectionner une file d'attente Amazon SQS d'un autre compte AWS en tant que file d'attente de lettres mortes et saisissez l'ARN de la file d'attente à utiliser. Vous devez joindre une stratégie basée sur les ressources à la file d'attente qui accorde l'autorisation EventBridge pour lui envoyer des messages.

  Pour de plus amples informations, veuillez consulter Octroi d'autorisations à la file d'attente de lettres mortes.

API destination

1. (Facultatif) Pour Configurer l'entrée cible, choisissez la manière dont vous souhaitez personnaliser le texte envoyé à la cible pour les événements correspondants. Sélectionnez l'une des méthodes suivantes :
   - Événements correspondants : EventBridge envoie l'intégralité de l'événement source d'origine à la cible. Il s'agit de l'option par défaut.
   - Une partie des événements correspondants : EventBridge envoie uniquement la partie spécifiée de l'événement source d'origine à la cible.

     Sous Spécifier la partie de l'événement correspondant, spécifiez un chemin JSON qui définit la partie de l'événement qu'EventBridge doit envoyer à la cible.
   - Constante (texte JSON) : EventBridge envoie uniquement le texte JSON spécifié à la cible. Aucune partie de l'événement source d'origine n'est envoyée.

     Sous Spécifier la constante au format JSON, spécifiez le texte JSON qu'EventBridge doit envoyer à la cible plutôt qu'à l'événement.
   - Transformateur d'entrée : configurez un transformateur d'entrée pour personnaliser le texte qu'EventBridge doit envoyer à la cible. Pour de plus amples informations, veuillez consulter ???.

     a. Sélectionnez Configurer le transformateur d'entrée.

     b. Configurez le transformateur d'entrée en suivant les étapes de la rubrique ???.

2. (Facultatif) Sous Politique de nouvelles tentatives, spécifiez la manière dont EventBridge doit réessayer d'envoyer un événement à une cible après qu'une erreur se soit produite.
   - Durée maximale de l'événement : entrez la durée maximale (en heures, minutes et secondes) pendant laquelle EventBridge doit retenir les événements non traités. La valeur par défaut est 24 heures.
   - Nouvelles tentatives : entrez le nombre maximal de fois où EventBridge doit réessayer d'envoyer un événement à la cible après qu'une erreur se soit produite. La valeur par défaut est 185 fois.
3. (Facultatif) Pour File d'attente de lettres mortes, indiquez s'il convient d'utiliser une file d'attente Amazon SQS standard en tant que file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente de lettres mortes s'ils ne sont pas correctement remis à la cible. Effectuez l'une des actions suivantes :
   - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
   - Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.
   - Choisissez Sélectionner une file d'attente Amazon SQS d'un autre compte AWS en tant que file d'attente de lettres mortes et saisissez l'ARN de la file d'attente à utiliser. Vous devez joindre une stratégie basée sur les ressources à la file d'attente qui accorde l'autorisation EventBridge pour lui envoyer des messages.

     Pour de plus amples informations, veuillez consulter Octroi d’autorisations à la file d’attente de lettres mortes.

AWS service

Notez qu'EventBridge peut ne pas afficher tous les champs suivants pour un service AWS donné.

1. (Facultatif) Pour Configurer l'entrée cible, choisissez la manière dont vous souhaitez personnaliser le texte envoyé à la cible pour les événements correspondants. Sélectionnez l'une des méthodes suivantes :
   - Événements correspondants : EventBridge envoie l'intégralité de l'événement source d'origine à la cible. Il s'agit de l'option par défaut.

- Une partie des événements correspondants : EventBridge envoie uniquement la partie spécifiée de l'événement source d'origine à la cible.

  Sous Spécifier la partie de l'événement correspondant, spécifiez un chemin JSON qui définit la partie de l'événement qu'EventBridge doit envoyer à la cible.

- Constante (texte JSON) : EventBridge envoie uniquement le texte JSON spécifié à la cible. Aucune partie de l'événement source d'origine n'est envoyée.

  Sous Spécifier la constante au format JSON, spécifiez le texte JSON qu'EventBridge doit envoyer à la cible plutôt qu'à l'événement.

- Transformateur d'entrée : configurez un transformateur d'entrée pour personnaliser le texte qu'EventBridge doit envoyer à la cible. Pour de plus amples informations, veuillez consulter ???.

  a. Sélectionnez Configurer le transformateur d'entrée.

  b. Configurez le transformateur d'entrée en suivant les étapes de la rubrique ???.

2. (Facultatif) Sous Politique de nouvelles tentatives, spécifiez la manière dont EventBridge doit réessayer d'envoyer un événement à une cible après qu'une erreur se soit produite.

   - Durée maximale de l'événement : entrez la durée maximale (en heures, minutes et secondes) pendant laquelle EventBridge doit retenir les événements non traités. La valeur par défaut est 24 heures.
   - Nouvelles tentatives : entrez le nombre maximal de fois où EventBridge doit réessayer d'envoyer un événement à la cible après qu'une erreur se soit produite. La valeur par défaut est 185 fois.

3. (Facultatif) Pour File d'attente de lettres mortes, indiquez s'il convient d'utiliser une file d'attente Amazon SQS standard en tant que file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente de lettres mortes s'ils ne sont pas correctement remis à la cible. Effectuez l'une des actions suivantes :

   - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
   - Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.
   - Choisissez Sélectionner une file d'attente Amazon SQS d'un autre compte AWS en tant que file d'attente de lettres mortes et saisissez l'ARN de la file d'attente à utiliser.

Vous devez joindre une stratégie basée sur les ressources à la file d'attente qui accorde l'autorisation EventBridge pour lui envoyer des messages.

Pour de plus amples informations, veuillez consulter Octroi d’autorisations à la file d’attente de lettres mortes.

4. (Facultatif) Sélectionnez Add another target (Ajouter une autre cible) pour ajouter une nouvelle cible pour cette règle.
5. Choisissez Next (Suivant).

Notez qu'EventBridge peut ne pas afficher tous les champs suivants pour un service AWS donné.

Configuration des balises et vérification de la règle

Pour finir, entrez les balises de votre choix pour la règle, puis passez en revue et créez la règle.

Pour configurer les balises, vérifier et créer la règle

1. (Facultatif) Saisissez une ou plusieurs balises pour la règle. Pour de plus amples informations, veuillez consulter EventBridge Balises Amazon.
2. Choisissez Next (Suivant).
3. Passez en revue les détails de la nouvelle règle. Pour apporter des modifications à une section, choisissez le bouton Modifier en regard de cette section.

   Lorsque vous êtes satisfait des détails de la règle, choisissez Créer une règle.

Utilisation du planificateur Amazon EventBridge avec Amazon EventBridge

Le planificateur Amazon EventBridge est un planificateur sans serveur qui vous permet de créer, d'exécuter et de gérer des tâches à partir d'un service central et géré. Avec le planificateur EventBridge, vous pouvez créer des planifications à l'aide d'expressions cron et rate pour les modèles récurrents, voire configurer des invocations ponctuelles. Vous pouvez configurer des fenêtres de temps flexibles pour la livraison, définir des limites de nouvelles tentatives ainsi que la durée de conservation maximale pour les invocations d'API en échec.

Le Planificateur EventBridge est hautement personnalisable et offre une meilleure capacité de mise à l'échelle par rapport aux règles planifiées d'EventBridge, avec un ensemble plus large d'opérations d'API cibles et de services AWS. Nous vous recommandons d'utiliser le Planificateur EventBridge pour invoquer des cibles selon un calendrier.

Rubriques

Configurer le rôle d'exécution

Lorsque vous créez une planification, le planificateur EventBridge doit être autorisé à invoquer son opération d'API cible en votre nom. Vous accordez ces autorisations au planificateur EventBridge à l'aide d'un rôle d'exécution. La politique d'autorisation que vous associez au rôle d'exécution de votre planification définit les autorisations requises. Ces autorisations dépendent de l'API cible que vous souhaitez que le planificateur EventBridge invoque.

Lorsque vous utilisez la console du planificateur EventBridge pour créer une planification, comme dans la procédure suivante, le planificateur EventBridge définit automatiquement un rôle d'exécution en fonction de la cible que vous avez sélectionnée. Si vous souhaitez créer une planification à l'aide de l'un des kits SDK du planificateur EventBridge, de la AWS CLI ou de AWS CloudFormation, vous devez disposer d'un rôle d'exécution existant qui accorde les autorisations dont le planificateur EventBridge a besoin pour invoquer une cible. Pour plus d'informations sur la configuration manuelle d'un rôle d'exécution pour votre planification, voir Configuration d'un rôle d'exécution dans le Guide de l'utilisateur du planificateur EventBridge.

Créer une planification

Pour créer une planification à l'aide de la console

1. Ouvrez la console du planificateur Amazon EventBridge à l'adresse https://console.aws.amazon.com/scheduler/home.
2. Sur la page Planifications, choisissez Créer une planification.
3. Sur la page Spécifier le détail de la planification, dans la section Nom et description de la planification, procédez comme suit :

   a. Pour Nom de la planification, saisissez un nom à attribuer à votre planification. Par exemple, **`MyTestSchedule`**.

   b. (Facultatif) Dans le champ Description, saisissez une description de la planification. Par exemple, **`My first schedule`**.

   c. Pour Groupe de planifications, choisissez un groupe de planifications dans la liste déroulante. Si vous n'avez pas de groupe, choisissez par défaut. Pour créer un groupe de planifications, choisissez Crée votre propre planification.

   Vous utilisez des groupes de planifications pour leur ajouter des balises.

4. • Choisissez vos options de planification.

Occurrence	Faites ceci...	
Planification ponctuelle Une planification ponctuelle n'invoque un objectif qu'une seule fois à la date et à l'heure que vous indiquez.	Pour Date et heure, procédez comme suit : • Entrez une date valide au format `YYYY/MM/DD` . • Entrez un horodatage au format `hh:mm` de 24 heures. • Dans le champ Fuseau horaire, choisissez le fuseau horaire.	
Planification récurrente	a. Pour Schedule type (Planifier le type),	

Occurrence	Faites ceci...	
Une planification récurrente invoque un objectif à un taux que vous spécifiez à l'aide d'une expression cron ou d'une expression rate.	effectuez l'une des étapes suivantes : • Pour utiliser une expression cron afin de définir la planification, choisissez Planification basée sur cron et entrez l'expression cron. • Pour utiliser une expression de rythme pour définir la planification, choisissez Planification basée sur le rythme. Pour plus d'informations sur les expressions cron et rate, consultez Types de planifications sur le planificateur EventBridge dans le Guide de l'utilisateur du planificateur Amazon EventBridge. b. Pour Fenêtre temporelle flexible, choisissez Désactivé pour désactiver cette option ou choisir l'une des fenêtres temporelles prédéfinies. Par exemple, si vous choisissez15 minutes et	

Occurrence	Faites ceci...	
	que vous définissez une planification récurrente pour invoquer son objectif une fois par heure, la planification s'exécute dans les 15 minutes suivant le début de chaque heure.	

5. (Facultatif) Si vous avez choisi Planification récurrente à l'étape précédente, dans la section Délai, procédez comme suit :

   a. Dans le champ Fuseau horaire, choisissez un fuseau horaire.

   b. Pour Date et heure de début, entrez une date valide au format `YYYY/MM/DD`, puis spécifiez un horodatage au format `hh:mm` de 24 heures.

   c. Pour Date et heure de fin, entrez une date valide au format `YYYY/MM/DD`, puis spécifiez un horodatage au format `hh:mm` de 24 heures.

6. Choisissez Next (Suivant).

7. Sur la page Sélectionner la cible, choisissez l'opération d'API AWS invoquée par le planificateur EventBridge :

   a. Pour API de la cible, choisissez Cibles modélisées.

   b. Choisissez Événements Put Amazon EventBridge.

   c. Sous Événements Put, spécifiez ce qui suit :

   - Pour Bus d'événements EventBridge, choisissez le bus d'événements dans le menu déroulant. Par exemple, **`default`**.

     Vous pouvez également créer un bus d'événements dans la console EventBridge en choisissant Créer un nouveau bus d'événements.
   - Pour Detail-type, entrez le type de détail des événements qui doivent correspondre. Par exemple, **`Object Created`**.
   - Dans Source, entrez le nom du service qui correspond à la source des événements.

Pour les événements d'un service AWS, spécifiez le préfixe du service en guise de source. N'incluez pas le préfixe `aws.`. Par exemple, pour les événements Amazon S3, entrez **`s3`**.

Pour déterminer le préfixe d'un service, consultez le tableau des clés de condition dans le guide de référence de l'autorisation de service. Pour plus d'informations sur les valeurs source et detail-type des événements, consultez ???.

- (Facultatif) Pour detail, entrez un modèle d'événement pour filtrer davantage les événements que le planificateur EventBridge envoie à EventBridge.

  Pour de plus amples informations, veuillez consulter ???.

8. Choisissez Next (Suivant).
9. Sur la page Settings (Paramètres), procédez comme suit :

   a. Pour activer la planification, sous État de la planification, activez Activer la planification.

   b. Pour configurer une stratégie de nouvelles tentatives pour votre planification, sous Politique de nouvelle tentative et file d'attente de lettres mortes (DLQ), procédez comme suit :

   - Activez Réessayer.
   - Pour Âge maximum de l'événement, entrez le nombre maximum d'heures et de minutes de conservation d'un événement non traité par le planificateur EventBridge.
   - La durée maximale est 24 heures.
   - Pour Nombre maximum de tentatives, entrez le nombre maximum de tentatives de renvoi d'une erreur par le planificateur EventBridge.

     La valeur maximale est 185 nouvelles tentatives.

   Avec les stratégies de nouvelles tentatives, si une planification ne parvient pas à invoquer sa cible, le planificateur EventBridge la réexécute. Si elle est configurée, vous devez définir la durée de rétention maximale et les nouvelles tentatives pour la planification.

   c. Choisissez où le planificateur EventBridge stocke les événements non livrés.

Option File d'attente de lettres mortes (DLQ)	Faites ceci...	
Ne stockez pas	Sélectionnez Aucun.	
Stocker l'événement dans le même Compte AWS où vous créez la planification	a. Choisissez Sélectionnez une file d'attente Amazon SQS dans mon Compte AWS en tant que DLQ. b. Choisissez l'Amazon Resource Name (ARN) de la file d'attente Amazon SQS.	
Stocker l'événement dans un autre Compte AWS que celui où vous créez la planification	a. Choisissez Spécifier une file d'attente Amazon SQS dans un autre Comptes AWS en tant que DLQ. b. Entrez l'Amazon Resource Name (ARN) de la file d'attente Amazon SQS.	

d. Pour utiliser une clé gérée par le client afin de chiffrer votre entrée cible, sous Chiffrement, choisissez Personnaliser les paramètres de chiffrement (avancé).

   Si vous choisissez cette option, entrez un ARN de clé KMS existant ou choisissez Créez un AWS KMS keypour accéder à la console AWS KMS. Pour plus d'informations sur la façon dont le planificateur EventBridge chiffre vos données au repos, voir Chiffrement au repos dans le Guide de l'utilisateur du planificateur Amazon EventBridge.

e. Pour que le planificateur EventBridge crée un rôle d'exécution pour vous, choisissez Créer un rôle pour cette planification. Ensuite, saisissez un nom pour Nom du rôle. Si vous choisissez cette option, le planificateur EventBridge associe au rôle les autorisations requises pour votre cible modélisée.

10. Choisissez Next (Suivant).
11. Sur la page Examiner et créer une planification, examinez les détails de votre planification. Dans chaque section, choisissez Modifier pour revenir à cette étape et modifier ses détails.
12. Choisissez Créer une planification.

    Vous pouvez consulter la liste de vos planifications nouvelles et existantes sur la page Planifications. Sous la colonne État, vérifiez que votre nouvelle planification est activée.

Ressources connexes

Pour de plus amples informations sur le planificateur EventBridge, veuillez consulter les ressources suivantes :

- Guide de l'utilisateur du planificateur EventBridge
- Référence de l'API du planificateur EventBridge
- Tarification du planificateur EventBridge

Création d'une règle Amazon EventBridge qui s'exécute selon un calendrier

Une règle peut être exécutée en réponse à un événement ou à certains intervalles de temps. Par exemple, pour exécuter régulièrement une fonction AWS Lambda, vous pouvez créer une règle à exécuter selon un calendrier.

> **Note**
>
> EventBridge propose le planificateur Amazon EventBridge, un planificateur sans serveur qui vous permet de créer, d'exécuter et de gérer des tâches à partir d'un service central et géré. Le Planificateur EventBridge est hautement personnalisable et offre une meilleure capacité de mise à l'échelle par rapport aux règles planifiées d'EventBridge, avec un ensemble plus large d'opérations d'API cibles et de services AWS.
> Nous vous recommandons d'utiliser le Planificateur EventBridge pour invoquer des cibles selon un calendrier. Pour de plus amples informations, veuillez consulter ???.

Dans EventBridge, vous pouvez créer deux types de règles planifiées :

- Règles qui s'exécutent à fréquence régulière

  EventBridge exécute ces règles à intervalles réguliers, toutes les 20 minutes par exemple.

  Pour spécifier le taux d'une règle planifiée, vous définissez une valeur de déclenchement.
- Règles qui s'exécutent à des moments précis

  EventBridge exécute ces règles à des heures et à des dates spécifiques, à 8 h 00 PST le premier lundi de chaque mois par exemple.

  Pour spécifier l'heure et les dates d'exécution d'une règle planifiée, vous définissez une expression cron.

Les valeurs de déclenchement sont plus simples à définir, tandis que les expressions cron offrent un contrôle détaillé du calendrier. Par exemple, une expression cron vous permet de définir une règle qui s'exécute à une heure spécifiée un certain jour de chaque semaine ou mois. En revanche, les valeurs de déclenchement exécutent une règle à une fréquence standard, par exemple une fois toutes les heures ou une fois par jour.

Tous les événements planifiés utilisent le fuseau horaire UTC+0 et la précision minimale d'un calendrier est de 1 minute.

> **Note**
>
> EventBridge ne fournit pas de précision de deuxième niveau dans les expressions de calendrier. Le niveau de résolution maximal lors de l'utilisation d'une expression cron est d'une minute. Compte tenu de la nature distribuée d'EventBridge et des services cibles, un décalage de plusieurs secondes peut être observé entre le moment où la règle planifiée est déclenchée et le moment où le service cible exécute la ressource cible.

La vidéo suivante donne un aperçu de la planification des tâches : Création de tâches planifiées avec EventBridge

Rubriques

- Création d'une règle qui s'exécute selon un calendrier
- Référence des expressions cron

- Référence des valeurs de déclenchement

Création d'une règle qui s'exécute selon un calendrier

Les étapes suivantes expliquent comment créer une règle EventBridge qui s'exécute selon un calendrier régulier.

> **Note**
>
> Vous pouvez uniquement créer des règles planifiées à l'aide du bus d'événements par défaut.

Étapes

- Définition de la règle
- Définition du calendrier
- Sélection des cibles
- Configuration des balises et vérification de la règle

Définition de la règle

Commencez par entrer un nom et une description pour identifier la règle.

Pour définir les détails de la règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule (Créer une règle).
4. Entrez un Nom et éventuellement une Description pour la règle.

   Une règle ne peut pas avoir le même nom qu'une autre règle de la même Région AWS et sur le même bus d'événement.
5. Pour Bus d'évènements, choisissez le bus d'événements par défaut. Vous pouvez uniquement créer des règles planifiées à l'aide du bus d'événements par défaut.
6. Pour que la règle entre en vigueur dès sa création, assurez-vous que l'option Activer la règle sur le bus d'événements sélectionné est activée.
7. Pour Rule type (Type de règle), choisissez Schedule (Planifier).

À ce stade, vous pouvez choisir de continuer à créer une règle qui s'exécute selon un calendrier ou d'utiliser le Planificateur Amazon EventBridge.

8. Choisissez ce que vous souhaitez faire ensuite :

   - Utilisez le planificateur EventBridge pour créer votre calendrier

     > **Note**
     >
     > Le Planificateur EventBridge est un planificateur sans serveur qui vous permet de créer, d'exécuter et de gérer des tâches à partir d'un service central et géré. Il fournit une fonctionnalité de planification ponctuelle et récurrente indépendamment des bus et des règles d'événement. Le Planificateur EventBridge est hautement personnalisable et offre une meilleure capacité de mise à l'échelle par rapport aux règles planifiées d'EventBridge, avec un ensemble plus large d'opérations d'API cibles et de services AWS.
     >
     > Nous vous recommandons d'utiliser le Planificateur EventBridge pour invoquer des cibles selon un calendrier. Pour plus d'informations, consultez Qu'est-ce que le planificateur Amazon EventBridge ? dans le Guide de l'utilisateur du planificateur Amazon EventBridge.

     1. Sélectionnez Continuer dans EventBridge Scheduler

        EventBridge ouvre la console du planificateur EventBridge à la page Créer une planification.

     2. Créez le calendrier dans la console du Planificateur EventBridge.

   - Continuez à utiliser EventBridge pour créer une règle planifiée pour le bus d'événements par défaut

     1. Sélectionnez Continuer à créer la règle.

Définition du calendrier

Vous allez ensuite définir le modèle de calendrier.

Pour définir le modèle de calendrier

1. Pour Modèle de planification, choisissez si vous souhaitez exécuter le calendrier à une heure précise ou à une fréquence régulière :

Specific time

1. Choisissez Un calendrier détaillé qui s'exécute à une heure précise, à 8 h 00 PST le premier lundi de chaque mois par exemple.
2. Pour Expression cron, spécifiez des champs pour définir l'expression cron qu'EventBridge doit utiliser pour déterminer quand exécuter cette règle planifiée.

   Une fois que vous avez spécifié tous les champs, EventBridge affiche les dix prochaines dates auxquelles il exécutera cette règle planifiée. Vous pouvez choisir d'afficher ces dates au format UTC ou au Fuseau horaire local.

   Pour plus d'informations sur la construction d'une expression cron, consultez ???.

Regular rate

1. Choisissez Un calendrier qui s'exécute à une fréquence régulière, toutes les 10 minutes par exemple.
2. Pour Expression de fréquence, spécifiez les champs Valeur et Unité pour définir la fréquence à laquelle EventBridge doit exécuter cette règle planifiée.

   Pour plus d'informations sur la construction d'une valeur de déclenchement, consultez ???.

2. Choisissez Next (Suivant).

Sélection des cibles

Choisissez une ou plusieurs cibles pour recevoir des événements correspondant au modèle spécifié. Les cibles peuvent inclure un bus d'événements EventBridge, des destinations d'API EventBridge, y compris des partenaires SaaS tels que Salesforce, ou un autre Service AWS.

Pour sélectionner des cibles

1. Pour Type de cible, choisissez l'un des types de cibles suivants :

   Event bus

   Pour sélectionner un bus d'événements EventBridge, sélectionnez Bus d'événements EventBridge, puis procédez comme suit :

- Pour utiliser un bus d'événements dans la même Région AWS que cette règle :
  1. Sélectionnez Bus d'événements dans le même compte et la même région.
  2. Pour Bus d'événements pour la cible, cliquez sur la liste déroulante et entrez le nom du bus d'événements. Vous pouvez également sélectionner le bus d'événements dans la liste déroulante.

     Pour de plus amples informations, veuillez consulter ???.
- Pour utiliser un bus d'événements dans une autre Région AWS ou un autre compte que cette règle :
  1. Sélectionnez Bus d'événements dans un compte ou une région différent.
  2. Pour Bus d'événements comme cible, entrez l'ARN du bus d'événements que vous souhaitez utiliser.

     Pour plus d'informations, consultez :
     - ???
     - ???

API destination

Pour utiliser une destination d'API EventBridge, sélectionnez Destination d'API EventBridge, puis effectuez l'une des opérations suivantes :

- Pour utiliser une destination d'API existante, sélectionnez Utiliser une destination d'API existante. Ensuite, sélectionnez une destination d'API dans la liste déroulante.
- Pour créer une nouvelle destination d'API, sélectionnez Créer une nouvelle destination API. Fournissez ensuite les informations suivantes pour la destination :
  - Nom : entrez un nom pour la destination.

    Les noms doivent être uniques dans votre Compte AWS. Les noms peuvent comporter jusqu'à 64 caractères. Les caractères valides sont A-Z, a-z, 0-9 et . _ - (tiret).
  - (Facultatif) Description : entrez une description pour la destination.

    Les descriptions peuvent comporter jusqu'à 512 caractères.
  - Point de terminaison de la destination d'API : point de terminaison d'URL de la cible.

L'URL du point de terminaison doit commencer par **`https`**. Vous pouvez inclure le caractère générique **`*`** en tant que paramètre de chemin. Vous pouvez définir les paramètres du chemin à partir de l'attribut `HttpParameters` de la cible.

- Méthode HTTP : sélectionnez la méthode HTTP utilisée lorsque vous invoquez le point de terminaison.
- (Facultatif) Limite du taux d'appel par seconde : entrez le nombre maximal d'invocations acceptées par seconde pour cette destination.

  Cette valeur doit être supérieure à zéro. Par défaut, cette valeur est définie sur 300.
- Connexion : choisissez d'utiliser une connexion nouvelle ou existante :
  - Pour utiliser une connexion existante, sélectionnez Utiliser une connexion existante et sélectionnez la connexion dans la liste déroulante.
  - Pour créer une nouvelle connexion pour cette destination, sélectionnez Créer une nouvelle connexion, puis définissez le Nom, le Type de destination et le Type d'autorisation de la connexion. Vous pouvez également ajouter une Description facultative pour cette connexion.

Pour de plus amples informations, veuillez consulter ???.

Service AWS

Pour utiliser un Service AWS, sélectionnez Service AWS, puis procédez comme suit :

1. Pour Sélectionner une cible, sélectionnez un Service AWS à utiliser comme cible. Fournissez les informations demandées pour le service que vous sélectionnez.

   > **Note**
   >
   > Les champs affichés varient en fonction du service sélectionné. Pour plus d'informations sur les cibles disponibles, consultez Cibles disponibles dans la EventBridge console.

2. Pour de nombreux types de cibles, EventBridge a besoin d'autorisations pour envoyer des événements à la cible. Dans ce cas, EventBridge peut créer le rôle IAM nécessaire à l'exécution de votre règle.

   Pour Rôle d'exécution, effectuez l'une des opérations suivantes :

- Pour créer un nouveau rôle d'exécution pour cette règle :

  a. Sélectionnez Créer un rôle pour cette ressource spécifique.

  b. Entrez un nom pour ce rôle d'exécution ou utilisez le nom généré par EventBridge.

- Pour utiliser un rôle d'exécution existant pour cette règle :

  a. Sélectionnez Utiliser le rôle existant.

  b. Entrez ou sélectionnez le nom du rôle d'exécution à utiliser dans la liste déroulante.

3. (Facultatif) Pour Réglages supplémentaires, spécifiez l'un des paramètres facultatifs disponibles pour votre type de cible :

   Event bus

   (Facultatif) Pour File d'attente de lettres mortes, indiquez s'il convient d'utiliser une file d'attente Amazon SQS standard en tant que file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente de lettres mortes s'ils ne sont pas correctement remis à la cible. Effectuez l'une des actions suivantes :

   - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
   - Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.
   - Choisissez Sélectionner une file d'attente Amazon SQS d'un autre compte AWS en tant que file d'attente de lettres mortes et saisissez l'ARN de la file d'attente à utiliser. Vous devez joindre une stratégie basée sur les ressources à la file d'attente qui accorde l'autorisation EventBridge pour lui envoyer des messages.

     Pour de plus amples informations, veuillez consulter Octroi d’autorisations à la file d’attente de lettres mortes.

   API destination

   1. (Facultatif) Pour Configurer l'entrée cible, choisissez la manière dont vous souhaitez personnaliser le texte envoyé à la cible pour les événements correspondants. Sélectionnez l'une des méthodes suivantes :

- Événements correspondants : EventBridge envoie l'intégralité de l'événement source d'origine à la cible. Il s'agit de l'option par défaut.
- Une partie des événements correspondants : EventBridge envoie uniquement la partie spécifiée de l'événement source d'origine à la cible.

  Sous Spécifier la partie de l'événement correspondant, spécifiez un chemin JSON qui définit la partie de l'événement qu'EventBridge doit envoyer à la cible.
- Constante (texte JSON) : EventBridge envoie uniquement le texte JSON spécifié à la cible. Aucune partie de l'événement source d'origine n'est envoyée.

  Sous Spécifier la constante au format JSON, spécifiez le texte JSON qu'EventBridge doit envoyer à la cible plutôt qu'à l'événement.
- Transformateur d'entrée : configurez un transformateur d'entrée pour personnaliser le texte qu'EventBridge doit envoyer à la cible. Pour de plus amples informations, veuillez consulter ???.

  a. Sélectionnez Configurer le transformateur d'entrée.

  b. Configurez le transformateur d'entrée en suivant les étapes de la rubrique ???.

2. (Facultatif) Sous Politique de nouvelles tentatives, spécifiez la manière dont EventBridge doit réessayer d'envoyer un événement à une cible après qu'une erreur se soit produite.
   - Durée maximale de l'événement : entrez la durée maximale (en heures, minutes et secondes) pendant laquelle EventBridge doit retenir les événements non traités. La valeur par défaut est 24 heures.
   - Nouvelles tentatives : entrez le nombre maximal de fois où EventBridge doit réessayer d'envoyer un événement à la cible après qu'une erreur se soit produite. La valeur par défaut est 185 fois.
3. (Facultatif) Pour File d'attente de lettres mortes, indiquez s'il convient d'utiliser une file d'attente Amazon SQS standard en tant que file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente de lettres mortes s'ils ne sont pas correctement remis à la cible. Effectuez l'une des actions suivantes :
   - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
   - Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.

- Choisissez Sélectionner une file d'attente Amazon SQS d'un autre compte AWS en tant que file d'attente de lettres mortes et saisissez l'ARN de la file d'attente à utiliser. Vous devez joindre une stratégie basée sur les ressources à la file d'attente qui accorde l'autorisation EventBridge pour lui envoyer des messages.

  Pour de plus amples informations, veuillez consulter Octroi d'autorisations à la file d'attente de lettres mortes.

AWS service

Notez qu'EventBridge peut ne pas afficher tous les champs suivants pour un service AWS donné.

1. (Facultatif) Pour Configurer l'entrée cible, choisissez la manière dont vous souhaitez personnaliser le texte envoyé à la cible pour les événements correspondants. Sélectionnez l'une des méthodes suivantes :
   - Événements correspondants : EventBridge envoie l'intégralité de l'événement source d'origine à la cible. Il s'agit de l'option par défaut.
   - Une partie des événements correspondants : EventBridge envoie uniquement la partie spécifiée de l'événement source d'origine à la cible.

     Sous Spécifier la partie de l'événement correspondant, spécifiez un chemin JSON qui définit la partie de l'événement qu'EventBridge doit envoyer à la cible.
   - Constante (texte JSON) : EventBridge envoie uniquement le texte JSON spécifié à la cible. Aucune partie de l'événement source d'origine n'est envoyée.

     Sous Spécifier la constante au format JSON, spécifiez le texte JSON qu'EventBridge doit envoyer à la cible plutôt qu'à l'événement.
   - Transformateur d'entrée : configurez un transformateur d'entrée pour personnaliser le texte qu'EventBridge doit envoyer à la cible. Pour de plus amples informations, veuillez consulter ???.

     a. Sélectionnez Configurer le transformateur d'entrée.

     b. Configurez le transformateur d'entrée en suivant les étapes de la rubrique ???.
2. (Facultatif) Sous Politique de nouvelles tentatives, spécifiez la manière dont EventBridge doit réessayer d'envoyer un événement à une cible après qu'une erreur se soit produite.

- Durée maximale de l'événement : entrez la durée maximale (en heures, minutes et secondes) pendant laquelle EventBridge doit retenir les événements non traités. La valeur par défaut est 24 heures.
- Nouvelles tentatives : entrez le nombre maximal de fois où EventBridge doit réessayer d'envoyer un événement à la cible après qu'une erreur se soit produite. La valeur par défaut est 185 fois.

3. (Facultatif) Pour File d'attente de lettres mortes, indiquez s'il convient d'utiliser une file d'attente Amazon SQS standard en tant que file d'attente de lettres mortes. EventBridge envoie les événements qui correspondent à cette règle à la file d'attente de lettres mortes s'ils ne sont pas correctement remis à la cible. Effectuez l'une des actions suivantes :
   - Choisissez None (Aucune) pour ne pas utiliser de file d'attente de lettres mortes.
   - Choisissez Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Sélectionner une file d'attente Amazon SQS du compte AWS actuel à utiliser en tant que file d'attente de lettres mortes) et sélectionnez la file d'attente à utiliser dans la liste déroulante.
   - Choisissez Sélectionner une file d'attente Amazon SQS d'un autre compte AWS en tant que file d'attente de lettres mortes et saisissez l'ARN de la file d'attente à utiliser. Vous devez joindre une stratégie basée sur les ressources à la file d'attente qui accorde l'autorisation EventBridge pour lui envoyer des messages.

     Pour de plus amples informations, veuillez consulter Octroi d'autorisations à la file d'attente de lettres mortes.

4. (Facultatif) Sélectionnez Add another target (Ajouter une autre cible) pour ajouter une nouvelle cible pour cette règle.
5. Choisissez Next (Suivant).

Configuration des balises et vérification de la règle

Pour finir, entrez les balises de votre choix pour la règle, puis passez en revue et créez la règle.

Pour configurer les balises, vérifier et créer la règle

1. (Facultatif) Saisissez une ou plusieurs balises pour la règle. Pour de plus amples informations, veuillez consulter EventBridge Balises Amazon.
2. Choisissez Next (Suivant).

3. Passez en revue les détails de la nouvelle règle. Pour apporter des modifications à une section, choisissez le bouton Modifier en regard de cette section.

   Lorsque vous êtes satisfait des détails de la règle, choisissez Créer une règle.

Référence des 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	, - * /
Jour de la semaine	1-7 ou DIM-SAM	, - * ? L #
Année	1970-2199	, - * /

Caractères génériques

- Le caractère générique , (virgule) inclut des valeurs supplémentaires. Dans le champ Month, JAN,FEB,MAR englobe January, February et March.
- Le caractère générique - (tiret) spécifie des plages. Dans le champ Day, 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 (Heures), * inclut toutes les heures. Vous ne pouvez pas utiliser * à la fois dans les champs Day-of-month et Day-of-week. 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 l'un des jours de la semaine est acceptable, vous pouvez entrer ? dans le champ Day-of-week.
- Le caractère générique L dans les champs Jour du mois ou Jour de la semaine spécifie le dernier jour du mois ou de la semaine.
- Le caractère générique **`W`** dans le champ Jour du mois spécifie un jour de la semaine. Dans le champ Jour du mois, **`3W`** indique le jour le plus proche du troisième jour de la semaine du mois.
- Le caractère générique # dans le champ Jour de la semaine 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.

Limites

- Vous ne pouvez pas spécifier les champs Jour du mois et Jour de la semaine dans une même expression cron. Si vous spécifiez une valeur ou le caractère * (astérisque) dans l'un de ces champs, vous devez utiliser le caractère ? (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.

Exemples

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	*	*	?	*	Exécution à 10 h 00 (UTC+0) tous les jours
15	12	*	*	?	*	Exécution à 12 h 15 (UTC+0) tous les jours
0	18	?	*	MON-FRI	*	Exécution à 18 h 00 (UTC+0) du lundi au vendredi
0	8	1	*	?	*	Exécution à 8 h 00 (UTC+0) 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

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
0/5	8-17	?	*	MON-FRI	*	Exécution toutes les 5 minutes du lundi au vendredi entre 8 h 00 et 17 h 55 (UTC+0)
0/30	20-2	?	*	MON-FRI	*	Exécution toutes les 30 minutes du lundi au vendredi entre 22 h 00 le jour de départ et 2 h 00 le jour suivant (UTC) Exécution de 00 h 00 à 2 h 00 le lundi matin (UTC).

L'exemple suivant crée une règle qui s'exécute tous les jours à 12 h 00 (UTC+0).

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

L'exemple suivant crée une règle qui s'exécute tous les jours à 14 h 05 et 14 h 35 (UTC+0).

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

L'exemple suivant crée une règle qui s'exécute à 10 h 15 (UTC+0) le dernier vendredi de chaque mois, pendant les années 2019 à 2022.

```
aws events put-rule --schedule-expression "cron(15 10 ? * 6L 2019-2022)" --name MyRule3
```

Référence des valeurs de déclenchement

Une valeur de déclenchement démarre au moment où vous créez la règle d'événement planifiée, puis s'exécute selon un calendrier défini.

Les valeurs de déclenchement se composent de deux champs obligatoires séparés par des espaces.

Syntaxe

```
rate(value unit)
```

value

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

Limites

Si la valeur est égale à 1, l'unité doit être au singulier. Si la valeur est supérieure à 1, l'unité doit être au pluriel. Par exemple, rate(1 hours) et rate(5 hour) ne sont pas valides, mais rate(1 hour) et rate(5 hours) sont valides.

Exemples

Les exemples suivants indiquent comment utiliser les valeurs de déclenchement avec la commande AWS CLI `put-rule`. Le premier exemple déclenche la règle toutes les minutes, le deuxième toutes les cinq minutes, le troisième une fois par heure et le quatrième 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
```

Désactivation ou suppression d'une règle Amazon EventBridge

Pour empêcher une règle de traiter des événements ou de s'exécuter selon un calendrier, vous pouvez la supprimer ou la désactiver. Les étapes suivantes expliquent comment supprimer ou désactiver une règle EventBridge.

Pour supprimer ou désactiver une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Rules.

   Sous Event bus (Bus d'événement), sélectionnez le bus d'événement associé à la règle.
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, entrez 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 Force delete (Forcer la suppression).

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

      Vous ne pouvez pas désactiver une règle gérée.

Bonnes pratiques lors de la définition de règles Amazon EventBridge

Vous trouverez ci-dessous certaines bonnes pratiques à prendre en compte lorsque vous créez des règles pour vos bus d'événements.

Définition d'une cible unique pour chaque règle

Bien que vous puissiez spécifier jusqu'à cinq cibles pour une règle donnée, la gestion des règles est plus facile lorsque vous spécifiez une cible unique pour chaque règle. Si plusieurs cibles doivent recevoir le même ensemble d'événements, nous vous recommandons de dupliquer la règle pour

livrer les mêmes événements à différentes cibles. Cette encapsulation simplifie la gestion des règles : si les besoins des cibles d'événements divergent au fil du temps, vous pouvez mettre à jour chaque règle et son modèle d'événement indépendamment des autres.

Définition d'autorisations de règle

Vous pouvez permettre aux composants ou aux services d'applications consommatrices d'événements de contrôler la gestion de leurs propres règles. Une approche architecturale couramment adoptée par les clients consiste à isoler ces composants ou services d'applications en utilisant des comptes AWS distincts. Pour permettre le flux d'événements d'un compte à un autre, vous devez créer une règle sur un bus d'événements qui route les événements vers un bus d'événements d'un autre compte. Vous pouvez permettre aux équipes ou aux services consommateurs d'événements de contrôler la gestion de leurs propres règles. Pour ce faire, spécifiez les autorisations appropriées pour leurs comptes par le biais de politiques de ressources. Cela fonctionne pour tous les comptes et toutes les régions.

Pour de plus amples informations, veuillez consulter ???.

Pour obtenir des exemples de politiques de ressources, consultez Modèles de conception sur plusieurs comptes avec Amazon EventBridge (langue française non garantie) sur GitHub.

Surveillance des performances des règles

Surveillez vos règles pour vous assurer qu'elles fonctionnent comme prévu :

- Surveillez la métrique `TriggeredRules` pour détecter les points de données manquants ou les anomalies. Cela peut vous aider à détecter les incohérences chez un diffuseur de publication qui a apporté une modification majeure. Pour de plus amples informations, veuillez consulter ???.
- L'alarme en cas d'anomalies ou le nombre maximal attendu peuvent également aider à détecter la correspondance entre une règle et de nouveaux événements. Cela peut se produire lorsque les diffuseurs de publication d'événements, y compris les services AWS et les partenaires SaaS, introduisent de nouveaux événements lors de l'activation de nouveaux cas d'utilisation et de nouvelles fonctionnalités. Lorsque ces nouveaux événements sont inattendus et génèrent un volume supérieur au taux de traitement de la cible en aval, ils peuvent entraîner un backlog d'événements.

  Ce traitement des événements inattendus peut également entraîner la facturation de frais indésirables.

Cela peut également déclencher une limitation des règles lorsque le compte dépasse son quota de service d'invocations cibles agrégées par seconde. EventBridge essaiera toujours de livrer des événements correspondant à des règles limitées et réessaiera dans les 24 heures ou conformément à la politique de nouvelle tentative personnalisée de la cible. Vous pouvez détecter et alerter les règles limitées à l'aide de la métrique `ThrottledRules`.

- Pour les cas d'utilisation à faible latence, vous pouvez également surveiller la latence en utilisant `IngestionToInvocationStartLatency`, qui indique l'état de votre bus d'événements. Toute période prolongée de latence élevée supérieure à 30 secondes peut indiquer une interruption de service ou une limitation des règles.

Utilisation d'Amazon EventBridge et de modèles AWS Serverless Application Model

Vous pouvez créer et tester des règles manuellement dans la console EventBridge, ce qui peut faciliter le processus de développement en affinant les modèles d'événements. Cependant, une fois que vous êtes prêt à déployer votre application, il est plus facile d'utiliser un framework tel qu'AWS SAM pour lancer toutes vos ressources sans serveur de manière cohérente.

Nous allons utiliser cet exemple d'application pour découvrir comment utiliser des modèles AWS SAM pour créer des ressources EventBridge. Dans cet exemple, le fichier template.yaml est un modèle AWS SAM qui définit quatre fonctions AWS Lambda et qui montre deux manières différentes d'intégrer les fonctions Lambda à EventBridge.

Pour une présentation détaillée de cet exemple d'application, consultez ???.

Pour utiliser EventBridge et les modèles AWS SAM, deux approches sont possibles. Pour les intégrations simples où une fonction Lambda est invoquée par une règle, l'approche du modèle combiné est recommandée. Si votre logique de routage est complexe ou si vous vous connectez à des ressources extérieures à votre modèle AWS SAM, l'approche du modèle séparé est le meilleur choix.

Approches :

- Modèle combiné
- Modèle séparé

Modèle combiné

La première approche utilise la propriété `Events` pour configurer la règle EventBridge. L'exemple de code suivant définit un événement qui invoque votre fonction Lambda.

> **Note**
>
> Cet exemple crée automatiquement la règle sur le bus d'événements par défaut, qui existe dans tous les comptes AWS. Pour associer la règle à un bus d'événements personnalisé, vous pouvez ajouter `EventBusName` au modèle.

```
atmConsumerCase3Fn:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: atmConsumer/
    Handler: handler.case3Handler
    Runtime: nodejs12.x
    Events:
      Trigger:
        Type: CloudWatchEvent
        Properties:
          Pattern:
            source:
              - custom.myATMapp
            detail-type:
              - transaction
            detail:
              result:
                - "anything-but": "approved"
```

Ce code YAML est équivalent à un modèle d'événement dans la console EventBridge. Dans YAML, il vous suffit de définir le modèle d'événement pour qu'AWS SAM crée automatiquement un rôle IAM avec les autorisations requises.

Modèle séparé

Dans la seconde approche de définition d'une configuration EventBridge dans AWS SAM, les ressources sont séparées plus clairement dans le modèle.

1. Vous devez d'abord définir la fonction Lambda :

   ```
   atmConsumerCase1Fn:
     Type: AWS::Serverless::Function
     Properties:
       CodeUri: atmConsumer/
       Handler: handler.case1Handler
       Runtime: nodejs12.x
   ```

2. Définissez ensuite la règle à l'aide d'une ressource `AWS::Events::Rule`. Les propriétés définissent le modèle d'événement et peuvent également spécifier des cibles. Vous pouvez définir plusieurs cibles de manière explicite.

```
EventRuleCase1:
  Type: AWS::Events::Rule
  Properties:
    Description: "Approved transactions"
    EventPattern:
      source:
        - "custom.myATMapp"
      detail-type:
        - transaction
      detail:
        result:
          - "approved"
    State: "ENABLED"
    Targets:
      -
        Arn:
          Fn::GetAtt:
            - "atmConsumerCase1Fn"
            - "Arn"
        Id: "atmConsumerTarget1"
```

3. Enfin, définissez une ressource `AWS::Lambda::Permission` qui autorise EventBridge à invoquer la cible.

```
PermissionForEventsToInvokeLambda:
  Type: AWS::Lambda::Permission
  Properties:
    FunctionName:
      Ref: "atmConsumerCase1Fn"
    Action: "lambda:InvokeFunction"
    Principal: "events.amazonaws.com"
    SourceArn:
      Fn::GetAtt:
        - "EventRuleCase1"
        - "Arn"
```

Génération d'un modèle AWS CloudFormation à partir de règles Amazon EventBridge

AWS CloudFormation vous permet de configurer et de gérer vos ressources AWS sur l'ensemble des comptes et des régions de manière centralisée et reproductible en traitant l'infrastructure comme du code. Pour ce faire, CloudFormation vous permet de créer des modèles qui définissent les ressources que vous souhaitez provisionner et gérer.

EventBridge vous permet de générer des modèles à partir des règles existantes dans votre compte, afin de vous aider à démarrer rapidement le développement de modèles CloudFormation. Vous pouvez sélectionner une seule règle ou plusieurs règles à inclure dans le modèle. Vous pouvez ensuite utiliser ces modèles comme base pour créer des piles de ressources sous la gestion de CloudFormation.

Pour plus d'informations sur CloudFormation, consultez le Guide de l'utilisateur AWS CloudFormation.

> **Note**
>
> EventBridge n'inclut pas de règles gérées dans le modèle généré.

Vous pouvez également générer un modèle à partir d'un bus d'événements existant, y compris les règles qu'il contient.

Pour générer un modèle AWS CloudFormation à partir d'une ou de plusieurs règles

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Rules.
3. Sous Sélectionnez un bus d'événements, choisissez le bus d'événements qui contient les règles que vous souhaitez inclure dans le modèle.
4. Sous Règles, choisissez les règles que vous souhaitez inclure dans le modèle AWS CloudFormation généré.

   Pour une règle unique, vous pouvez également choisir le nom de la règle pour afficher la page de détails de la règle.
5. Choisissez Modèle CloudFormation, puis choisissez le format dans lequel EventBridge doit générer le modèle : JSON ou YAML.

EventBridge affiche le modèle, généré dans le format sélectionné.

6. EventBridge vous donne la possibilité de télécharger le fichier modèle ou de copier le modèle dans le presse-papiers.

   - Pour télécharger le fichier modèle, choisissez Télécharger.
   - Pour copier le modèle dans le presse-papiers, choisissez Copier.

7. Pour quitter le modèle, choisissez Annuler.

Une fois que vous avez personnalisé votre modèle AWS CloudFormation en fonction de votre cas d'utilisation, vous pouvez l'utiliser pour créer des piles dans AWS CloudFormation.

Considérations lors de l’utilisation de modèles CloudFormation générés à partir d’Amazon EventBridge

Tenez compte des facteurs suivants lorsque vous utilisez un modèle CloudFormation que vous avez généré à partir d’EventBridge :

- EventBridge n'inclut pas de mots de passe dans le modèle généré.

  Vous pouvez modifier le modèle pour y inclure des paramètres de modèle qui permettent aux utilisateurs de spécifier des mots de passe ou d’autres informations sensibles lorsqu’ils utilisent le modèle pour créer ou mettre à jour une pile CloudFormation.

  En outre, les utilisateurs peuvent utiliser Secrets Manager pour créer un secret dans la région souhaitée, puis modifier le modèle généré pour utiliser des paramètres dynamiques.

- Les cibles du modèle généré restent exactement telles qu'elles ont été spécifiées dans le bus d'événements d'origine. Cela peut entraîner des problèmes entre régions si vous ne modifiez pas correctement le modèle avant de l'utiliser pour créer des piles dans d'autres régions.

  De plus, le modèle généré ne crée pas automatiquement les cibles en aval.

EventBridge Objectifs d'Amazon

Une cible est une ressource ou un point de terminaison qui EventBridge envoie un événement lorsque celui-ci correspond au modèle d'événement défini pour une règle. La règle traite les données de l'événement et envoie les informations pertinentes à la cible. Pour fournir des données d'événements à une cible, vous devez EventBridge disposer d'une autorisation pour accéder à la ressource cible. Vous pouvez définir jusqu'à cinq cibles pour chaque règle.

Lorsque vous ajoutez des cibles à une règle et que cette règle s'exécute peu de temps après, il se peut que les cibles nouvelles ou mises à jour ne soient pas invoquées immédiatement. Les modifications ne prennent pas effet instantanément.

La vidéo suivante explique les principes de base des cibles : Qu'est-ce qu'une cible ?

Cibles disponibles dans la EventBridge console

Vous pouvez configurer les cibles suivantes pour les événements dans la EventBridge console :

- Destination d'API
- API Gateway
- AWS AppSync;
- File d'attente des tâches de traitement par lot
- CloudWatch groupe de journaux
- CodeBuild projet
- CodePipeline
- Appel d'API `CreateSnapshot` Amazon EBS
- EC2 Image Builder
- Appel d'API EC2 `RebootInstances`
- Appel d'API EC2 `StopInstances`
- Appel d'API EC2 `TerminateInstances`
- Tâche ECS
- Bus d'événements dans un autre compte ou une autre région
- Bus d'événements dans le même compte et la même région

- Flux de diffusion Firehose
- Flux de travail Glue
- Plan de réponse Incident Manager
- Modèle d'évaluation Inspector
- Flux Kinesis
- Fonction Lambda (ASYNC)
- Requêtes d’API relatives aux données du cluster Amazon Redshift
- Requêtes d’API relatives aux données du groupe de travail Amazon Redshift sans serveur
- SageMaker Pipeline
- Rubrique Amazon SNS

  EventBridge ne prend pas en charge les rubriques Amazon SNS FIFO (premier entré, premier sorti).
- File d’attente Amazon SQS
- Machine d’état Step Functions (ASYNC)
- Systems Manager Automation
- Systems Manager OpsItem
- Run Command de Systems Manager

Paramètres de cible

Certaines cibles n'envoient pas les informations contenues dans la charge utile de l'événement à la cible, mais traitent l'événement comme un déclencheur pour appeler une API spécifique. EventBridge utilise les paramètres de la cible pour déterminer ce qui se passe avec cette cible. Tel est le cas des éléments suivants :

- Destinations d’API (Les données envoyées à une destination d’API doivent correspondre à la structure de l’API. Vous devez utiliser l’objet `InputTransformer` pour vous assurer que les données sont correctement structurées. Si vous souhaitez inclure la charge utile de l’événement d’origine, référencez-la dans `InputTransformer`.)
- API Gateway (Les données envoyées à API Gateway doivent correspondre à la structure de l’API. Vous devez utiliser l’objet `InputTransformer` pour vous assurer que les données sont correctement structurées. Si vous souhaitez inclure la charge utile de l’événement d’origine, référencez-la dans `InputTransformer`.)

- Amazon EC2 Image Builder
- `RedshiftDataParameters` (clusters d'API de données Amazon Redshift)
- `SageMakerPipelineParameters`(Pipelines de création de modèles Amazon SageMaker Runtime)

> **Note**
>
> EventBridge ne prend pas en charge toutes les syntaxes JSON Path et ne l'évalue pas lors de l'exécution. La syntaxe prise en charge inclut :
>
> - notation par points (par exemple, `$.detail`)
> - tirets
> - traits de soulignement
> - caractères alphanumériques
> - index de tableau
> - caractères génériques (*)

Paramètres de chemin dynamiques

Certains paramètres de cible prennent en charge la syntaxe de chemin JSON dynamique facultative. Cette syntaxe vous permet de spécifier des chemins JSON au lieu de valeurs statiques (par exemple, `$.detail.state`). La valeur entière doit être un chemin JSON, pas seulement une partie de celui-ci. Par exemple, `RedshiftParameters.Sql` peut avoir la valeur `$.detail.state`, mais pas la valeur `"SELECT * FROM $.detail.state"`. Ces chemins sont remplacés de manière dynamique lors de l'exécution par des données provenant de la charge utile de l'événement elle-même au niveau du chemin spécifié. Les paramètres de chemin dynamiques ne peuvent pas faire référence à des valeurs nouvelles ou transformées résultant d'une transformation d'entrée. La syntaxe prise en charge pour les chemins JSON de paramètres dynamiques est la même que lors de la transformation d'une entrée. Pour plus d'informations, consultez ???.

La syntaxe dynamique peut être utilisée sur tous les champs de type chaîne non enum de ces paramètres :

- `EcsParameters`
- `HttpParameters` (à l'exception des clés `HeaderParameters`)

- `RedshiftDataParameters`
- `SageMakerPipelineParameters`

Autorisations

Pour effectuer des appels d'API sur les ressources que vous possédez, vous devez EventBridge disposer des autorisations appropriées. Pour AWS Lambda et les ressources Amazon SNS, EventBridge utilise des politiques basées sur les ressources. Pour les instances EC2, les flux de données Kinesis et les machines d'état Step Functions EventBridge , utilise les rôles IAM que vous spécifiez dans le paramètre dans`RoleARN`. `PutTargets` Vous pouvez invoquer un point de terminaison API Gateway avec une autorisation IAM configurée, mais le rôle est facultatif si vous n'avez pas configuré d'autorisation. Pour plus d'informations, consultez Amazon EventBridge et AWS Identity and Access Management.

Si un autre compte se trouve dans la même région et vous a accordé l'autorisation, vous pouvez envoyer des événements à ce compte. Pour plus d'informations, consultez Envoyer et recevoir des EventBridge événements Amazon entre AWS comptes.

Si votre cible est chiffrée, vous devez inclure la section suivante dans votre stratégie de clé KMS.

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

EventBridge spécificités de la cible

AWS Batch files d'attente pour les emplois

Certains paramètres AWS Batch `submitJob` peuvent être configurés via BatchParameters.

Les autres paramètres peuvent être spécifiées dans la charge utile de l'événement. Si la charge utile de l'événement (transmise par ou via InputTransformers) contient les clés suivantes, elles sont mappées aux paramètres de `submitJob` demande :

- `ContainerOverrides: containerOverrides`

  > **Note**
  >
  > Inclut uniquement command, environment, memory et vcpus

- `DependsOn: dependsOn`

  > **Note**
  >
  > Inclut uniquement jobId

- `Parameters: parameters`

CloudWatch Groupe de journaux

Si vous n'utilisez pas un InputTransformeravec une cible CloudWatch Logs, la charge utile de l'événement est utilisée comme message de journal et la source de l'événement comme horodatage. Si vous utilisez un InputTransformer, le modèle doit être :

```
{"timestamp":<timestamp>,"message":<message>}
```

EventBridge regroupe les entrées envoyées à un flux de journal ; par conséquent, il EventBridge peut transmettre un ou plusieurs événements à un flux de journal, en fonction du trafic.

CodeBuild projet

Si vous façonnez l'événement d'entrée en une cible pour qu'elle corresponde à la CodeBuild StartBuildRequeststructure, les paramètres seront mappés 1 à 1 et transmis à. InputTransformers`codeBuild.StartBuild`

Tâches Amazon ECS

Si vous façonnez l'événement d'entrée sur une cible afin qu'elle corresponde à la RunTask TaskOverridestructure Amazon ECS, les paramètres seront mappés 1 à 1 et transmis à. InputTransformers`ecs.RunTask`

Plan de réponse Incident Manager

Si l'événement correspondant provient d' CloudWatch alarmes, les détails du changement d'état de l'alarme sont renseignés dans les détails du déclencheur de l' StartIncidentRequest appel à Incident Manager.

Configuration de cibles

Découvrez comment configurer les paramètres des EventBridge cibles.

Cibles :

- Destinations d’API
- EventBridge Objectifs Amazon pour Amazon API Gateway
- AWS AppSync cibles pour Amazon EventBridge
- Connexions pour les cibles de point de terminaison HTTP
- Envoyer et recevoir des EventBridge événements Amazon entre AWS comptes
- Envoyer et recevoir des EventBridge événements Amazon entre les AWS régions
- Envoyer et recevoir des EventBridge événements Amazon entre les bus d'événements d'un même compte et d'une même région

Destinations d’API

Les destinations EventBridge d'API Amazon sont des points de terminaison HTTP que vous pouvez invoquer en tant que cible d'une règle, de la même manière que vous invoquez un AWS service ou une ressource en tant que cible. À l'aide des destinations d'API, vous pouvez acheminer les événements entre les AWS services, les applications intégrées de type logiciel en tant que service (SaaS) et vos applications en dehors de l'extérieur en AWS utilisant des appels d'API. Lorsque vous spécifiez une destination d'API comme cible d'une règle, EventBridge invoque le point de terminaison HTTP pour tout événement correspondant au modèle d'événement spécifié dans la règle, puis fournit les informations relatives à l'événement avec la demande. Avec EventBridge, vous pouvez utiliser n'importe quelle méthode HTTP sauf CONNECT et TRACE pour la requête. Les méthodes HTTP les plus couramment utilisées sont PUT et POST. Vous pouvez également utiliser des transformateurs d’entrée pour personnaliser l’événement en fonction des paramètres d’un point de terminaison HTTP spécifique. Pour plus d’informations, consultez Transformation des EventBridge entrées Amazon.

> **Note**
>
> Les destinations d'API ne prennent pas en charge les destinations privées, telles que les points de terminaison VPC d'interface, y compris les API HTTPS privées dans des clouds privés virtuels (VPC) utilisant un réseau privé et un Application Load Balancer et des points de terminaison VPC d'interface.
> Pour plus d’informations, consultez ???.

> **Important**
>
> EventBridge les demandes adressées à un point de terminaison de destination de l'API doivent avoir un délai d'exécution maximal du client de 5 secondes. Si le point de terminaison cible met plus de 5 secondes à répondre, EventBridge la demande expire. EventBridge les demandes ont dépassé le délai maximum défini dans votre politique de nouvelles tentatives. Par défaut, ces valeurs maximales sont de 24 heures et 185 fois. Une fois le nombre maximal de nouvelles tentatives atteint, les événements sont envoyés à votre file d’attente de lettres mortes si vous en avez une. Sinon, l’événement est supprimé.

La vidéo suivante montre l’utilisation de la destination d’API : Utilisation de destinations d’API

Dans cette rubrique :

Création d’une destination d’API

Chaque destination d’API requiert une connexion. Une connexion spécifie le type d’autorisation et les informations d’identification à utiliser pour être autorisée auprès du point de terminaison de destination d’API. Vous pouvez choisir une connexion existante ou créer une connexion en même temps que vous créez la destination d’API. Pour plus d’informations, consultez ???.

Pour créer une destination d'API à l'aide de la EventBridge console

1. Connectez-vous à AWS l'aide d'un compte autorisé à gérer EventBridge et à ouvrir la EventBridgeconsole.
2. Dans le volet de navigation de gauche, choisissez Destinations d’API.
3. Faites défiler la page jusqu’au tableau Destinations d’API, puis choisissez Créer une destination d’API.
4. Sur la page Créer une destination d’API, entrez un Nom pour la destination d’API. Vous pouvez utiliser jusqu’à 64 lettres majuscules ou minuscules, des chiffres, des points (.), des tirets (-) ou des traits de soulignement (_).

   Le nom doit être unique pour votre compte dans la région actuelle.
5. Entrez une Description pour la destination d’API.
6. Entrez un Point de terminaison de la destination d’API pour la destination d’API. Le point de terminaison de la destination d’API est une cible de point de terminaison d’invocation HTTP pour les événements. Les informations d’autorisation que vous incluez dans la connexion utilisée pour cette destination d’API sont utilisées pour autoriser la connexion auprès de ce point de terminaison. L’URL doit utiliser HTTPS.

7. Entrez la Méthode HTTP à utiliser pour se connecter au Point de terminaison de la destination d'API.
8. (Facultatif) Dans le champ Limite du taux d'appel par seconde, entrez le nombre maximal d'invocations par seconde à envoyer au point de terminaison de destination d'API.

   La limite de débit que vous définissez peut affecter le mode de EventBridge diffusion des événements. Pour plus d'informations, consultez Comment le taux d'invocation affecte la livraison d'événements.
9. Pour Connexion, effectuez l'une des actions suivantes :
   - Choisissez Utiliser une connexion existante, puis sélectionnez la connexion à utiliser pour cette destination d'API.
   - Choisissez Créer une nouvelle connexion, puis entrez les détails de la connexion à créer. Pour plus d'informations, consultez Connexions.
10. Choisissez Créer.

Création de règles qui envoient des événements vers une destination d'API

Après avoir créé une destination d'API, vous pouvez la sélectionner en tant que cible d'une règle. Pour utiliser une destination d'API en tant que cible, vous devez fournir un rôle IAM avec les autorisations appropriées. Pour plus d'informations, consultez ???.

La sélection d'une destination d'API en tant que cible fait partie de la création de la règle.

Pour créer une règle qui envoie des événements à une destination d'API à l'aide de la console

1. Suivez les étapes de la procédure ???.
2. Au cours de l'???étape, lorsque vous êtes invité à choisir une destination d'API comme type de cible :

   a. Sélectionnez la destination de EventBridge l'API.

   b. Effectuez l'une des actions suivantes :

      - Choisissez Utiliser une destination d'API existante et sélectionnez une destination d'API existante
      - Choisissez Créer une nouvelle destination d'API et spécifiez le paramètre nécessaire pour définir votre nouvelle destination d'API.

Pour plus d'informations sur la définition des paramètres requis, consultez???.

c. (Facultatif) : Pour spécifier les paramètres d'en-tête de l'événement, sous Paramètres d'en-tête, choisissez Ajouter un paramètre d'en-tête.

   Spécifiez ensuite la clé et la valeur du paramètre d'en-tête.

d. (Facultatif) : Pour spécifier les paramètres de chaîne de requête pour l'événement, sous Paramètres de chaîne de requête, choisissez Ajouter un paramètre de chaîne de requête.

   Spécifiez ensuite la clé et la valeur du paramètre de chaîne de requête.

3. Terminez la création de la règle en suivant les étapes de la procédure.

Rôle lié à un service pour les destinations d’API

Lorsque vous créez une connexion pour une destination d'API, un rôle lié à un service nommé AWS ServiceRoleForAmazonEventBridgeApiDestinationsest ajouté à votre compte. EventBridge utilise le rôle lié au service pour créer et stocker un secret dans Secrets Manager. Pour accorder les autorisations nécessaires au rôle lié au service, associez EventBridge la AmazonEventBridgeApiDestinationsServiceRolePolicypolitique au rôle. La politique limite les autorisations accordées aux autorisations nécessaires pour que le rôle interagisse avec le secret de la connexion. Aucune autre autorisation n’est incluse et le rôle ne peut interagir qu’avec les connexions de votre compte pour gérer le secret.

La politique suivante est `AmazonEventBridgeApiDestinationsServiceRolePolicy`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DescribeSecret",
                "secretsmanager:DeleteSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:events!connection/*"
        }
```

```
    ]
}
```

Pour plus d'informations sur les rôles lié à un service, consultez Utilisation des rôles liés à un service dans la documentation d'IAM.

Le rôle `AmazonEventBridgeApiDestinationsServiceRolePolicy` lié à un service est pris en charge dans les régions suivantes : AWS

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Californie du Nord)
- USA Ouest (Oregon)
- Afrique (Le Cap)
- Asie-Pacifique (Hong Kong)
- Asia Pacific (Mumbai)
- Asie-Pacifique (Osaka)
- Asia Pacific (Seoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Tokyo)
- Canada (Centre)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Amérique du Sud (São Paulo)
- Chine (Ningxia)
- Chine (Beijing)

En-têtes dans les demandes adressées aux destinations d’API

La section suivante explique comment EventBridge gérer les en-têtes HTTP dans les demandes adressées aux destinations d'API.

En-têtes inclus dans les demandes adressées aux destinations d’API

Outre les en-têtes d'autorisation définis pour la connexion utilisée pour une destination d'API, EventBridge inclut les en-têtes suivants dans chaque demande.

Clé d’en-tête	Valeur d'en-tête
Agent utilisateur	Amazon//EventBridgeApiDestinations
Content-Type	Si aucune valeur de type de contenu personnalisée n'est spécifiée, EventBridge inclut la valeur par défaut suivante en tant que type de contenu : application/json; charset=utf-8
Range	bytes=0-1048575
Accept-Encoding	gzip,deflate
Connexion	close
Content-Length	En-tête d’entité qui indique la taille de entity-body, en octets, envoyé au destinataire.
Host (Hôte)	En-tête de demande qui indique l’hôte et le numéro de port du serveur sur lequel la demande est envoyée.

En-têtes qui ne peuvent pas être remplacés dans les demandes adressées aux destinations d’API

EventBridge ne vous permet pas de remplacer les en-têtes suivants :

- Agent utilisateur
- Range

Les en-têtes sont EventBridge supprimés des demandes destinées aux destinations de l'API

EventBridge supprime les en-têtes suivants pour toutes les demandes de destination d'API :

- A-IM
- Accept-Charset
- Accept-Datetime
- Accept-Encoding
- Cache-Control
- Connexion
- Encodage-Contenu
- Content-Length
- Content-MD5
- Date
- Expect
- Forwarded
- De
- Host (Hôte)
- HTTP2-Settings
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Origin
- Pragma
- Proxy-Authorization
- Range
- Référent
- TE
- Trailer

- Transfer-Encoding
- Agent utilisateur
- Upgrade
- Via
- Avertissement

Codes d’erreur de destination d’API

Lorsque EventBridge vous essayez de transmettre un événement à une destination d'API et qu'une erreur se produit EventBridge, procédez comme suit :

- Il tente à nouveau de livrer les événements associés aux codes d’erreur 409, 429 et 5xx.
- Il ne tente pas à nouveau de livrer les événements associés aux codes d’erreur 1xx, 2xx, 3xx et 4xx (sauf 429).

EventBridge Les destinations d'API lisent l'en-tête de réponse HTTP standard `Retry-After` pour savoir combien de temps il faut attendre avant de faire une demande de suivi. EventBridge choisit la valeur la plus prudente entre la politique de réessai définie et l'`Retry-After`en-tête. Si `Retry-After` la valeur est négative, EventBridge arrête toute nouvelle tentative de livraison pour cet événement.

Comment le taux d’invocation affecte la livraison d’événements

Si vous définissez le taux d’invocation par seconde sur une valeur bien inférieure au nombre d’invocations générées, les événements peuvent ne pas être livrés dans le délai de 24 heures imparti pour les nouvelles tentatives. Par exemple, si vous définissez le taux d’invocation sur 10 invocations par seconde, mais que des milliers d’événements sont générés par seconde, votre retard au niveau des événements à livrer dépassera rapidement 24 heures. Pour vous assurer qu’aucun événement n’est perdu, configurez une file d’attente de lettres mortes à laquelle envoyer les événements dont les invocations ont échoué afin de pouvoir les traiter ultérieurement. Pour plus d’informations, consultez Utilisation de files d'attente de lettres mortes pour traiter les événements non livrés.

Envoi d' CloudEvents événements vers des destinations d'API

CloudEvents est une spécification indépendante du fournisseur pour le formatage des événements, dans le but d'assurer l'interopérabilité entre les services, les plateformes et les systèmes. Vous

pouvez l'utiliser EventBridge pour transformer les événements de AWS service CloudEvents avant qu'ils ne soient envoyés à une cible, telle qu'une destination d'API.

> **Note**
>
> La procédure suivante explique comment transformer les événements source en mode structuré. CloudEvents Dans la CloudEvents spécification, un message en mode structuré est un message dans lequel l'ensemble de l'événement (attributs et données) est codé dans la charge utile de l'événement.

Pour plus d'informations sur les CloudEvents spécifications, consultez cloudevents.io.

Pour transformer les AWS événements au CloudEvents format souhaité à l'aide de la console

Pour transformer les événements au CloudEvents format avant leur diffusion à une cible, vous devez commencer par créer une règle de bus d'événements. Dans le cadre de la définition de la règle, vous utilisez un transformateur d'entrée pour obtenir des événements de EventBridge transformation avant de les envoyer à la cible que vous spécifiez.

1. Suivez les étapes de la procédure ???.
2. Au cours de l'???étape, lorsque vous êtes invité à choisir une destination d'API comme type de cible :
   a. Sélectionnez la destination de EventBridge l'API.
   b. Effectuez l'une des actions suivantes :
      - Choisissez Utiliser une destination d'API existante et sélectionnez une destination d'API existante
      - Choisissez Créer une nouvelle destination d'API et spécifiez le paramètre nécessaire pour définir votre nouvelle destination d'API.

        Pour plus d'informations sur la définition des paramètres requis, consultez???.
   c. Spécifiez les paramètres d'en-tête Content-Type nécessaires pour les CloudEvents événements :
      - Sous Paramètres d'en-tête, choisissez Ajouter un paramètre d'en-tête.
      - Pour la clé, spécifiez`Content-Type`.

Pour la valeur, spécifiez`application/cloudevents+json; charset=UTF-8`.

3. Spécifiez un rôle d'exécution pour votre cible.
4. Définissez un transformateur d'entrée pour transformer les données de l'événement source au CloudEvents format suivant :

   a. Sous Paramètres supplémentaires, pour Configurer l'entrée cible, choisissez Transformateur d'entrée.

      Choisissez ensuite Configurer le transformateur d'entrée.

   b. Sous Transformateur d'entrée cible, spécifiez le chemin d'entrée.

      Dans le chemin d'entrée ci-dessous, l'attribut region est un attribut d'extension personnalisé du CloudEvents format. En tant que tel, il n'est pas nécessaire pour respecter les CloudEvents spécifications.

      CloudEvents vous permet d'utiliser et de créer des attributs d'extension non définis dans la spécification de base. Pour plus d'informations, y compris une liste des attributs d'extension connus, consultez la section Attributs d'CloudEvents extension dans la documentation de CloudEvents spécification sur GitHub.

      ```
      {
        "detail": "$.detail",
        "detail-type": "$.detail-type",
        "id": "$.id",
        "region": "$.region",
        "source": "$.source",
        "time": "$.time"
      }
      ```

   c. Dans Modèle, entrez le modèle pour transformer les données d'événement source au CloudEvents format.

      Dans le modèle ci-dessous, `region` ce n'est pas strictement obligatoire, car l'`region`attribut du chemin d'entrée est un attribut d'extension de la CloudEvents spécification.

      ```
      {
        "specversion":"1.0",
        "id":<id>,
      ```

```
  "source":<source>,
  "type":<detail-type>,
  "time":<time>,
  "region":<region>,
  "data":<detail>
}
```

5. Terminez la création de la règle en suivant les étapes de la procédure.

Partenaires de destination d’API

Utilisez les informations fournies par les AWS partenaires suivants pour configurer une destination d'API et une connexion pour leur service ou application.

Observabilité dans le cloud de Cisco

URL du point de terminaison d’invocation de destination d’API :

`https://tenantName.observe.appdynamics.com/rest/awsevents/aws-eventbridge-integration/endpoint`

Types d’autorisations pris en charge :

Informations d’identification client OAuth

Les jetons OAuth sont actualisés lorsqu'une réponse 401 ou 407 est renvoyée

Paramètres d’autorisation supplémentaires requis :

ID AppDynamics client Cisco et secret du client

Point de terminaison OAuth :

`https://tenantName.observe.appdynamics.com/auth/tenantId/default/oauth2/token`

Les paramètres de paire clé/valeur OAuth suivants :

Type	Clé	Valeur
Champ corporel	grant_type	client_credentials

Type	Clé	Valeur
En-tête	Content-Type	application/ x-www-form-urlencoded ; jeu de caractères = utf-8

AppDynamics Documentation Cisco :

AWS ingestion d'événements

Opérations d’API couramment utilisées :

Ne s’applique pas

Informations supplémentaires :

En choisissant Cisco AppDynamics dans le menu déroulant Destinations des partenaires, les informations OAuth nécessaires sont préremplies, notamment les paires clé/valeur d'en-tête et de corps requises pour les appels d'API.

Pour plus d'informations, voir ingestion d'AWS événements dans la AppDynamics documentation Cisco.

Confluent

URL du point de terminaison d’invocation de destination d’API :

Généralement, le format suivant :

```
https://random-id.region.aws.confluent.cloud:443/kafka/v3/
clusters/cluster-id/topics/topic-name/records
```

Pour plus d'informations, consultez Rechercher l'adresse du point de terminaison REST et l'ID du cluster dans la documentation Confluent.

Types d’autorisations pris en charge :

Base

Paramètres d’autorisation supplémentaires requis :

Ne s’applique pas

Documentation Confluent :

Produire des disques

Proxy REST Confluent pour Apache Kafka

Opérations d'API couramment utilisées :

`POST`

Informations supplémentaires :

Pour transformer les données d'événement en un message que le point de terminaison peut traiter, créez un transformateur d'entrée cible.

- Pour générer un enregistrement sans spécifier de clé de partitionnement Kafka, utilisez le modèle suivant pour votre transformateur d'entrée. Aucun chemin d'entrée n'est requis.

```
{
  "value":{
      "type":"JSON",
      "data":aws.events.event.json
  },
}
```

- Pour générer un enregistrement en utilisant un champ de données d'événement comme clé de partitionnement Kafka, suivez le chemin d'entrée et l'exemple de modèle ci-dessous. Cet exemple définit le chemin d'entrée du `orderId` champ, puis indique ce champ comme clé de partition.

  Définissez d'abord le chemin d'entrée pour le champ de données d'événement :

```
{
  "orderId":"$.detail.orderId"
}
```

  Utilisez ensuite le modèle de transformateur d'entrée pour spécifier le champ de données comme clé de partition :

```
{
  "value":{
    "type":"JSON",
    "data":aws.events.event.json
```

```
  },
  "key":{
    "data":"<orderId>",
    "type":"STRING"
  }
}
```

Coralogix

URL du point de terminaison d'invocation de destination d'API

Pour obtenir la liste complète des points de terminaison, consultez la Référence des API Coralogix.

Types d'autorisations pris en charge

Clé d'API

Paramètres d'autorisation supplémentaires requis

En-tête `"x-amz-event-bridge-access-key"`, la valeur est la clé d'API Coralogix

Documentation Coralogix

EventBridgeAuthentification Amazon

Opérations d'API couramment utilisées

États-Unis : https://ingress.coralogix.us/aws/event-bridge

Singapour : https://ingress.coralogixsg.com/aws/event-bridge

Irlande : https://ingress.coralogix.com/aws/event-bridge

Stockholm : https://ingress.eu2.coralogix.com/aws/event-bridge

Inde : https://ingress.coralogix.in/aws/event-bridge

Informations supplémentaires

Les événements sont stockés sous forme d'entrées de journal avec `applicationName=[AWS Account]` et `subsystemName=[event.source]`.

Datadog

URL du point de terminaison d’invocation de destination d’API

Pour obtenir la liste complète des points de terminaison, consultez la Référence des API Datadog.

Types d’autorisations pris en charge

Clé d'API

Paramètres d’autorisation supplémentaires requis

Aucun

Documentation Datadog

Authentification

Opérations d’API couramment utilisées

POST https://api.datadoghq.com/api/v1/events

POST https://http-intake.logs.datadoghq.com/v1/input

Informations supplémentaires

Les URL de point de terminaison varient en fonction de l’emplacement de votre organisation Datadog. Pour connaître l’URL appropriée à votre organisation, consultez la documentation.

Freshworks

URL du point de terminaison d’invocation de destination d’API

Pour obtenir la liste des points de terminaison, consultez https://developers.freshworks.com/documentation/

Types d’autorisations pris en charge

Basique, clé d’API

Paramètres d’autorisation supplémentaires requis

Ne s’applique pas

Documentation Freshworks

Authentification

Opérations d’API couramment utilisées

https://developers.freshdesk.com/api/#create_ticket

https://developers.freshdesk.com/api/#update_ticket

https://developer.freshsales.io/api/#create_lead

https://developer.freshsales.io/api/#update_lead

Informations supplémentaires

Aucun

MongoDB

URL du point de terminaison d’invocation de destination d’API

https://data.mongodb-api.com/app/*ID_application*/endpoint/

Types d’autorisations pris en charge

Clé d'API

E-mail/mot de passe

Authentification JWT personnalisée

Paramètres d’autorisation supplémentaires requis

Aucun

Documentation MongoDB

API de données Atlas (langue française non garantie)

Points de terminaison

Points de terminaison HTTPS personnalisés (langue française non garantie)

Authentification

Opérations d’API couramment utilisées

Aucun

Informations supplémentaires

Aucun

New Relic

URL du point de terminaison d’invocation de destination d’API

Pour plus d’informations, consultez Nos centres de données pour les régions EU et US (langue française non garantie).

Evénements

US : https://insights-collector.newrelic.com/v1/accounts/*VOTRE_ID_COMPTE_NEW_RELIC*/events

EU : https://insights-collector.eu01.nr-data.net/v1/accounts/*VOTRE_ID_COMPTE_NEW_RELIC*/events

Métriques

US : https://metric-api.newrelic.com/metric/v1

EU : https://metric-api.eu.newrelic.com/metric/v1

Journaux

US : https://log-api.newrelic.com/log/v1

EU : https://log-api.eu.newrelic.com/log/v1

Suivis

US : https://trace-api.newrelic.com/trace/v1

EU : https://trace-api.eu.newrelic.com/trace/v1

Types d’autorisations pris en charge

Clé d'API

Documentation New Relic

API de métrique (langue française non garantie)

API d’événement (langue française non garantie)

API de journal (langue française non garantie)

API de suivi (langue française non garantie)

Opérations d’API couramment utilisées

API de métrique (langue française non garantie)

API d’événement (langue française non garantie)

API de journal (langue française non garantie)

API de suivi (langue française non garantie)

Informations supplémentaires

Limites de l’API de métrique (langue française non garantie)

Limites de l’API d’événement (langue française non garantie)

Limites de l’API de journal (langue française non garantie)

Limites de l’API de suivi (langue française non garantie)

Operata

URL du point de terminaison d’invocation de destination d’API :

https://api.operata.io/v2/aws/events/contact-record

Types d’autorisations pris en charge :

Base

Paramètres d’autorisation supplémentaires requis :

Aucun

Documentation d’Operata :

Comment créer, afficher, modifier et révoquer des jetons d’API ? (langue française non garantie)

AWS Intégration d'Operata à l'aide d'Amazon EventBridge Scheduler Pipes

Opérations d'API couramment utilisées :

```
POST https://api.operata.io/v2/aws/events/contact-record
```

Informations supplémentaires :

L'élément `username` correspond à l'ID du groupe Operata et le mot de passe est votre jeton d'API.

Salesforce

URL du point de terminaison d'invocation de destination d'API

Objet : myDomainNamehttps://.my.salesforce.com/services/data/VersionNumber /sobjects//* SobjectEndpoint

Événements de plateforme personnalisés *: https ://myDomainName.my.salesforce.com/services/data/VersionNumber / sobjects/ /* customPlatformEndpoint*

Pour obtenir la liste complète des points de terminaison, consultez la Référence des API Salesforce (langue française non garantie).

Types d'autorisations pris en charge

Informations d'identification client OAuth

Les jetons OAUTH sont actualisés lorsqu'une réponse 401 ou 407 est renvoyée.

Paramètres d'autorisation supplémentaires requis

ID client et secret client de l'application connectée Salesforce.

L'un des points de terminaison d'autorisation suivants :

- Production : https ://*MyDomainName*.my.salesforce.com. /services/oauth2/jeton
- Sandbox sans domaines améliorés : https ://*MyDomainName*-- *SandboxName*.my. salesforce.com/services /oauth2/token
- Sandbox avec domaines améliorés : https ://***MyDomainName***-- ***SandboxName***.sandbox.my.salesforce.com/services/oauth2/token

La paire clé/valeur suivante :

Clé	Valeur
grant_type	client_credentials

Documentation Salesforce

Guide du développeur d’API REST (langue française non garantie)

Opérations d’API couramment utilisées

Utilisation des métadonnées d’objet (langue française non garantie)

Utilisation des enregistrements

Informations supplémentaires

Pour un didacticiel expliquant comment utiliser la EventBridge console pour créer une connexionSalesforce, une destination d'API et une règle vers laquelle acheminer les informationsSalesforce, consultez???.

Slack

URL du point de terminaison d’invocation de destination d’API

Pour obtenir la liste des points de terminaison et d’autres ressources, consultez Utilisation de l’API web Slack (langue française non garantie).

Types d’autorisations pris en charge

OAuth 2.0

Les jetons OAUTH sont actualisés lorsqu'une réponse 401 ou 407 est renvoyée.

Lorsque vous créez une application Slack et que vous l’installez dans votre espace de travail, un jeton porteur OAuth est créé en votre nom pour authentifier les appels par votre connexion de destination d’API.

Paramètres d’autorisation supplémentaires requis

Ne s’applique pas

Documentation Slack

Configuration de l’application de base (langue française non garantie)

Installation avec OAuth (langue française non garantie)

Extraction des messages (langue française non garantie)

Envoi de messages

Envoi de messages à l’aide de webhooks entrants (langue française non garantie)

Opérations d’API couramment utilisées

https://slack.com/api/chat.postMessage

Informations supplémentaires

Lorsque vous configurez votre EventBridge règle, vous devez mettre en évidence deux configurations :

- Incluez un paramètre d’en-tête qui définit le type de contenu en tant que « application/json;charset=utf-8 ».
- Utilisez un transformateur d’entrée pour mapper l’événement d’entrée à la sortie attendue pour l’API Slack, c’est-à-dire pour vous assurer que la charge utile envoyée à l’API Slack comporte les paires de clé/valeur « channel » et « text ».

Shopify

URL du point de terminaison d’invocation de destination d’API

Pour obtenir la liste des points de terminaison et d’autres ressources et méthodes, consultez Points de terminaison et demandes (langue française non garantie).

Types d’autorisations pris en charge

OAuth, clé d’API

> **Note**
>
> Les jetons OAUTH sont actualisés lorsqu'une réponse 401 ou 407 est renvoyée.

Paramètres d’autorisation supplémentaires requis

Ne s’applique pas

Documentation Shopify

Présentation de l’authentification et de l’autorisation (langue française non garantie)

Opérations d’API couramment utilisées

POST - /admin/api/2022-01/products.json

GET - admin/api/2022-01/products/{product_id}.json

PUT - admin/api/2022-01/products/{product_id}.json

DELETE - admin/api/2022-01/products/{product_id}.json

Informations supplémentaires

Création d’une application (langue française non garantie)

Livraison de EventBridge webhooks Amazon

Jetons d’accès pour les applications personnalisées dans l’admin Shopify (langue française non garantie)

Produit (langue française non garantie)

API d’admin Shopify (langue française non garantie)

Splunk

URL du point de terminaison d’invocation de destination d’API

https://*POINT_DE_TERMINAISON_HEC_SPLUNK*:*port_facultatif*/services/collector/raw

Types d’autorisations pris en charge

Basique, clé d’API

Paramètres d’autorisation supplémentaires requis

Aucun

Documentation Splunk

Pour les deux types d’autorisations, vous avez besoin d’un ID de jeton HEC. Pour plus d’informations, consultez Configuration et utilisation du collecteur d’événements HTTP dans Splunk Web (langue française non garantie).

Opérations d’API couramment utilisées

POST https://*POINT_DE_TERMINAISON_HEC_SPLUNK*:*port_facultatif*/services/collector/raw

Informations supplémentaires

Clé d'API — Lors de la configuration du point de terminaison pour EventBridge, le nom de la clé d'API est « Authorization » et la valeur est l'ID du jeton Splunk HEC.

Basique (nom d'utilisateur/mot de passe) — Lors de la configuration du point de terminaison pour EventBridge, le nom d'utilisateur est « Splunk » et le mot de passe est l'identifiant du jeton Splunk HEC.

Sumo Logic

URL du point de terminaison d’invocation de destination d’API

Les URL des points de terminaison sources de métrique et de journal HTTP seront différentes pour chaque utilisateur. Pour plus d’informations, consultez Source de métriques et de journaux HTTP (langue française non garantie).

Types d’autorisations pris en charge

Sumo Logic ne requiert pas d’authentification sur ses sources HTTP, car une clé unique est précalculée dans l’URL. Pour cette raison, vous devez veiller à traiter cette URL comme un secret.

Lorsque vous configurez la destination de l' EventBridge API, un type d'autorisation est requis. Pour répondre à cette exigence, sélectionnez Clé d’API et donnez-lui le nom de clé « dummy-key » et la valeur de clé « dummy-value ».

Paramètres d’autorisation supplémentaires requis

Ne s’applique pas

Documentation Sumo Logic

Sumo Logica déjà créé des sources hébergées pour collecter les journaux et les métriques de nombreux AWS services et vous pouvez utiliser les informations de son site Web pour travailler avec ces sources. Pour plus d’informations, consultez Amazon Web Services.

Si vous générez des événements personnalisés à partir d'une application et que vous souhaitez les envoyer Sumo Logic sous forme de journaux ou de métriques, utilisez les destinations d' EventBridge API et les points de terminaison Sumo Logic HTTP Log and Metric Source.

- Pour vous inscrire et créer une instance Sumo Logic gratuite, consultez Commencer votre essai gratuit dès aujourd'hui (langue française non garantie).
- Pour plus d'informations sur l'utilisation de Sumo Logic, consultez Source de métriques et de journaux HTTP (langue française non garantie).

Opérations d'API couramment utilisées

POST https://endpoint4.collection.us2.sumologic.com/receiver/v1/http/*ID_UNIQUE_PAR_COLLECTEUR*

Informations supplémentaires

Aucun

TriggerMesh

URL du point de terminaison d'invocation de destination d'API

Utilisez les informations de la rubrique Source d'événement pour HTTP (langue française non garantie) pour formuler l'URL du point de terminaison. L'URL d'un point de terminaison inclut le nom de la source d'événement et l'espace de noms utilisateur au format suivant :

https://*nom-source.espaces-noms-utilisateur*.cloud.triggermesh.io

Inclut les paramètres d'autorisation Basique dans la demande au point de terminaison.

Types d'autorisations pris en charge

Base

Paramètres d'autorisation supplémentaires requis

Aucun

Documentation TriggerMesh

Source d'événement pour HTTP (langue française non garantie)

Opérations d'API couramment utilisées

Ne s'applique pas

Informations supplémentaires

Aucun

Zendesk

URL du point de terminaison d’invocation de destination d’API

https://developer.zendesk.com/rest_api/docs/support/tickets

Types d’autorisations pris en charge

Basique, clé d’API

Paramètres d’autorisation supplémentaires requis

Aucun

Documentation Zendesk

Sécurité et authentification (langue française non garantie)

Opérations d’API couramment utilisées

POST https://*votre_sous-domaine_Zendesk*/api/v2/tickets

Informations supplémentaires

Les demandes d'API EventBridge sont prises en compte dans les limites de votre API Zendesk. Pour en savoir plus sur les limites Zendesk pour votre offre, consultez Limites d’utilisation (langue française non garantie).

Pour mieux protéger votre compte et vos données, nous vous recommandons d’utiliser une clé API plutôt que l’authentification de base avec vos informations d’identification de connexion.

EventBridge Objectifs Amazon pour Amazon API Gateway

Vous pouvez utiliser Amazon API Gateway pour créer, publier, gérer et surveiller des API. Amazon EventBridge prend en charge l'envoi d'événements vers un point de terminaison API Gateway. Lorsque vous spécifiez un point de terminaison API Gateway en tant que cible, chaque événement envoyé à la cible correspond à une demande envoyée au point de terminaison.

> **Important**
>
> EventBridge prend en charge l'utilisation de points de terminaison régionaux et optimisés pour API Gateway Edge comme cibles. Les points de terminaison privés ne sont actuellement

pas pris en charge. Pour en savoir plus sur les points de terminaison, consultez https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-api-endpoint-types.html.

Vous pouvez utiliser une cible API Gateway pour les cas d'utilisation suivants :

- Pour appeler une API spécifiée par le client et hébergée dans API Gateway en fonction d'événements tiers AWS ou en fonction d'événements tiers.
- Pour invoquer un point de terminaison périodiquement selon un calendrier.

Les informations d'événement EventBridge JSON sont envoyées dans le corps de la requête HTTP à votre point de terminaison. Vous pouvez spécifier les autres attributs de demande dans le champ `HttpParameters` de la cible comme suit :

- `PathParameterValues` répertorie les valeurs qui correspondent séquentiellement à toutes les variables de chemin dans l'ARN de votre point de terminaison, par exemple `"arn:aws:execute-api:us-east-1:112233445566:myapi/dev/POST/pets/*/*"`.
- `QueryStringParameters`représente les paramètres de la chaîne de requête qui s' EventBridge ajoutent au point de terminaison invoqué.
- `HeaderParameters` définit les en-têtes HTTP à ajouter à la demande.

> **Note**
>
> Pour des raisons de sécurité, les clés d'en-tête HTTP suivantes ne sont pas autorisées :
>
> - Toutes les clés ayant le préfixe `X-Amz` ou `X-Amzn`
> - `Authorization`
> - `Connection`
> - `Content-Encoding`
> - `Content-Length`
> - `Host`
> - `Max-Forwards`
> - `TE`
> - `Transfer-Encoding`

- `Trailer`
- `Upgrade`
- `Via`
- `WWW-Authenticate`
- `X-Forwarded-For`

Paramètres dynamiques

Lorsque vous invoquez une cible API Gateway, vous pouvez ajouter dynamiquement des données aux événements envoyés à la cible. Pour plus d'informations, consultez the section called "Paramètres de cible".

Nouvelles tentatives d'invocation

Comme pour toutes les cibles, EventBridge réessaie certains appels qui ont échoué. Pour API Gateway, EventBridge réessaie les réponses envoyées avec un code d'état HTTP 5xx ou 429 pendant 24 heures au maximum, avec un décalage et une instabilité exponentiels. Ensuite, EventBridge publie une `FailedInvocations` métrique sur Amazon CloudWatch. EventBridge ne réessaie pas d'autres erreurs HTTP 4xx.

Expiration

EventBridge règle Les demandes API Gateway doivent avoir un délai d'exécution maximal du client de 5 secondes. Si API Gateway met plus de 5 secondes à répondre, EventBridge expire la demande, puis réessaie.

EventBridge Les requêtes Pipes API Gateway ont un délai maximum de 29 secondes, le délai maximum pour l'API Gateway.

AWS AppSync cibles pour Amazon EventBridge

AWS AppSync permet aux développeurs de connecter leurs applications et services aux données et aux événements grâce à des API GraphQL et Pub/Sub sécurisées, sans serveur et hautement performantes. Avec AWS AppSync, vous pouvez publier des mises à jour de données en temps réel pour vos applications à l'aide de mutations GraphQL. EventBridge prend en charge l'appel d'une opération de mutation GraphQL valide pour les événements correspondants. Lorsque vous spécifiez

une mutation d' AWS AppSync API comme cible, AWS AppSync traite l'événement via une opération de mutation, qui peut ensuite déclencher des abonnements liés à la mutation.

> **Note**
>
> EventBridge prend en charge AWS AppSync les API GraphQL publiques. EventBridge ne prend actuellement pas en charge les API AWS AppSync privées.

Vous pouvez utiliser une cible d'API AWS AppSync GraphQL dans les cas d'utilisation suivants :

- Pour transmettre, transformer et stocker des données d'événements dans vos sources de données configurées.
- Pour envoyer des notifications en temps réel aux clients d'applications connectés.

> **Note**
>
> AWS AppSync les cibles prennent uniquement en charge l'appel des API AWS AppSync GraphQL à l'aide du type `AWS_IAM`d'autorisation.

Pour plus d'informations sur les API AWS AppSync GraphQL, consultez GraphQL et AWS AppSync son architecture dans le Guide du développeur.AWS AppSync

Pour spécifier une AWS AppSync cible pour une EventBridge règle à l'aide de la console

1. Créez ou modifiez la règle.
2. Sous Cible, spécifiez la cible en choisissant Service AWS , puis AWS AppSync.
3. Spécifiez l'opération de mutation à analyser et à exécuter, ainsi que le jeu de sélection.
   - Choisissez l' AWS AppSync API, puis la mutation de l'API GraphQL à invoquer.
   - Sous Configurer les paramètres et le jeu de sélection, choisissez de créer un jeu de sélection à l'aide d'un mappage clé-valeur ou d'un transformateur d'entrée.

     Key-value mapping

     Pour utiliser le mappage clé-valeur afin de créer votre jeu de sélection :

- Spécifiez des variables pour les paramètres de l’API. Chaque variable peut être une valeur statique ou une expression de chemin JSON dynamique vers la charge utile de l’événement.
- Sous Jeu de sélection, choisissez les variables que vous souhaitez inclure dans la réponse.

Input transformer

Pour utiliser un transformateur d’entrée afin de créer votre jeu de sélection :

- Spécifiez un chemin d’entrée qui définit les variables à utiliser.
- Spécifiez un modèle d’entrée pour définir et formater les informations que vous souhaitez transmettre à la cible.

Pour plus d’informations, consultez ???.

4. Pour Rôle d’exécution, choisissez de créer un nouveau rôle ou d’en utiliser un existant.
5. Terminez la création ou la modification de la règle.

Exemple : AWS AppSync cibles pour Amazon EventBridge

Dans l'exemple suivant, nous allons expliquer comment spécifier une AWS AppSync cible pour une EventBridge règle, notamment définir une transformation d'entrée pour formater les événements en vue de leur diffusion.

Supposons que vous disposiez d'une API AWS AppSync GraphQL définie par le schéma suivant : `Ec2EventAPI`

```
type Event {
    id: ID!
    statusCode: String
    instanceId: String
}

type Mutation {
    pushEvent(id: ID!, statusCode: String!, instanceId: String): Event
}

type Query {
    listEvents: [Event]
}
```

```
type Subscription {
    subscribeToEvent(id: ID, statusCode: String, instanceId: String): Event
        @aws_subscribe(mutations: ["pushEvent"])
}
```

Les clients des applications qui utilisent cette API peuvent souscrire à l'abonnement `subscribeToEvent`, qui est déclenché par la mutation `pushEvent`.

Vous pouvez créer une EventBridge règle avec une cible qui envoie des événements à l' AppSync API via la `pushEvent` mutation. Lorsque la mutation est invoquée, tout client abonné reçoit l'événement.

Pour spécifier cette API comme cible d'une EventBridge règle, procédez comme suit :

1. Définissez l'Amazon Resource Name (ARN) de la cible de règle sur l'ARN du point de terminaison GraphQL de l'API `Ec2EventAPI`.
2. Spécifiez l'opération GraphQL de mutation en tant que paramètre cible :

   ```
   mutation CreatePushEvent($id: ID!, $statusCode: String, $instanceId: String) {
     pushEvent(id: $input, statusCode: $statusCode, instanceId: $instanceId) {
       id
       statusCode
       instanceId
     }
   }
   ```

   Votre jeu de sélection de mutation doit inclure tous les champs auxquels vous souhaitez vous abonner dans votre abonnement GraphQL.
3. Configurez un transformateur d'entrée pour spécifier comment les données des événements correspondants sont utilisées dans votre opération.

   Supposons que vous ayez sélectionné l'exemple d'événement `"EC2 Instance Launch Successful"` :

   ```
   {
     "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": ["arn:aws:autoscaling:us-
east-1:123456789012:autoScalingGroup:eb56d16b-bbf0-401d-b893-
d5978ed4a025:autoScalingGroupName/sampleLuanchSucASG", "arn:aws:ec2:us-
east-1:123456789012:instance/i-b188560f"],
  "detail": {
    "StatusCode": "InProgress",
    "AutoScalingGroupName": "sampleLuanchSucASG",
    "ActivityId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
    "Details": {
      "Availability Zone": "us-east-1b",
      "Subnet ID": "subnet-95bfcebe"
    },
    "RequestId": "9cabb81f-42de-417d-8aa7-ce16bf026590",
    "EndTime": "2015-11-11T21:31:47.208Z",
    "EC2InstanceId": "i-b188560f",
    "StartTime": "2015-11-11T21:31:13.671Z",
    "Cause": "At 2015-11-11T21:31:10Z a user request created an AutoScalingGroup
 changing the desired capacity from 0 to 1.  At 2015-11-11T21:31:11Z an instance was
 started in response to a difference between desired and actual capacity, increasing
 the capacity from 0 to 1."
  }
}
```

Vous pouvez définir les variables suivantes à utiliser dans votre modèle, à l'aide du chemin d'entrée du transformateur d'entrée cible :

```
{
  "id": "$.id",
  "statusCode": "$.detail.StatusCode",
  "EC2InstanceId": "$.detail.EC2InstanceId"
}
```

Composez le modèle de transformateur d'entrée pour définir les variables qui EventBridge sont transmises à l'opération de AWS AppSync mutation. Le modèle doit être évalué au format JSON. Compte tenu de notre chemin d'entrée, vous pouvez composer le modèle suivant :

```
{
  "id": <id>,
  "statusCode": <statusCode>,
  "instanceId": <EC2InstanceId>
```

```
}
```

Connexions pour les cibles de point de terminaison HTTP

Une connexion définit la méthode d'autorisation et les informations d'identification EventBridge à utiliser pour se connecter à un point de terminaison HTTP donné. Lorsque vous configurez les paramètres d'autorisation et créez une connexion, un secret est créé AWS Secrets Manager pour stocker de manière sécurisée les informations d'autorisation. Vous pouvez également ajouter des paramètres supplémentaires à inclure dans la connexion en fonction de votre point de terminaison HTTP cible.

Utilisez des connexions avec :

- Destinations d’API

  Lorsque vous créez une destination d’API, vous spécifiez une connexion destinée à son utilisation. Vous pouvez choisir une connexion existante depuis votre compte ou créer une connexion lorsque vous créez une destination d'API.

Méthodes d'autorisation pour les connexions

EventBridge les connexions prennent en charge les méthodes d'autorisation suivantes :

- Base
- Clé d'API

  Pour l'autorisation de base et l'autorisation par clé API, EventBridge renseigne les en-têtes d'autorisation requis pour vous.
- OAuth

  Pour l'autorisation OAuth, échangez EventBridge également votre identifiant client et votre secret contre un jeton d'accès, puis gérez-le en toute sécurité.

  Les jetons OAUTH sont actualisés lorsqu'une réponse 401 ou 407 est renvoyée.

Lorsque vous créez une connexion, vous pouvez également inclure les paramètres d’en-tête, de corps et de requête qui sont requis pour l’autorisation auprès d’un point de terminaison. Vous pouvez

utiliser la même connexion pour plusieurs points de terminaison HTTP si l'autorisation pour le point de terminaison est identique.

Lorsque vous créez une connexion et que vous ajoutez des paramètres d'autorisation, EventBridge crée un secret dans AWS Secrets Manager. Le coût de l'accès au secret de Secrets Manager et de son stockage est inclus dans les frais d'utilisation d'une destination d'API. Pour en savoir plus sur les meilleures pratiques relatives à l'utilisation des secrets avec les destinations d'API, consultez AWS::Events::ApiDestinationle guide de CloudFormation l'utilisateur.

> **Note**
>
> Pour créer ou mettre à jour une connexion avec succès, vous devez utiliser un compte autorisé à utiliser Secrets Manager. L'autorisation requise est incluse dans la AmazonEventBridgeFullAccess politique. La même autorisation est accordée au rôle lié à un service créé dans votre compte pour la connexion.

Création de connexions pour les cibles de point de terminaison HTTP

Pour créer une connexion à utiliser avec les points de terminaison HTTP à l'aide de la console EventBridge

1. Connectez-vous à AWS l'aide d'un compte autorisé à gérer EventBridge et à ouvrir la EventBridge console.
2. Dans le volet de navigation de gauche, choisissez Destinations d'API.
3. Faites défiler la page jusqu'au tableau Destinations d'API, puis cliquez sur l'onglet Connexions.
4. Choisissez Créer une connexion.
5. Sur la page Créer une connexion, entrez le Nom de la connexion.
6. Entrez une Description pour la connexion.
7. Pour Type d'autorisation, sélectionnez le type d'autorisation à utiliser pour autoriser les connexions au point de terminaison HTTP spécifié pour la destination d'API qui utilise cette connexion. Effectuez l'une des actions suivantes :
   - Choisissez Basic (nom d'utilisateur/mot de passe), puis entrez le Nom d'utilisateur et le Mot de passe à utiliser pour autoriser la connexion auprès du point de terminaison HTTP.

- Choisissez Informations d’identification du client OAuth, puis entrez le Point de terminaison d’autorisation, la Méthode HTTP, l’ID client et la Clé secrète du client à utiliser pour autoriser la connexion auprès du point de terminaison.

  Sous Paramètres Http OAuth, ajoutez tous les paramètres supplémentaires à inclure pour l’autorisation auprès du point de terminaison d’autorisation. Sélectionnez un Paramètre dans la liste déroulante, puis entrez une Clé et une Valeur. Pour inclure un paramètre supplémentaire, choisissez Ajouter un paramètre.

  Sous Paramètres Http d’appel, ajoutez tous les paramètres supplémentaires à inclure dans la demande d’autorisation. Pour ajouter un paramètre, sélectionnez un Paramètre dans la liste déroulante, puis entrez une Clé et une Valeur. Pour inclure un paramètre supplémentaire, choisissez Ajouter un paramètre.
- Choisissez Clé API, puis entrez le Nom de clé de l’API et la Valeur associée à utiliser pour l’autorisation de la clé d’API.

  Sous Paramètres Http d’appel, ajoutez tous les paramètres supplémentaires à inclure dans la demande d’autorisation. Pour ajouter un paramètre, sélectionnez un Paramètre dans la liste déroulante, puis entrez une Clé et une Valeur. Pour inclure un paramètre supplémentaire, choisissez Ajouter un paramètre.

8. Choisissez Créer.

Modification des connexions à l'aide de la EventBridge console

Vous pouvez modifier les connexions existantes.

Pour modifier une connexion à l'aide de la EventBridge console

1. Connectez-vous à AWS l'aide d'un compte autorisé à gérer EventBridge et à ouvrir la EventBridge console.
2. Dans le volet de navigation de gauche, choisissez Destinations d’API.
3. Faites défiler la page jusqu’au tableau Destinations d’API, puis cliquez sur l’onglet Connexions.
4. Dans le tableau Connexions, choisissez la connexion à modifier.
5. Sur la page Détails de la connexion, choisissez Modifier.
6. Mettez à jour les valeurs de la connexion, puis choisissez Mettre à jour.

Annulation de l'autorisation des connexions à l'aide de la console EventBridge

Lorsque vous annulez l’autorisation d’une connexion, tous les paramètres d’autorisation sont supprimés. La suppression des paramètres d’autorisation supprime le secret de la connexion, ce qui vous permet de le réutiliser sans avoir à créer une nouvelle connexion.

> **Note**
>
> Vous devez mettre à jour tous les points de terminaison HTTP qui utilisent la connexion non autorisée afin d'utiliser une autre connexion afin d'envoyer correctement des demandes au point de terminaison HTTP.

Pour annuler l’autorisation d’une connexion

1. Connectez-vous à AWS l'aide d'un compte autorisé à gérer EventBridge et à ouvrir la EventBridge console.
2. Dans le volet de navigation de gauche, choisissez Destinations d’API.
3. Faites défiler la page jusqu’au tableau Destinations d’API, puis cliquez sur l’onglet Connexions.
4. Dans le tableau Connexions, choisissez la connexion.
5. Sur la page Détails de la connexion, choisissez Annuler l’autorisation.
6. Dans la boîte de dialogue Annuler l’autorisation de la connexion ?, entrez le nom de la connexion, puis choisissez Annuler l’autorisation.

Le statut de la connexion passe à Annuler l’autorisation jusqu’à ce que le processus soit terminé. Le statut passe ensuite à Autorisation annulée. Vous pouvez à présent modifier la connexion de sorte à ajouter de nouveaux paramètres d’autorisation.

Envoyer et recevoir des EventBridge événements Amazon entre AWS comptes

Vous pouvez configurer EventBridge pour envoyer et recevoir des événements entre les bus d'événements des AWS comptes. Lorsque vous configurez EventBridge pour envoyer ou recevoir des événements entre comptes, vous pouvez spécifier quels AWS comptes peuvent envoyer ou recevoir des événements depuis le bus d'événements de votre compte. Vous pouvez également autoriser ou refuser des événements à partir de règles spécifiques associées au bus d’événements,

ou des événements provenant de sources spécifiques. Pour plus d'informations, consultez Simplifier l'accès entre comptes grâce aux politiques de ressources d'Amazon EventBridge

> **Note**
>
> Si vous l'utilisez AWS Organizations, vous pouvez spécifier une organisation et accorder l'accès à tous les comptes de cette organisation. En outre, des rôles IAM doivent être attachés au bus d'événements lors de l'envoi d'événements à un autre compte. Pour plus d'informations, consultez Présentation de AWS Organizations dans le Guide de l'utilisateur AWS Organizations .

> **Note**
>
> Si vous utilisez un plan de réponse Incident Manager en tant que cible, tous les plans de réponse partagés avec votre compte sont disponibles par défaut.

Vous pouvez envoyer et recevoir des événements entre bus d'événements sur AWS des comptes d'une même région dans toutes les régions et entre des comptes situés dans différentes régions, à condition que la région de destination soit une région de destination interrégionale prise en charge.

Les étapes à suivre EventBridge pour configurer l'envoi ou la réception d'événements depuis un bus d'événements d'un autre compte sont les suivantes :

- Sur le compte destinataire, modifiez les autorisations sur un bus d'événements pour autoriser AWS des comptes spécifiques, une organisation ou tous les AWS comptes à envoyer des événements au compte destinataire.
- Sur le compte expéditeur, configurez une ou plusieurs règles comportant le bus d'événement du compte récepteur comme cible.

  Si le compte expéditeur hérite des autorisations d'envoi d'événements d'une AWS organisation, le compte expéditeur doit également avoir un rôle IAM avec des politiques lui permettant d'envoyer des événements au compte destinataire. Si vous utilisez le AWS Management Console pour créer la règle qui cible le bus d'événements dans le compte récepteur, le rôle est créé automatiquement. Si vous utilisez le 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.

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 Amazon EventBridge Pricing.

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.

Si vous avez trois bus d'événements dans le même compte et que vous configurez une règle sur le premier bus d'événements pour transférer les événements du deuxième bus d'événements vers un troisième bus d'événements, ces événements ne sont pas envoyés au troisième bus d'événements.

La vidéo suivante décrit les événements de routage entre comptes : Acheminement des événements vers les bus d'autres AWS comptes

Accorder des autorisations pour autoriser des événements provenant d'autres AWS comptes

Pour recevoir des événements d’autres comptes ou organisations, vous devez d’abord modifier les autorisations sur le bus d’événements où vous prévoyez de recevoir des événements. Le bus d'événements par défaut accepte les événements provenant de AWS services, d'autres AWS comptes autorisés et d'`PutEvents`appels. Les autorisations pour un bus d’événements sont accordées ou refusées à l’aide d’une politique basée sur les ressources attachée au bus d’événements. Dans la politique, vous pouvez accorder des autorisations à d'autres AWS comptes à l'aide de l'ID de compte ou à une AWS organisation à l'aide de l'ID d'organisation. Pour en savoir plus sur les autorisations relatives au bus d’événements, y compris des exemples de politiques, consultez Autorisations pour les bus d'événements Amazon EventBridge.

> **Note**
>
> EventBridge nécessite désormais l'ajout de rôles IAM à toutes les nouvelles cibles de bus d'événements entre comptes. Cela ne s’applique qu’aux cibles de bus d’événements créées après le 2 mars 2023. Les applications créées sans rôle IAM avant cette date ne sont pas concernées. Toutefois, nous recommandons d’ajouter des rôles IAM pour accorder aux utilisateurs l’accès aux ressources d’un autre compte, car cela garantit que les limites de l’organisation basées sur des politiques de contrôle des services (SCP) sont appliquées afin de déterminer qui peut envoyer et recevoir des événements depuis les comptes de votre organisation.

> **Important**
>
> Si vous choisissez de recevoir des événements provenant de tous les AWS comptes, veillez à créer des règles qui ne correspondent qu'aux événements destinés aux autres comptes. 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 plus d'informations, consultez EventBridge Événements Amazon.

Règles relatives aux événements entre AWS comptes

Si votre compte est configuré pour recevoir des événements provenant de bus d'événements sur d'autres AWS comptes, vous pouvez rédiger des règles correspondant à ces événements. Définissez le modèle d'événement de la règle pour correspondre aux événements que vous recevez des bus d'événements dans l'autre compte.

À moins que vous ne spécifiiez `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 de bus d'événements dans d'autres comptes se déclenchent en fonction de ces événements. Si vous recevez des événements de bus d'événements dans 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 AWS compte pour accepter les événements provenant des bus d'événements sur tous les AWS comptes, nous vous recommandons vivement d'ajouter des `account` éléments à chaque EventBridge règle de votre compte. Cela empêche les règles de votre compte de se déclencher en cas d'événements provenant de AWS comptes 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 déclenche un événement correspondant à partir de n'importe quel bus d'événements du AWS compte auquel vous avez accordé des autorisations, ne spécifiez pas * dans le `account` champ 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.

Création de règles permettant d'envoyer des événements entre AWS comptes

La spécification d’un bus d’événements dans un autre compte en tant que cible fait partie de la création de la règle.

Pour créer une règle qui envoie des événements à un autre AWS compte à l'aide de la console

1. Suivez les étapes de la procédure ???.
2. Au cours de l’étape ???, lorsque vous êtes invité à choisir un type de cible :
    a. Sélectionnez le bus EventBridge événementiel.
    b. Sélectionnez Bus d’événements dans un compte ou une région différent.
    c. Pour Bus d’événements comme cible, entrez l’ARN du bus d’événements que vous souhaitez utiliser.
3. Créez la règle en suivant les étapes de la procédure.

Envoyer et recevoir des EventBridge événements Amazon entre les AWS régions

Vous pouvez configurer EventBridge pour envoyer et recevoir des événements entre les AWS régions. Vous pouvez également autoriser ou refuser des événements à partir de régions spécifiques et de règles spécifiques associées au bus d’événements, ou des événements provenant de sources spécifiques. Pour plus d'informations, consultez Présentation du routage d'événements entre régions avec Amazon EventBridge

Les régions suivantes sont des régions de destination prises en charge :

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Californie du Nord)
- USA Ouest (Oregon)
- Afrique (Le Cap)
- Asie-Pacifique (Hong Kong)

- Asie-Pacifique (Tokyo)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Osaka)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Hyderabad)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Melbourne)
- Canada (Centre)
- Canada Ouest (Calgary)
- Europe (Francfort)
- Europe (Espagne)
- Europe (Zurich)
- Europe (Stockholm)
- Europe (Milan)
- Europe (Irlande)
- Europe (Londres)
- Europe (Paris)
- Israël (Tel Aviv)
- Moyen-Orient (EAU)
- Moyen-Orient (Bahreïn)
- Amérique du Sud (Sao Paulo)

La vidéo suivante décrit les événements de routage entre les régions à l'aide du site https://console.aws.amazon.com/events/ AWS CloudFormation, et AWS Serverless Application Model : Le routage des événements entre régions

Création de règles qui envoient des événements vers une autre AWS région

La définition d'un bus d'événements dans une autre AWS région comme cible fait partie de la création de la règle.

Pour créer une règle qui envoie des événements à un autre AWS compte à l'aide de la console

1. Suivez les étapes de la procédure ???.
2. Au cours de l'étape ???, lorsque vous êtes invité à choisir un type de cible :
   a. Sélectionnez le bus EventBridge événementiel.
   b. Sélectionnez Bus d'événements dans un compte ou une région différent.
   c. Pour Bus d'événements comme cible, entrez l'ARN du bus d'événements que vous souhaitez utiliser.
3. Créez la règle en suivant les étapes de la procédure.

Envoyer et recevoir des EventBridge événements Amazon entre les bus d'événements d'un même compte et d'une même région

Vous pouvez configurer EventBridge pour envoyer et recevoir des événements entre les bus d'événements d'un même AWS compte et d'une même région.

Lorsque vous configurez EventBridge pour envoyer ou recevoir des événements entre des bus d'événements, vous utilisez des rôles IAM sur le bus d'événements de l'expéditeur pour autoriser le bus d'événements de l'expéditeur à envoyer des événements au bus d'événements du récepteur. Vous utilisez des politiques basées sur les ressources sur le bus d'événements récepteur pour autoriser le bus d'événements récepteur à recevoir des événements du bus d'événements expéditeur. Vous pouvez également autoriser ou refuser des événements à partir de certains bus d'événements et de règles spécifiques associées au bus d'événements, ou des événements provenant de sources spécifiques. Pour plus d'informations sur les autorisations relatives au bus d'événements, y compris des exemples de politiques, consultez Autorisations pour les bus d'événements Amazon EventBridge.

Les étapes à suivre EventBridge pour configurer l'envoi ou la réception d'événements entre les bus d'événements de votre compte sont les suivantes :

- Pour utiliser un rôle IAM existant, vous devez accorder des autorisations de bus d'événements expéditeur au bus d'événements récepteur ou des autorisations de bus d'événements récepteur au bus d'événements expéditeur.
- Sur le bus d'événements expéditeur, configurez une ou plusieurs règles comportant le bus d'événements récepteur en tant que cible et créez un rôle IAM. Pour obtenir un exemple de la politique qui doit être attachée au rôle, consultez ???.
- Sur le bus d'événements récepteur, modifiez les autorisations pour autoriser le transfert d'événements à partir de l'autre bus d'événements.
- Sur l'événement récepteur, configurez une ou plusieurs des règles qui correspondent à des événements provenant du bus d'événements expéditeur.

> **Note**
>
> EventBridge Impossible d'acheminer les événements reçus d'un bus d'événements de l'expéditeur vers un troisième bus d'événements.

Les événements envoyés d'un bus d'événements à un autre sont facturés en tant qu'événements personnalisés. Pour en savoir plus, consultez Tarification Amazon EventBridge.

Création de règles qui envoient des événements à un bus d'événements différent dans le même AWS compte et dans la même région

Pour envoyer des événements vers un autre bus d'événements, vous devez créer une règle avec un bus d'événements en tant que cible. La définition d'un bus d'événements dans le même AWS compte et dans la même région comme cible fait partie de la création de la règle.

Pour créer une règle qui envoie des événements à un bus d'événements différent dans le même AWS compte et dans la même région à l'aide de la console

1. Suivez les étapes de la procédure ???.
2. Au cours de l'étape ???, lorsque vous êtes invité à choisir un type de cible :
   a. Sélectionnez le bus EventBridge événementiel.
   b. Sélectionnez Event Bus dans le même AWS compte et dans la même région.
   c. Pour Bus d'événements en tant que cible, sélectionnez un bus d'événements dans la liste déroulante.

3. Créez la règle en suivant les étapes de la procédure.

Transformation des EventBridge entrées Amazon

Vous pouvez personnaliser le texte d'un événement avant de EventBridge transmettre les informations à la cible d'une règle. À l'aide du transformateur d'entrée de la console ou de l'API, vous définissez des variables qui utilisent le chemin JSON pour référencer les valeurs de la source d'événement d'origine. L'événement transformé est envoyé à une cible plutôt qu'à l'événement d'origine. Toutefois, les paramètres de chemin dynamiques doivent faire référence à l'événement d'origine, et non à l'événement transformé. Vous pouvez définir jusqu'à 100 variables, en attribuant à chacune une valeur à partir de l'entrée. Vous pouvez ensuite utiliser ces variables dans le modèle d'entrée au format <*variable-name*>.

Pour obtenir un didacticiel sur l'utilisation du transformateur d'entrée, consultez ???.

> **Note**
>
> EventBridge ne prend pas en charge toutes les syntaxes JSON Path et ne l'évalue pas lors de l'exécution. La syntaxe prise en charge inclut :
>
> - notation par points (par exemple, `$.detail`)
> - tirets
> - traits de soulignement
> - caractères alphanumériques
> - index de tableau
> - caractères génériques (*)

Dans cette rubrique :

Variables prédéfinies

Il existe des variables prédéfinies que vous pouvez utiliser sans définir de chemin JSON. Ces variables sont réservées et vous ne pouvez pas créer de variables avec ces noms :

- `aws.events.rule-arn`— Le nom de ressource Amazon (ARN) de la EventBridge règle.
- `aws.events.rule-name`— Le nom de la EventBridge règle.
- `aws.events.event.ingestion-time`— L'heure à laquelle l'événement a été reçu par EventBridge. Il s'agit d'un horodatage ISO 8601. Cette variable est générée par EventBridge et ne peut pas être remplacée.
- `aws.events.event` : la charge utile de l'événement d'origine au format JSON (sans le champ `detail`). Peut seulement être utilisée comme valeur pour un champ JSON, car son contenu n'est pas mis en échappement.
- `aws.events.event.json` : la charge utile complète de l'événement d'origine au format JSON (avec le champ `detail`). Peut seulement être utilisée comme valeur pour un champ JSON, car son contenu n'est pas mis en échappement.

Exemples de transformation d'entrée

Voici un exemple d'événement Amazon EC2.

```
{
  "version": "0",
  "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-abcd1111"
  ],
  "detail": {
    "instance-id": "i-0123456789",
    "state": "RUNNING"
  }
}
```

Lorsque vous définissez une règle dans la console, sélectionnez l'option Transformateur d'entrée sous Configurer l'entrée. Cette option affiche deux zones de texte : une pour Chemin d'entrée et l'autre pour Modèle d'entrée.

Le chemin d’entrée est utilisé pour définir des variables. Utilisez le chemin JSON pour référencer des éléments dans votre événement et stocker ces valeurs dans des variables. Par exemple, vous pouvez créer un chemin d'entrée pour référencer des valeurs dans l'exemple d'événement en entrant les informations suivantes dans la première zone de texte. Vous pouvez également utiliser des crochets et des index pour obtenir des éléments à partir de tableaux.

> **Note**
>
> EventBridge remplace les transformateurs d'entrée lors de l'exécution pour garantir une sortie JSON valide. Pour cette raison, placez des guillemets autour des variables qui font référence aux paramètres de chemin JSON, mais ne placez pas de guillemets autour des variables qui font référence à des objets ou des tableaux JSON.

```
{
  "timestamp" : "$.time",
  "instance" : "$.detail.instance-id",
  "state" : "$.detail.state",
  "resource" : "$.resources[0]"
}
```

Cela permet de définir quatre variables : `<timestamp>`, `<instance>`, `<state>` et `<resource>`. Vous pouvez référencer ces variables lors de la création de votre modèle d’entrée.

Le modèle d’entrée est un modèle pour les informations que vous souhaitez transmettre à votre cible. Vous pouvez créer un modèle qui transmet une chaîne ou un fichier JSON à la cible. À l'aide de l'événement précédent et du chemin d'entrée, les exemples de modèle d’entrée suivants transformeront l'événement en l’exemple de sortie suivant avant de l’acheminer vers une cible.

Description	Modèle	Sortie
Chaîne simple	`"instance <instance> is in <state>"`	`"instance i-0123456789 is in RUNNING"`

Description	Modèle	Sortie
Chaîne avec guillemets échappés	`"instance \"<instance> \" is in <state>"`	`"instance \"i-01234 56789\" is in RUNNING"` Notez qu'il s'agit du comportement de la EventBrid ge console. L' AWS CLI échappe les caractères de barre oblique et le résultat est `"instance "i-012345 6789" is in RUNNING"` .
Fichier JSON simple	`{ "instance" : <instance>, "state": <state> }`	`{ "instance" : "i-0123456789", "state": "RUNNING" }`
Fichier JSON avec chaînes et variables	`{ "instance" : <instance >, "state": "<state>", "instanceStatus": "instance \"<instance> \" is in <state>" }`	`{ "instance" : "i-012345 6789", "state": "RUNNING", "instanceStatus": "instance \"i-01234 56789\" is in RUNNING" }`
Fichier JSON avec un mélange de variables et d'informations statiques	`{ "instance" : <instance>, "state": [ 9, <state>, true ], "Transformed" : "Yes" }`	`{ "instance" : "i-0123456789", "state": [ 9, "RUNNING", true ], "Transformed" : "Yes" }`

Description	Modèle	Sortie
Inclut des variables réservées dans JSON	`{ "instance" : <instance>, "state": <state>, "ruleArn" : <aws.events.rule-arn>, "ruleName" : <aws.events.rule-name>, "originalEvent" : <aws.events.event.json> }`	`{ "instance" : "i-0123456789", "state": "RUNNING", "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example", "ruleName" : "example", "originalEvent" : { ... // commented for brevity } }`
Inclut des variables réservées dans une chaîne	`"<aws.events.rule-name> triggered"`	`"example triggered"`
Groupe de CloudWatch journaux Amazon	`{ "timestamp" : <timestamp>, "message": "instance \"<instance>\" is in <state>" }`	`{ "timestamp" : 2015-11-11T21:29:54Z, "message": "instance "i-0123456789" is in RUNNING }`

Transformation des entrées à l'aide de l' EventBridgeAPI

Pour plus d'informations sur l'utilisation de l' EventBridge API pour transformer les entrées, voir Utiliser le transformateur d'entrée pour extraire les données d'un événement et les saisir dans la cible.

Transformation des données en utilisant AWS CloudFormation

Pour plus d'informations sur l'utilisation AWS CloudFormation pour transformer les entrées, consultez AWS::Events::Rule InputTransformer.

Problèmes courants liés à la transformation d’entrée

Voici quelques problèmes courants lors de la transformation des entrées en EventBridge :

- Pour les chaînes, des guillemets sont requis.
- Il n'y a pas de validation lors de la création du chemin JSON pour votre modèle.
- Si vous spécifiez une variable à mettre en correspondance avec un chemin JSON qui n’existe pas dans l’événement, cette variable n’est pas créée et n’apparaîtra pas dans la sortie.
- Les propriétés JSON telles que `aws.events.event.json` ne peuvent être utilisées que comme valeur d’un champ JSON, et non en ligne dans d’autres chaînes.
- EventBridge n'échappe pas aux valeurs extraites par Input Path, lors du remplissage du modèle d'entrée pour une cible.
- Si un chemin JSON fait référence à un objet ou à un tableau JSON, mais que la variable est référencée dans une chaîne, EventBridge supprime tous les guillemets internes pour garantir la validité de la chaîne. Par exemple, pour une variable `<detail>` pointée`$.detail`, « Detail is <detail>» entraînerait la EventBridge suppression des guillemets de l'objet.

  Par conséquent, si vous souhaitez générer un objet JSON basé sur une seule variable de chemin JSON, vous devez le placer sous forme de clé. Dans cet exemple, `{"detail": <detail>}`.
- Les guillemets ne sont pas obligatoires pour les variables qui représentent des chaînes. Ils sont autorisés, mais ajoutent EventBridge automatiquement des guillemets aux valeurs des variables de chaîne lors de la transformation, afin de garantir que le résultat de la transformation est un JSON valide. EventBridge n'ajoute pas de guillemets aux variables qui représentent des objets ou des tableaux JSON. N’ajoutez pas de guillemets pour les variables qui représentent des objets ou des tableaux JSON.

  Par exemple, le modèle d’entrée suivant inclut des variables qui représentent à la fois des chaînes et des objets JSON :

```
{
  "ruleArn" : <aws.events.rule-arn>,
  "ruleName" : <aws.events.rule-name>,
```

```
  "originalEvent" : <aws.events.event.json>
}
```

Le résultat est un code JSON valide avec des guillemets appropriés :

```
{
  "ruleArn" : "arn:aws:events:us-east-2:123456789012:rule/example",
  "ruleName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Pour une sortie de texte (non JSON) sous forme de chaînes multilignes, placez chaque ligne distincte de votre modèle de saisie entre guillemets.

  Par exemple, si vous faisiez correspondre les événements de Amazon Inspector recherche au modèle d'événements suivant :

```
{
  "detail": {
    "severity": ["HIGH"],
    "status": ["ACTIVE"]
  },
  "detail-type": ["Inspector2 Finding"],
  "source": ["inspector2"]
}
```

  Et en utilisant le chemin d'entrée suivant :

```
{
  "account": "$.detail.awsAccountId",
  "ami": "$.detail.resources[0].details.awsEc2Instance.imageId",
  "arn": "$.detail.findingArn",
  "description": "$.detail.description",
  "instance": "$.detail.resources[0].id",
  "platform": "$.detail.resources[0].details.awsEc2Instance.platform",
  "region": "$.detail.resources[0].region",
  "severity": "$.detail.severity",
  "time": "$.time",
  "title": "$.detail.title",
  "type": "$.detail.type"
```

```
}
```

Vous pouvez utiliser le modèle de saisie ci-dessous pour générer une sortie de chaîne multiligne :

```
"<severity> severity finding <title>"
"Description: <description>"
"ARN: \"<arn>\""
"Type: <type>"
"AWS Account: <account>"
"Region: <region>"
"EC2 Instance: <instance>"
"Platform: <platform>"
"AMI: <ami>"
```

Configuration d’un transformateur d’entrée dans le cadre de la création d’une règle

Lors de la création d'une règle, vous pouvez spécifier un transformateur d'entrée EventBridge à utiliser pour traiter les événements correspondants avant de les envoyer à la cible spécifiée. Vous pouvez configurer des transformateurs d'entrée pour des cibles qui sont des AWS services ou des destinations d'API.

Pour créer un transformateur d’entrée cible dans le cadre d’une règle

1. Suivez les étapes de création d’une règle, comme décrit dans ???.
2. À l’Étape 3 : Sélectionner la ou les cibles, développez Réglages supplémentaires.
3. Pour Configurer l’entrée cible, choisissez Transformateur d’entrée dans la liste déroulante.

   Cliquez sur Configurer le transformateur d’entrée.

   EventBridge affiche la boîte de dialogue Configurer le transformateur d'entrée.
4. Dans la section Exemple d’événement, choisissez un Type d’exemple d’événement par rapport auquel vous souhaitez tester votre modèle d’événement. Vous pouvez choisir un AWS événement, un événement partenaire ou créer votre propre événement personnalisé.

   AWS events

   Faites votre choix parmi les événements émis par les Services AWS pris en charge.

1. Sélectionnez Événements AWS .
2. Sous Exemples d'événements, sélectionnez l' AWS événement souhaité. Les événements sont organisés par AWS service.

   Lorsque vous sélectionnez un événement, il EventBridge renseigne l'exemple d'événement.

   Par exemple, si vous choisissez S3 Object Created, EventBridge affiche un exemple d'événement S3 Object Created.
3. (Facultatif) Vous pouvez également sélectionner Copier pour copier l’exemple d’événement dans le presse-papiers de votre appareil.

Partner events

Choisissez parmi les événements émis par les services tiers pris en charge EventBridge, tels que Salesforce.

1. Sélectionnez les événements EventBridge partenaires.
2. Sous Exemples d’événements, choisissez l’événement partenaire souhaité. Les événements sont organisés par partenaire.

   Lorsque vous sélectionnez un événement, il EventBridge renseigne l'exemple d'événement.
3. (Facultatif) Vous pouvez également sélectionner Copier pour copier l’exemple d’événement dans le presse-papiers de votre appareil.

Enter your own

Entrez votre propre événement au format texte JSON.

1. Sélectionnez Saisir mon propre.
2. EventBridge remplit l'exemple d'événement avec un modèle d'attributs d'événement obligatoires.
3. Modifiez et faites des ajouts à l’exemple d’événement selon vos besoins. L’exemple d’événement doit être au format JSON valide.
4. (Facultatif) Vous pouvez également choisir l’une des options suivantes :

- Copier : copiez l’exemple d’événement dans le presse-papiers de votre appareil.
- Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.

5. (Facultatif) Développez la section Exemples de chemins d’entrée, de modèles et de sorties pour afficher les exemples suivants :

   - Comment les chemins JSON sont-ils utilisés pour définir des variables qui représentent les données d’événement
   - Comment ces variables peuvent être utilisées dans un modèle de transformateur d’entrée
   - La sortie résultante qui est EventBridge envoyée à la cible

   Pour obtenir des exemples plus détaillés de transformations d’entrée, consultez ???.

6. Dans la section Transformateur d’entrée cible, définissez les variables que vous souhaitez utiliser dans le modèle d’entrée.

   Les variables utilisent le chemin JSON pour référencer des valeurs dans la source d’événement d’origine. Vous pouvez ensuite référencer ces variables dans le modèle d'entrée afin d'inclure les données de l'événement source d'origine dans l'événement transformé EventBridge transmis à la cible. Vous pouvez définir jusqu’à 100 variables. Le transformateur d’entrée doit être au format JSON valide.

   Supposons, par exemple, que vous ayez choisi l' AWS événement S3 Object Created comme exemple d'événement pour ce transformateur d'entrée. Vous pouvez alors définir les variables suivantes à utiliser dans votre modèle :

   ```
   {
     "requester": "$.detail.requester",
     "key": "$.detail.object.key",
     "bucket": "$.detail.bucket.name"
   }
   ```

   (Facultatif) Vous pouvez également choisir Copier pour copier le transformateur d’entrée dans le presse-papiers de votre appareil.

7. Dans la section Modèle, composez le modèle que vous souhaitez utiliser pour déterminer ce qui EventBridge passe à la cible.

Vous pouvez utiliser du code JSON, des chaînes, des informations statiques, les variables que vous avez définies ainsi que des variables réservées. Pour obtenir des exemples plus détaillés de transformations d'entrée, consultez ???.

Par exemple, supposons que vous avez défini les variables dans l'exemple précédent. Vous pouvez alors composer le modèle suivant, qui fait référence à ces variables, ainsi qu'aux variables réservées et aux informations statiques.

```
{
    "message": "<requester> has created the object \"<key>\" in the bucket
 \"<bucket>\"",
    "RuleName": <aws.events.rule-name>,
    "ruleArn" : <aws.events.rule-arn>,
    "Transformed": "Yes"
}
```

(Facultatif) Vous pouvez également choisir Copier pour copier le modèle dans le presse-papiers de votre appareil.

8. Pour tester votre modèle, sélectionnez Générer une sortie.

   EventBridge traite l'exemple d'événement en fonction du modèle d'entrée et affiche la sortie transformée générée sous Sortie. Il s'agit des informations qui EventBridge seront transmises à la cible à la place de l'événement source d'origine.

   La sortie générée pour l'exemple de modèle d'entrée décrit ci-dessus est la suivante :

   ```
   {
       "message": "123456789012 has created the object "example-key" in the bucket
    "example-bucket"",
       "RuleName": rule-name,
       "ruleArn" : arn:aws:events:us-east-1:123456789012:rule/rule-name,
       "Transformed": "Yes"
   }
   ```

   (Facultatif) Vous pouvez également choisir Copier pour copier la sortie générée dans le presse-papiers de votre appareil.

9. Sélectionnez Confirmer.
10. Suivez le reste des étapes de création d'une règle, comme décrit dans ???.

Test d'un transformateur d'entrée cible à l'aide du EventBridge Sandbox

Vous pouvez utiliser des transformateurs d'entrée pour personnaliser le texte d'un événement avant de EventBridge transmettre les informations à la cible d'une règle.

La configuration d’un transformateur d’entrée fait généralement partie du processus plus vaste qui consiste à spécifier une cible lors de la création d’une nouvelle règle ou de la modification d’une règle existante. En utilisant le Sandbox EventBridge, vous pouvez toutefois configurer rapidement un transformateur d'entrée et utiliser un exemple d'événement pour confirmer que vous obtenez la sortie souhaitée, sans avoir à créer ou à modifier de règle.

Pour plus d’informations sur les transformations d’entrée, consultez ???.

Pour tester un transformateur d’entrée cible

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Sous Ressources pour développeurs, choisissez Environnement de test (sandbox), puis sur la page Environnement de test (sandbox), cliquez sur l’onglet Transformateur d’entrée cible.
3. Dans la section Exemple d’événement, choisissez un Type d’exemple d’événement par rapport auquel vous souhaitez tester votre modèle d’événement. Vous pouvez choisir un AWS événement, un événement partenaire ou créer votre propre événement personnalisé.

   AWS events

   Faites votre choix parmi les événements émis par les Services AWS pris en charge.

   1. Sélectionnez Événements AWS .
   2. Sous Exemples d'événements, sélectionnez l' AWS événement souhaité. Les événements sont organisés par AWS service.

      Lorsque vous sélectionnez un événement, il EventBridge renseigne l'exemple d'événement.

      Par exemple, si vous choisissez S3 Object Created, EventBridge affiche un exemple d'événement S3 Object Created.
   3. (Facultatif) Vous pouvez également sélectionner Copier pour copier l’exemple d’événement dans le presse-papiers de votre appareil.

Partner events

Choisissez parmi les événements émis par les services tiers pris en charge EventBridge, tels que Salesforce.

1. Sélectionnez les événements EventBridge partenaires.
2. Sous Exemples d’événements, choisissez l’événement partenaire souhaité. Les événements sont organisés par partenaire.

   Lorsque vous sélectionnez un événement, il EventBridge renseigne l'exemple d'événement.
3. (Facultatif) Vous pouvez également sélectionner Copier pour copier l’exemple d’événement dans le presse-papiers de votre appareil.

Enter your own

Entrez votre propre événement au format texte JSON.

1. Sélectionnez Saisir mon propre.
2. EventBridge remplit l'exemple d'événement avec un modèle d'attributs d'événement obligatoires.
3. Modifiez et faites des ajouts à l’exemple d’événement selon vos besoins. L’exemple d’événement doit être au format JSON valide.
4. (Facultatif) Vous pouvez également choisir l’une des options suivantes :
   - Copier : copiez l’exemple d’événement dans le presse-papiers de votre appareil.
   - Prettify : facilite la lecture du texte JSON en ajoutant des sauts de ligne, des tabulations et des espaces.

4. (Facultatif) Développez la section Exemples de chemins d’entrée, de modèles et de sorties pour afficher les exemples suivants :

- Comment les chemins JSON sont-ils utilisés pour définir des variables qui représentent les données d’événement
- Comment ces variables peuvent être utilisées dans un modèle de transformateur d’entrée
- La sortie résultante qui est EventBridge envoyée à la cible

Pour obtenir des exemples plus détaillés de transformations d'entrée, consultez ???.

5. Dans la section Transformateur d'entrée cible, définissez les variables que vous souhaitez utiliser dans le modèle d'entrée.

   Les variables utilisent le chemin JSON pour référencer des valeurs dans la source d'événement d'origine. Vous pouvez ensuite référencer ces variables dans le modèle d'entrée afin d'inclure les données de l'événement source d'origine dans l'événement transformé EventBridge transmis à la cible. Vous pouvez définir jusqu'à 100 variables. Le transformateur d'entrée doit être au format JSON valide.

   Supposons, par exemple, que vous ayez choisi l' AWS événement S3 Object Created comme exemple d'événement pour ce transformateur d'entrée. Vous pouvez alors définir les variables suivantes à utiliser dans votre modèle :

   ```
   {
     "requester": "$.detail.requester",
     "key": "$.detail.object.key",
     "bucket": "$.detail.bucket.name"
   }
   ```

   (Facultatif) Vous pouvez également choisir Copier pour copier le transformateur d'entrée dans le presse-papiers de votre appareil.

6. Dans la section Modèle, composez le modèle que vous souhaitez utiliser pour déterminer ce qui EventBridge passe à la cible.

   Vous pouvez utiliser du code JSON, des chaînes, des informations statiques, les variables que vous avez définies ainsi que des variables réservées. Pour obtenir des exemples plus détaillés de transformations d'entrée, consultez ???.

   Par exemple, supposons que vous avez défini les variables dans l'exemple précédent. Vous pouvez alors composer le modèle suivant, qui fait référence à ces variables, ainsi qu'aux variables réservées et aux informations statiques.

   ```
   {
       "message": "<requester> has created the object \"<key>\" in the bucket
    \"<bucket>\"",
       "RuleName": <aws.events.rule-name>,
       "ruleArn" : <aws.events.rule-arn>,
   ```

```
    "Transformed": "Yes"
}
```

(Facultatif) Vous pouvez également choisir Copier pour copier le modèle dans le presse-papiers de votre appareil.

7. Pour tester votre modèle, sélectionnez Générer une sortie.

   EventBridge traite l'exemple d'événement en fonction du modèle d'entrée et affiche la sortie transformée générée sous Sortie. Il s'agit des informations qui EventBridge seront transmises à la cible à la place de l'événement source d'origine.

   La sortie générée pour l’exemple de modèle d’entrée décrit ci-dessus est la suivante :

```
{
    "message": "123456789012 has created the object "example-key" in the bucket
 "example-bucket"",
    "RuleName": rule-name,
    "ruleArn" : arn:aws:events:us-east-1:123456789012:rule/rule-name,
    "Transformed": "Yes"
}
```

   (Facultatif) Vous pouvez également choisir Copier pour copier la sortie générée dans le presse-papiers de votre appareil.

Archivage-relecture Amazon EventBridge

Dans EventBridge, vous pouvez créer une archive d'événements pour pouvoir facilement les relire ultérieurement. Par exemple, vous souhaitez peut-être relire des événements pour récupérer suite à des erreurs ou pour valider une nouvelle fonctionnalité de votre application.

> **Note**
>
> Il peut y avoir un délai entre la publication d'un événement sur un bus d'événements et son arrivée dans l'archive. Nous vous recommandons de retarder la relecture des événements archivés de 10 minutes pour garantir que tous les événements soient relus.

La vidéo suivante montre l'utilisation de l'archivage-relecture : Création d'archives et de relectures

Rubriques

- Archivage des événements Amazon EventBridge
- Relecture d'événements Amazon EventBridge archivés

Archivage des événements Amazon EventBridge

Lorsque vous créez une archive dans EventBridge, vous pouvez déterminer quels événements sont envoyés à l'archive en spécifiant un modèle d'événement. EventBridge envoie des événements qui correspondent au modèle d'événement à l'archive. Vous définissez également la période de conservation pour stocker les événements dans l'archive avant qu'ils ne soient supprimés.

Par défaut, EventBridge chiffre les données d'événements d'une archive à l'aide de la norme de chiffrement avancée 256 bits (AES-256) sous une clé CMK AWS propriétaire, ce qui permet de protéger vos données contre tout accès non autorisé.

> **Note**
>
> Les `SizeBytes` valeurs `EventCount` et de l'`DescribeArchive`opération ont une période de rapprochement de 24 heures. Par conséquent, les événements récemment expirés ou récemment archivés peuvent ne pas être immédiatement reflétés dans ces valeurs.

Pour créer une archive de tous les événements

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation de gauche, choisissez Archives.
3. Choisissez Créer une archive.
4. Sous Informations de l'archive, entrez le Nom de l'archive. Le nom doit être unique dans la région sélectionnée pour votre compte.

   Vous ne pouvez pas modifier le nom une fois que vous avez créé l'archive.
5. (Facultatif) Entrez une Description de l'archive.
6. Pour Source, sélectionnez le bus d'événements qui émet les événements à envoyer à l'archive.
7. Pour Période de conservation, effectuez l'une des opérations suivantes :
   - Choisissez Indéfini pour conserver les événements dans l'archive et ne jamais les supprimer.
   - Entrez le nombre de jours pendant lesquels les événements doivent être conservés. Après le nombre de jours spécifié, EventBridge supprime les événements de l'archive.
8. Choisissez Suivant.
9. Sous Modèle d'événement, choisissez Aucun filtrage d'événement.

10. Choisissez Créer une archive.

Pour créer une archive avec un modèle d’événement

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation de gauche, choisissez Archives.
3. Choisissez Créer une archive.
4. Sous Informations de l’archive, entrez le Nom de l’archive. Le nom doit être unique dans la région sélectionnée pour votre compte.

   Vous ne pouvez pas modifier le nom une fois que vous avez créé l’archive.
5. (Facultatif) Entrez une Description de l’archive.
6. Pour Source, sélectionnez le bus d’événements qui émet les événements à envoyer à l’archive.
7. Pour Période de conservation, effectuez l’une des opérations suivantes :
   - Choisissez Indéfini* pour conserver les événements dans l’archive et ne jamais les supprimer.
   - Entrez le nombre de jours pendant lesquels les événements doivent être conservés. Après le nombre de jours spécifié, EventBridge supprime les événements de l'archive.
8. Choisissez Suivant.
9. Sous Modèle d’événement, choisissez Filtrage des événements par correspondance de modèle d’événement.
10. Effectuez l’une des actions suivantes :
    - Sélectionnez Générateur de modèle, puis choisissez le Fournisseur de service. Si vous choisissez AWS, sélectionnez également le Nom du service AWS et le Type d’événement à utiliser dans le modèle.
    - Sélectionnez Éditeur JSON pour créer un modèle manuellement. Vous pouvez également copier le modèle à partir d’une règle, puis le coller dans l’éditeur JSON.
11. Choisissez Créer une archive.

Pour vérifier que les événements ont bien été envoyés à l'archive, vous pouvez utiliser le `DescribeArchive`fonctionnement de l' EventBridge API pour voir si cela `EventCount` reflète le nombre d'événements contenus dans l'archive. Si la valeur est égale à 0, il n’y a aucun événement dans l’archive.

Relecture d'événements Amazon EventBridge archivés

Après avoir créé une archive, vous pouvez relire les événements de l'archive. Par exemple, si vous mettez à jour une application avec une fonctionnalité supplémentaire, vous pouvez relire les événements historiques pour vous assurer qu'ils sont retraités afin de garantir la cohérence de l'application. Vous pouvez également utiliser une archive pour relire les événements pour une nouvelle fonctionnalité. Lorsque vous relisez des événements, vous pouvez spécifier l'archive à partir de laquelle les événements doivent être relus, l'heure de début et de fin de l'événement à relire, le bus d'événements ou une ou plusieurs règles selon lesquelles les événements doivent être relus.

Les événements ne sont pas nécessairement relus dans l'ordre dans lequel ils ont été ajoutés à l'archive. Une relecture traite les événements à relire en fonction de l'heure à laquelle ils se sont produits et les relit toutes les minutes. Si vous spécifiez une heure de début et une heure de fin d'événement qui couvrent un intervalle de 20 minutes, les événements sont relus à partir de la première minute de cet intervalle de 20 minutes. Ensuite, les événements sont relus à partir de la deuxième minute. Vous pouvez utiliser l'opération `DescribeReplay` de l'API EventBridge pour déterminer la progression d'une relecture. `EventLastReplayedTime` renvoie l'horodatage du dernier événement relu.

Les événements sont relus en se basant sur le nombre maximal de transactions `PutEvents` par seconde pour le compte AWS, mais séparément de celui-ci. Vous pouvez demander une augmentation du nombre maximal de PutEvents. Pour plus d'informations, consultez Quotas Amazon EventBridge.

> **Note**
>
> Vous pouvez effectuer un maximum de 10 relectures simultanées actives par compte et par région AWS.

Pour démarrer une relecture d'événement

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation de gauche, choisissez Relectures.
3. Choisissez Démarrer une nouvelle lecture.
4. Entrez un Nom pour la relecture et éventuellement une Description.
5. Pour Source, sélectionnez l'archive à partir de laquelle relire les événements.

6. Pour la destination, vous ne pouvez relire les événements que sur le même bus d'événements qui les a émis.
7. Pour Spécifier des règles, effectuez l'une des actions suivantes :
   - Choisissez Toutes les règles pour relire les événements selon toutes les règles.
   - Choisissez Spécifier des règles, puis sélectionnez la ou les règles selon lesquelles relire les événements.
8. Sous Période de relecture, spécifiez la Date, l'Heure et le Fuseau horaire pour Heure de début et Heure de fin. Seuls les événements qui se sont produits entre l'Heure de début et l'Heure de fin sont relus.
9. Sélectionnez Lancer la relecture.

Lorsque les événements de l'archive sont relus, le statut de la relecture indique Terminé.

Si vous démarrez une relecture et que vous souhaitez l'interrompre, vous pouvez l'annuler tant que le statut indique Démarrage ou En cours d'exécution.

Pour annuler une relecture

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation de gauche, choisissez Relectures.
3. Choisissez la relecture à annuler.
4. Choisissez Cancel (Annuler).

Amazon EventBridge Pipes

Amazon EventBridge Pipes connecte les sources aux cibles. Les tubes sont destinés aux point-to-point intégrations entre les sources et les cibles prises en charge, avec la prise en charge des transformations avancées et de l'enrichissement. Ils réduisent le besoin de connaissances spécialisées et de code d'intégration lors du développement d'architectures pilotées par les événements, ce qui favorise la cohérence dans les applications de votre entreprise. Pour configurer un canal, vous devez choisir la source, ajouter un filtrage facultatif, définir un enrichissement facultatif et choisir la cible pour les données d'événement.

> **Note**
>
> Vous pouvez également router les événements en utilisant des bus d'événements. Les bus d'événements sont parfaitement adaptés au many-to-many routage d'événements entre des services pilotés par des événements. Pour plus d'informations, consultez ???.

Comment fonctionnent EventBridge les tuyaux

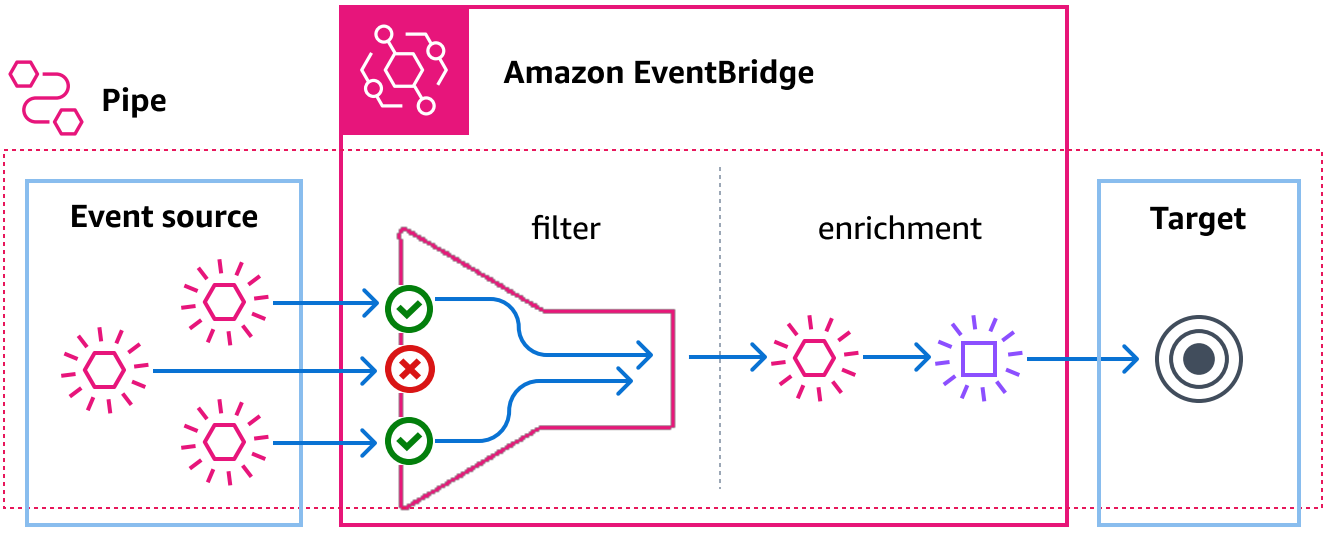
De manière générale, voici comment fonctionne EventBridge Pipes :

1. Vous créez un canal dans votre compte. Cela consiste notamment à :
   - Spécifier l'une des sources d'événements prises en charge dont votre canal recevra les événements.
   - Vous pouvez éventuellement configurer un filtre de sorte que le canal ne traite qu'un sous-ensemble des événements qu'il reçoit de la source.
   - Vous pouvez éventuellement configurer une étape d'enrichissement qui améliore les données d'événements avant de les envoyer à la cible.
   - Spécifier l'une des cibles prises en charge à laquelle votre canal enverra les événements.
2. La source d'événements commence à envoyer les événements au canal, qui les traite avant de les envoyer à la cible.
   - Si vous avez configuré un filtre, le canal évalue les événements et ne les envoie à la cible que s'ils correspondent à ce filtre.

     Seuls les événements qui correspondent au filtre vous sont facturés.

- Si vous avez configuré un enrichissement, le canal exécute cet enrichissement sur les événements avant de les envoyer à la cible.

  Si les événements se présentent sous forme de lot, l'enrichissement conserve l'ordre des événements dans le lot.



Par exemple, un canal pourrait être utilisé pour créer un système de commerce électronique. Supposez que vous disposez d'une API qui contient des informations sur les clients, telles que les adresses de livraison.

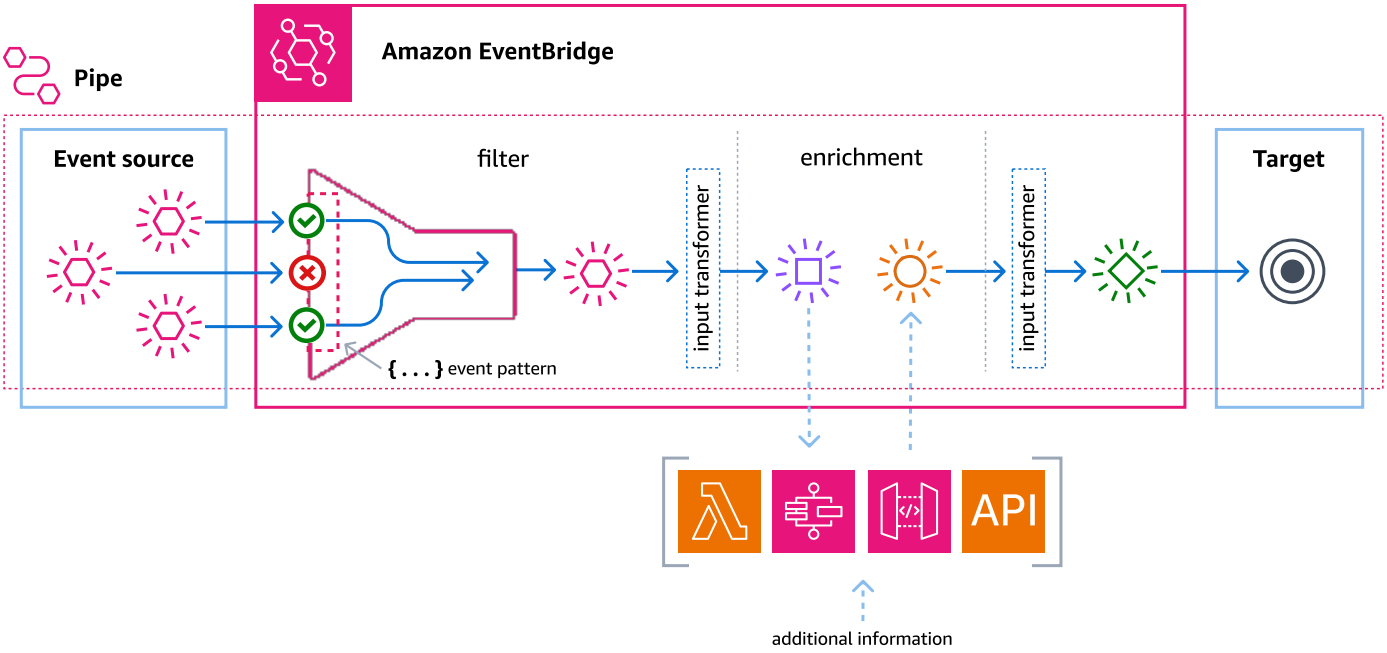
1. Vous créez alors un canal avec les éléments suivants :
   - Une file d'attente de messages de réception de commande Amazon SQS faisant office de source d'événements.
   - Une destination EventBridge d'API en tant qu'enrichissement
   - Une machine à AWS Step Functions états comme cible
2. Ensuite, lorsqu'un message de réception de commande Amazon SQS apparaît dans la file d'attente, il est envoyé au canal.
3. Le canal envoie ensuite ces données à l' EventBridge API Destination Enrichment, qui renvoie les informations client relatives à cette commande.
4. Enfin, le canal envoie les données enrichies à la machine AWS Step Functions d'état, qui traite la commande.

EventBridge Concepts de tuyaux

Voici un aperçu des composants de base de EventBridge Pipes.

Barre verticale

Un canal route les événements d'une source unique vers une cible unique. Il offre également la possibilité d'effectuer un filtrage sur des événements spécifiques et d'enrichir leurs données avant d'être envoyées à la cible.



Source

EventBridge Pipes reçoit des données d'événements provenant de diverses sources, applique des filtres facultatifs et un enrichissement à ces données, puis les envoie à une cible. Si une source impose un ordre aux événements envoyés aux canaux, cet ordre est maintenu tout au long du processus menant à la cible.

Pour plus d'informations sur les sources, consultez ???.

Filtres

Un canal peut filtrer les événements d'une source déterminée pour ne traiter qu'un sous-ensemble de ces événements. Pour configurer le filtrage d'un canal, vous devez définir le modèle d'événement que le canal utilisera pour déterminer les événements à envoyer à la cible.

Seuls les événements qui correspondent au filtre vous sont facturés.

Pour plus d'informations, consultez ???.

Enrichissement

Avec l'étape d'enrichissement de EventBridge Pipes, vous pouvez améliorer les données de la source avant de les envoyer à la cible. Par exemple, vous pouvez recevoir des événements de type Ticket créé qui n'incluent pas l'ensemble des données de ticket. Grâce à l'enrichissement, vous pouvez demander à une fonction Lambda d'appeler l'API `get-ticket` pour obtenir les détails complets du ticket. Le canal peut ensuite envoyer ces informations à une cible.

Pour plus d'informations sur l'enrichissement des données d'événements, consultez ???.

Cible

Une fois que les données de l'événement ont été filtrées et enrichies, vous pouvez spécifier le canal pour les envoyer à une cible spécifique, telle qu'un flux Amazon Kinesis ou un groupe de CloudWatch logs Amazon. Pour obtenir la liste des cibles disponibles, consultez ???.

Vous pouvez transformer les données après avoir été améliorées et avant d'avoir été envoyées à la cible par le canal. Pour plus d'informations, consultez ???.

Plusieurs canaux, chacun avec une source différente, peuvent envoyer les événements à la même cible.

Vous pouvez également utiliser des canaux et des bus d'événements conjointement de façon à envoyer les événements à plusieurs cibles. Un cas d'utilisation courant est la création d'un canal avec un bus d'événements en tant que cible ; le canal envoie les événements au bus d'événements, qui les transmet ensuite à plusieurs cibles. Par exemple, vous pouvez créer un canal dont la source est un flux DynamoDB, ainsi qu'un bus d'événements faisant office de cible. Le canal reçoit les événements du flux DynamoDB et les envoie au bus d'événements, qui les envoie ensuite à plusieurs cibles en fonction des règles que vous avez spécifiées pour le bus d'événements.

Autorisations pour Amazon EventBridge Pipes

Lorsque vous configurez un canal, vous pouvez utiliser un rôle d'exécution existant ou demander à EventBridge d'en créer un pour vous avec les autorisations nécessaires. Les autorisations requises

par EventBridge Pipes varient en fonction du type de source et sont répertoriées ci-dessous. Si vous configurez votre propre rôle d'exécution, vous devez ajouter ces autorisations vous-même.

> **Note**
>
> Si vous n'êtes pas sûr des autorisations précises requises pour accéder à la source, utilisez la console EventBridge Pipes pour créer un nouveau rôle, puis examinez les actions répertoriées dans la politique.

Rubriques

Autorisations du rôle d'exécution DynamoDB

Pour DynamoDB Streams, EventBridge Pipes exige les autorisations suivantes pour gérer les ressources liées à votre flux de données DynamoDB.

- `dynamodb:DescribeStream`
- `dynamodb:GetRecords`
- `dynamodb:GetShardIterator`
- `dynamodb:ListStreams`

Pour envoyer des enregistrements de lots ayant échoué à la file d'attente de lettres mortes, le rôle d'exécution de votre canal doit disposer de l'autorisation suivante :

- `sqs:SendMessage`

Autorisations du rôle d'exécution Kinesis

Pour Kinesis, EventBridge Pipes exige les autorisations suivantes pour gérer les ressources liées à votre flux de données Kinesis.

- kinesis:DescribeStream
- kinesis:DescribeStreamSummary
- kinesis:GetRecords
- kinesis:GetShardIterator
- kinesis:ListShards
- kinesis:ListStreams
- kinesis:SubscribeToShard

Pour envoyer des enregistrements de lots ayant échoué à la file d'attente de lettres mortes, le rôle d'exécution de votre canal doit disposer de l'autorisation suivante :

- sqs:SendMessage

Autorisations du rôle d'exécution Amazon MQ

Pour Amazon MQ, EventBridge Pipes exige les autorisations suivantes pour gérer les ressources liées à votre agent de messages Amazon MQ.

- mq:DescribeBroker
- secretsmanager:GetSecretValue
- ec2:CreateNetworkInterface
- ec2:DeleteNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeSecurityGroups
- ec2:DescribeSubnets
- ec2:DescribeVpcs
- logs:CreateLogGroup
- logs:CreateLogStream

- logs:PutLogEvents

Autorisations du rôle d'exécution Amazon MSK

Pour Amazon MSK, EventBridge exige les autorisations suivantes pour gérer les ressources liées à votre rubrique Amazon MSK.

> **Note**
>
> Si vous utilisez l'authentification basée sur les rôles IAM, votre rôle d'exécution aura besoin des autorisations répertoriées dans ??? en plus de celles répertoriées ci-dessous.

- kafka:DescribeClusterV2
- kafka:GetBootstrapBrokers
- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets
- ec2:DescribeSecurityGroups
- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Autorisations de rôle d'exécution Apache Kafka autogéré

Pour Apache Kafka autogéré, EventBridge exige les autorisations suivantes pour gérer les ressources liées à votre flux Apache Kafka autogéré.

Autorisations nécessaires

Pour pouvoir créer et stocker des journaux dans un groupe de journaux dans Amazon CloudWatch Logs, votre canal doit disposer des autorisations suivantes dans son rôle d'exécution :

- logs:CreateLogGroup
- logs:CreateLogStream
- logs:PutLogEvents

Autorisations facultatives

Votre canal peut également nécessiter ces autorisations pour :

- Décrivez votre secret Secrets Manager.
- Accédez à votre clé gérée par le client AWS Key Management Service (AWS KMS).
- Accédez à votre Amazon VPC.

Secrets Manager et autorisations AWS KMS

Selon le type de contrôle d'accès que vous configurez pour vos agents Apache Kafka, votre canal peut avoir besoin d'une autorisation pour accéder à votre secret Secrets Manager ou pour déchiffrer votre clé gérée par le client AWS KMS. Afin d'accéder à ces ressources, le rôle d'exécution de votre fonction doit disposer des autorisations suivantes :

- secretsmanager:GetSecretValue
- kms:Decrypt

Autorisations VPC

Si seuls des utilisateurs au sein d'un VPC peuvent accéder à votre cluster Apache Kafka autogéré, votre canal doit être autorisé à accéder à vos ressources Amazon VPC. Ces ressources incluent les sous-réseaux, groupes de sécurité et interfaces réseau de votre VPC. Afin d'accéder à ces ressources, le rôle d'exécution de votre canal doit disposer des autorisations suivantes :

- ec2:CreateNetworkInterface
- ec2:DescribeNetworkInterfaces
- ec2:DescribeVpcs
- ec2:DeleteNetworkInterface
- ec2:DescribeSubnets

- `ec2:DescribeSecurityGroups`

Autorisations du rôle d'exécution Amazon SQS

Pour Amazon SQS, EventBridge exige les autorisations suivantes pour gérer les ressources liées à votre file d'attente Amazon SQS.

- `sqs:ReceiveMessage`
- `sqs:DeleteMessage`
- `sqs:GetQueueAttributes`

Enrichissement et autorisations cibles

EventBridge Pipes a besoin d'une autorisation appropriée pour effectuer des appels d'API sur les ressources que vous possédez. EventBridge Pipes utilise le rôle IAM que vous spécifiez sur le canal pour l'enrichissement et les appels cibles à l'aide du principal IAM `pipes.amazonaws.com`.

Création d'un EventBridge canal Amazon

EventBridge Pipes vous permet de créer des point-to-point intégrations entre les sources et les cibles, notamment des transformations et des enrichissements d'événements avancés. Pour créer un EventBridge canal, vous devez suivre les étapes suivantes :

1. ???
2. ???
3. ???
4. ???
5. ???

Pour plus d'informations sur la création d'un canal à l'aide de la AWS CLI, voir create-pipe dans le manuel de référence des commandes de la AWS CLI.

Spécification d’une source

Pour commencer, spécifiez la source à partir de laquelle vous souhaitez que le canal reçoive les événements.

Pour spécifier une source de canal à l’aide de la console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Pipelines.
3. Choisissez Créer un pipeline.
4. Entrez un nom pour le canal.
5. (Facultatif) Ajoutez une description du canal.
6. Dans l’onglet Construire un pipeline, pour Source, choisissez le type de source que vous souhaitez spécifier pour ce canal, puis configurez la source.

   Les propriétés de configuration varient en fonction du type de source que vous choisissez :

   Confluent

   Pour configurer un flux Confluent Cloud en tant que source, à l'aide de la console

   1. Pour Source, choisissez Confluent Cloud.
   2. Pour Serveurs Bootstrap, entrez les adresses des paires `host:port` de vos agents.
   3. Pour Nom de la rubrique, entrez le nom de la rubrique à partir de laquelle le canal effectuera la lecture.
   4. (Facultatif) Pour VPC, choisissez un VPC. Ensuite, pour Sous-réseaux VPC, choisissez les sous-réseaux souhaités. Pour Groupes de sécurité VPC, choisissez les groupes de sécurité.
   5. Pour Authentification (facultatif), activez l'option Utiliser l'authentification et procédez comme suit :

      a. Pour Méthode d’authentification, choisissez le type d’authentification.

      b. Pour Clé secrète, choisissez la clé secrète.

      Pour plus d'informations, consultez la section Authentification auprès des ressources Confluent Cloud dans la documentation Confluent.
   6. (Facultatif) Pour Paramètre supplémentaire - facultatif, procédez comme suit :

      a. Pour Position de départ, choisissez l’une des valeurs suivantes :

      - Dernier : commencez à lire le flux par l’enregistrement le plus récent de la partition.
      - Trim horizon : commencez à lire le flux par le dernier enregistrement non découpé de la partition. Il s’agit de l’enregistrement le plus ancien de la partition.

b. Pour Taille du lot - facultatif, entrez un nombre maximal d'enregistrements par lot. La valeur par défaut est 100.

c. Pour Fenêtre du lot - facultatif, entrez un nombre maximal de secondes pour collecter les enregistrements avant de continuer.

DynamoDB

1. Pour Source, choisissez DynamoDB.
2. Pour Flux DynamoDB, choisissez le flux que vous souhaitez utiliser en tant que source.
3. Pour Position de départ, choisissez l'une des valeurs suivantes :
   - Dernier : commencez à lire le flux par l'enregistrement le plus récent de la partition.
   - Trim horizon : commencez à lire le flux par le dernier enregistrement non découpé de la partition. Il s'agit de l'enregistrement le plus ancien de la partition.
4. (Facultatif) Pour Paramètre supplémentaire - facultatif, procédez comme suit :
   a. Pour Taille du lot - facultatif, entrez un nombre maximal d'enregistrements par lot. La valeur par défaut est 100.
   b. Pour Fenêtre du lot - facultatif, entrez un nombre maximal de secondes pour collecter les enregistrements avant de continuer.
   c. Pour Lots simultanés par partition - facultatif, entrez le nombre de lots issus de la même partition qui peuvent être lus simultanément.
   d. Pour En cas de défaillance partielle d'un lot, choisissez ce qui suit :
      - AUTOMATIC_BISECT : divisez chaque lot en deux et effectuez une nouvelle tentative avec chaque moitié jusqu'à ce que tous les enregistrements soient traités ou qu'il reste un message d'échec dans le lot.

> **Note**
>
> Si vous ne choisissez pas AUTOMATIC_BISECT, vous pouvez renvoyer certains enregistrements ayant échoué et uniquement ceux ayant fait l'objet d'une nouvelle tentative.

Kinesis

Pour configurer une source Kinesis à l'aide de la console

1. Pour Source, choisissez Kinesis.
2. Pour Flux Kinesis, choisissez le flux que vous souhaitez utiliser en tant que source.
3. Pour Position de départ, choisissez l'une des valeurs suivantes :
   - Dernier : commencez à lire le flux par l'enregistrement le plus récent de la partition.
   - Trim horizon : commencez à lire le flux par le dernier enregistrement non découpé de la partition. Il s'agit de l'enregistrement le plus ancien de la partition.
   - À l'horodatage : commencez à lire le flux à partir d'une heure spécifiée. Sous Horodatage, entrez une date et une heure au format AAAA/MM/JJ et hh:mm:ss.
4. (Facultatif) Pour Paramètre supplémentaire - facultatif, procédez comme suit :
   a. Pour Taille du lot - facultatif, entrez un nombre maximal d'enregistrements par lot. La valeur par défaut est 100.
   b. (Facultatif) Pour Fenêtre du lot - facultatif, entrez un nombre maximal de secondes pour collecter les enregistrements avant de continuer.
   c. Pour Lots simultanés par partition - facultatif, entrez le nombre de lots issus de la même partition qui peuvent être lus simultanément.
   d. Pour En cas de défaillance partielle d'un lot, choisissez ce qui suit :
      - AUTOMATIC_BISECT : divisez chaque lot en deux et effectuez une nouvelle tentative avec chaque moitié jusqu'à ce que tous les enregistrements soient traités ou qu'il reste un message d'échec dans le lot.

> **Note**
>
> Si vous ne choisissez pas AUTOMATIC_BISECT, vous pouvez renvoyer certains enregistrements ayant échoué et uniquement ceux ayant fait l'objet d'une nouvelle tentative.

Amazon MQ

Pour configurer une source Amazon MQ à l’aide de la console

1. Pour Source, choisissez Amazon MQ.
2. Pour Courtier Amazon MQ, choisissez le flux que vous souhaitez utiliser en tant que source.
3. Pour Nom de la file, entrez le nom de la file d’attente à partir de laquelle le canal effectuera la lecture.
4. Pour Méthode d’authentification, choisissez BASIC_AUTH.
5. Pour Clé secrète, choisissez la clé secrète.
6. (Facultatif) Pour Paramètre supplémentaire - facultatif, procédez comme suit :
   a. Pour Taille du lot - facultatif, entrez un nombre maximal de messages par lot. La valeur par défaut est 100.
   b. Pour Fenêtre du lot - facultatif, entrez un nombre maximal de secondes pour collecter les enregistrements avant de continuer.

Amazon MSK

Pour configurer une source Amazon MSK à l’aide de la console

1. Pour Source, choisissez Amazon MSK.
2. Pour Cluster Amazon MSK, choisissez le cluster à utiliser.
3. Pour Nom de la rubrique, entrez le nom de la rubrique à partir de laquelle le canal effectuera la lecture.
4. (Facultatif) Pour ID du groupe de consommateurs - facultatif, entrez l’identifiant du groupe de consommateurs que le canal doit rejoindre.
5. (Facultatif) Pour Authentification - facultatif, activez Utiliser l’authentification et procédez comme suit :
   a. Pour Méthode d’authentification, choisissez le type souhaité.
   b. Pour Clé secrète, choisissez la clé secrète.
6. (Facultatif) Pour Paramètre supplémentaire - facultatif, procédez comme suit :

a. Pour Taille du lot - facultatif, entrez un nombre maximal d'enregistrements par lot. La valeur par défaut est 100.

b. Pour Fenêtre du lot - facultatif, entrez un nombre maximal de secondes pour collecter les enregistrements avant de continuer.

c. Pour Position de départ, choisissez l'une des valeurs suivantes :

- Dernier : commencez à lire la rubrique par l'enregistrement le plus récent de la partition.
- Trim horizon : commencez à lire la rubrique par le dernier enregistrement non découpé de la partition. Il s'agit de l'enregistrement le plus ancien de la partition.

> **Note**
>
> Pour Apache Kafka, Trim horizon est identique à Premier.

Self managed Apache Kafka

Pour configurer une source Apache Kafka autogérée à l'aide de la console

1. Pour Source, choisissez Apache Kafka autogéré.
2. Pour Serveurs Bootstrap, entrez les adresses des paires `host:port` de vos agents.
3. Pour Nom de la rubrique, entrez le nom de la rubrique à partir de laquelle le canal effectuera la lecture.
4. (Facultatif) Pour VPC, choisissez un VPC. Ensuite, pour Sous-réseaux VPC, choisissez les sous-réseaux souhaités. Pour Groupes de sécurité VPC, choisissez les groupes de sécurité.
5. (Facultatif) Pour Authentification - facultatif, activez Utiliser l'authentification et procédez comme suit :
   a. Pour Méthode d'authentification, choisissez le type d'authentification.
   b. Pour Clé secrète, choisissez la clé secrète.
6. (Facultatif) Pour Paramètre supplémentaire - facultatif, procédez comme suit :
   a. Pour Position de départ, choisissez l'une des valeurs suivantes :
      - Dernier : commencez à lire le flux par l'enregistrement le plus récent de la partition.

- Trim horizon : commencez à lire le flux par le dernier enregistrement non découpé de la partition. Il s'agit de l'enregistrement le plus ancien de la partition.

b. Pour Taille du lot - facultatif, entrez un nombre maximal d'enregistrements par lot. La valeur par défaut est 100.

c. Pour Fenêtre du lot - facultatif, entrez un nombre maximal de secondes pour collecter les enregistrements avant de continuer.

Amazon SQS

Pour configurer une source Amazon SQS à l'aide de la console

1. Pour Source, choisissez SQS.
2. Pour File d'attente SQS, choisissez la file d'attente que vous souhaitez utiliser.
3. (Facultatif) Pour Paramètre supplémentaire - facultatif, procédez comme suit :
   a. Pour Taille du lot - facultatif, entrez un nombre maximal d'enregistrements par lot. La valeur par défaut est 100.
   b. Pour Fenêtre du lot - facultatif, entrez un nombre maximal de secondes pour collecter les enregistrements avant de continuer.

Configuration du filtrage des événements (facultatif)

Vous pouvez ajouter un filtrage à votre canal afin d'envoyer uniquement un sous-ensemble d'événements de votre source à la cible.

Pour configurer le filtrage à l'aide de la console

1. Choisissez Filtrage.
2. Sous Exemple d'événement - facultatif, vous verrez un exemple d'événement que vous pouvez utiliser pour créer votre modèle d'événement, ou vous pouvez entrer votre propre événement en choisissant Saisir mon propre.
3. Sous Modèle d'événement, entrez le modèle d'événement que vous souhaitez utiliser pour filtrer les événements. Pour plus d'informations sur la création de filtres, consultez???.

   L'exemple de modèle d'événement suivant envoie uniquement des événements dont le champ City contient la valeur Seattle.

```
{
  "data": {
    "City": ["Seattle"]
  }
}
```

À présent que les événements sont filtrés, vous pouvez ajouter un enrichissement facultatif et une cible pour le canal.

Définition de l’enrichissement des événements (facultatif)

Vous pouvez envoyer les données d'événement pour les enrichir à une fonction Lambda, à une machine à AWS Step Functions états, à Amazon API Gateway ou à une destination d'API.

Pour sélectionner l’enrichissement

1. Choisissez Enrichissement.
2. Sous Informations, pour Service, sélectionnez le service et les paramètres associés que vous souhaitez utiliser pour l’enrichissement.

Vous pouvez également transformer les données avant de les envoyer pour les améliorer.

(Facultatif) Pour définir le transformateur d’entrée

1. Choisissez Transformateur d’entrée d’enrichissement - facultatif.
2. Pour Exemples d’événements/charge utile d’événement, choisissez le type d’exemple d’événement.
3. Pour Transformateur, entrez la syntaxe du transformateur, par exemple `"Event happened at <$.detail.field>."`, où `<$.detail.field>` est une référence à un champ issu de l’exemple d’événement. Vous pouvez également double-cliquer sur un champ dans l’exemple d’événement pour l’ajouter au transformateur.
4. Pour Sortie, vérifiez que la sortie ressemble à ce que vous souhaitez.

À présent que les données ont été filtrées et améliorées, vous devez définir une cible à laquelle envoyer les données d’événement.

Configuration d’une cible

Pour configurer une cible

1. Choisissez Target.
2. Sous Informations, pour Service cible, choisissez la cible. Les champs qui s’affichent varient en fonction de la cible que vous choisissez. Entrez les informations spécifiques à ce type de cible, selon vos besoins.

Vous pouvez également transformer les données avant de les envoyer à la cible.

(Facultatif) Pour définir le transformateur d’entrée

1. Choisissez Transformateur d’entrée cible - facultatif.
2. Pour Exemples d’événements/charge utile d’événement, choisissez le type d’exemple d’événement.
3. Pour Transformateur, entrez la syntaxe du transformateur, par exemple `"Event happened at <$.detail.field>."`, où `<$.detail.field>` est une référence à un champ issu de l’exemple d’événement. Vous pouvez également double-cliquer sur un champ dans l’exemple d’événement pour l’ajouter au transformateur.
4. Pour Sortie, vérifiez que la sortie ressemble à ce que vous souhaitez.

À présent que le canal est configuré, assurez-vous que ses paramètres sont correctement configurés.

Configuration des paramètres de canal

Un canal est actif par défaut, mais vous pouvez le désactiver. Vous pouvez également spécifier les autorisations du canal, configurer la journalisation du canal et ajouter des balises.

Pour configurer les paramètres de canal

1. Cliquez sur l’onglet Réglages des pipelines.
2. Par défaut, les canaux récemment créés sont actifs dès leur création. Si vous souhaitez créer un canal inactif, sous Activation, pour Activer le pipeline, désactivez Actif.
3. Sous Autorisations, pour Rôle d’exécution, effectuez l’une des opérations suivantes :

a. Pour EventBridge créer un nouveau rôle d'exécution pour ce canal, choisissez Create a new role for this specific resource. Sous Nom du rôle, vous pouvez éventuellement modifier le nom du rôle.

b. Pour utiliser un rôle d’exécution existant, choisissez Utiliser un rôle existant. Sous Nom du rôle, choisissez le rôle.

4. (Facultatif) Si vous avez spécifié un DynamoDB flux Kinesis ou comme source de canal, vous pouvez configurer une politique de nouvelles tentatives et une file d'attente de lettres mortes (DLQ).

   Pour Stratégie de nouvelles tentatives et file d’attente de lettres mortes - facultatif, procédez comme suit :

   Sous Politique de nouvelles tentatives, procédez comme suit :

   a. Si vous souhaitez activer les politiques de nouvelles tentatives, activez Réessayer. Par défaut, aucune politique de nouvelle tentative n’est activée pour les canaux récemment créés.

   b. Pour Maximum age of event (Âge maximal de l'événement), saisissez une valeur comprise entre une minute (00:01) et 24 heures (24:00).

   c. Pour Retry attempts (Nouvelles tentatives), saisissez un nombre compris entre 0 et 185.

   d. Si vous souhaitez utiliser une file d’attente de lettres mortes (DLQ), activez File d’attente de lettres mortes, choisissez la méthode de votre choix et choisissez la file d’attente ou la rubrique que vous souhaitez utiliser. Par défaut, les canaux récemment créés n’utilisent pas de DLQ.

5. (Facultatif) Sous Journaux - facultatif, vous pouvez configurer la manière dont EventBridge Pipes envoie les informations de journalisation aux services pris en charge, notamment comment configurer ces journaux.

   Pour plus d’informations sur la journalisation des enregistrements de journaux, consultez ???.

   CloudWatch les journaux sont sélectionnés comme destination des journaux par défaut, tout comme le niveau du `ERROR` journal. Ainsi, par défaut, EventBridge Pipes crée un nouveau groupe de CloudWatch journaux auquel il envoie des enregistrements contenant le `ERROR` niveau de détail.

   Pour que EventBridge Pipes envoie des enregistrements de journal vers l'une des destinations de journal prises en charge, procédez comme suit :

a. Sous Journaux - facultatif, choisissez les destinations vers lesquelles vous souhaitez que les enregistrements de journaux soient livrés.

b. Pour Niveau du journal, choisissez le niveau d'information EventBridge à inclure dans les enregistrements du journal. Le niveau de journalisation `ERROR` est sélectionné par défaut.

   Pour plus d'informations, consultez ???.

c. Sélectionnez Inclure les données d'exécution si vous EventBridge souhaitez inclure les informations de charge utile des événements et les informations de demande et de réponse de service dans les enregistrements du journal.

   Pour plus d'informations, consultez ???.

d. Configurez chaque destination de journal que vous avez sélectionnée :

   Pour CloudWatch Logs les journaux, sous CloudWatch journaux, procédez comme suit :

   - Pour le groupe de CloudWatch journaux, choisissez de EventBridge créer un nouveau groupe de journaux, de sélectionner un groupe de journaux existant ou de spécifier l'ARN d'un groupe de journaux existant.
   - Pour les nouveaux groupes de journaux, modifiez le nom du groupe de journaux comme vous le souhaitez.

   CloudWatch logs est sélectionné par défaut.

   Pour les journaux de Firehose flux, sous journal de Firehose flux, sélectionnez le Firehose flux.

   Pour Amazon S3 les journaux, sous S3 logs, procédez comme suit :

   - Entrez le nom du compartiment à utiliser comme destination du journal.
   - Entrez l'ID de AWS compte du propriétaire du compartiment.
   - Entrez le texte de préfixe que vous souhaitez utiliser lorsque EventBridge crée des objets S3.

     Pour plus d'informations, consultez Organisation des objets à l'aide de préfixes dans le Guide de l'utilisateur Amazon Simple Storage Service .
   - Choisissez la manière dont vous EventBridge souhaitez formater les enregistrements du journal S3 :

- `json` : JSON
- `plain` : texte brut
- `w3c` : Format de fichier de journalisation étendu W3C

6. (Facultatif) Sous Balises - facultatif, choisissez Ajouter une nouvelle balise et entrez une ou plusieurs balises pour la règle. Pour plus d'informations, consultez ???.
7. Choisissez Créer un pipeline.

Validation des paramètres de configuration

Après la création d'un canal, EventBridge valide les paramètres de configuration suivants :

- Rôle IAM : étant donné que la source d'un canal ne peut pas être modifiée une fois le canal créé, EventBridge vérifie que le rôle IAM fourni peut accéder à la source.

> **Note**
>
> EventBridge n'effectue pas la même validation pour les enrichissements ou les cibles car ils peuvent être mis à jour après la création du canal.

- Traitement par lots : EventBridge vérifie que la taille du lot de la source ne dépasse pas la taille de lot maximale de la cible. Si tel est le cas, EventBridge nécessite une taille de lot inférieure. En outre, si une cible ne prend pas en charge le traitement par lots, vous ne pouvez pas configurer le traitement par lots EventBridge pour la source.
- Enrichissements : EventBridge confirme que la taille de lot pour les enrichissements d'API Gateway et de destination d'API est de 1, car seules les tailles de lot de 1 sont prises en charge.

Démarrage ou arrêt d'un canal

Par défaut, un canal est à l'état `Running` et traite les événements lors de sa création.

Si vous créez un canal avec des sources Amazon SQS, Kinesis ou DynamoDB, sa création peut généralement prendre une minute ou deux.

Si vous créez un canal avec des sources Amazon MSK, Apache Kafka autogéré ou Amazon MQ, sa création peut prendre jusqu'à dix minutes.

Pour créer un canal sans traiter les événements à l'aide de la console

- Désactivez le paramètre Activer le pipeline.

Pour créer un canal sans traiter les événements par programmation

- Dans votre appel d'API, définissez `DesiredState` sur `Stopped`.

Pour démarrer ou arrêter un canal existant à l'aide de la console

- Dans l'onglet Réglages des pipelines, sous Activation, pour Activer le pipeline, activez ou désactivez Actif.

Pour démarrer ou arrêter un canal existant par programmation

- Dans votre appel d'API, définissez le paramètre `DesiredState` sur `RUNNING` ou `STOPPED`.

Il peut y avoir un délai entre le moment où un canal est à l'état `STOPPED` et celui où il ne traite plus les événements :

- Pour Amazon SQS et les sources de flux, ce délai est généralement inférieur à deux minutes.
- Pour les sources Amazon MQ et Apache Kafka, ce délai peut atteindre quinze minutes.

Sources d'Amazon EventBridge Pipes

EventBridge Pipes reçoit des données d'événements provenant de diverses sources, applique des filtres et des enrichissements facultatifs à ces données et les envoie vers une destination.

Si une source impose un ordre aux événements envoyés à EventBridge Pipes, cet ordre est maintenu tout au long du processus jusqu'à la destination.

Les AWS services suivants peuvent être spécifiés comme sources pour EventBridge Pipes :

- Flux Amazon DynamoDB
- Flux Amazon Kinesis
- Agent Amazon MQ
- Flux Amazon MSK

- File d’attente Amazon SQS
- Stream Apache Kafka

  Lorsque vous spécifiez un flux Apache Kafka comme source de canal, vous pouvez spécifier un flux Apache Kafka que vous gérez vous-même ou un flux géré par un fournisseur tiers tel que :

  - Confluent Cloud
  - CloudKarafka
  - Redpanda

Flux Amazon DynamoDB en tant que source

Vous pouvez utiliser EventBridge Pipes pour recevoir des enregistrements dans un flux DynamoDB. Ensuite, vous pouvez éventuellement filtrer ou améliorer ces enregistrements avant de les envoyer à une cible pour être traités. Vous pouvez choisir des paramètres spécifiques à Amazon DynamoDB Streams lors de la configuration du canal. EventBridge Pipes conserve l'ordre des enregistrements du flux de données lors de l'envoi de ces données vers la destination.

> **Important**
>
> La désactivation d'un flux DynamoDB qui est la source d'un canal rend ce canal inutilisable, même si vous réactivez le flux par la suite. Cela se produit pour les raisons suivantes :
>
> - Vous ne pouvez pas arrêter, démarrer ou mettre à jour un canal dont la source est désactivée.
> - Vous ne pouvez pas mettre à jour un canal avec une nouvelle source après sa création. Lorsque vous réactivez un flux DynamoDB, un nouvel Amazon Resource Name (ARN) lui est affecté et il n'est plus associé à votre canal.
>
> Si vous réactivez le flux DynamoDB, vous devrez créer un nouveau canal à l'aide du nouvel ARN du flux.

Exemple d'évènement

L'exemple d'événement suivant montre les informations reçues par le canal. Vous pouvez utiliser cet événement pour créer et filtrer vos modèles d'événements, ou pour définir la transformation

d'entrée. Tous les champs ne peuvent pas être filtrés. Pour plus d'informations sur les champs que vous pouvez filtrer, consultez ???.


```
[
  {
    "eventID": "1",
    "eventVersion": "1.0",
    "dynamodb": {
      "Keys": {
        "Id": {
          "N": "101"
        }
      },
      "NewImage": {
        "Message": {
          "S": "New item!"
        },
        "Id": {
          "N": "101"
        }
      },
      "StreamViewType": "NEW_AND_OLD_IMAGES",
      "SequenceNumber": "111",
      "SizeBytes": 26
    },
    "awsRegion": "us-west-2",
    "eventName": "INSERT",
    "eventSourceARN": "arn:aws:dynamodb:us-east-1:111122223333:table/EventSourceTable",
    "eventSource": "aws:dynamodb"
  },
  {
    "eventID": "2",
    "eventVersion": "1.0",
    "dynamodb": {
      "OldImage": {
        "Message": {
          "S": "New item!"
        },
        "Id": {
          "N": "101"
        }
      },
      "SequenceNumber": "222",
```

```
      "Keys": {
        "Id": {
          "N": "101"
        }
      },
      "SizeBytes": 59,
      "NewImage": {
        "Message": {
          "S": "This item has changed"
        },
        "Id": {
          "N": "101"
        }
      },
      "StreamViewType": "NEW_AND_OLD_IMAGES"
    },
    "awsRegion": "us-west-2",
    "eventName": "MODIFY",
    "eventSourceARN": "arn:aws:dynamodb:us-east-1:111122223333:table/EventSourceTable",
    "eventSource": "aws:dynamodb"
  }
]
```

Flux d'interrogation et de mise en lots

EventBridge interroge les partitions de votre flux DynamoDB en quête d'enregistrements à une fréquence de base de quatre fois par seconde. Lorsque des enregistrements sont disponibles, EventBridge traite l'événement et attend le résultat. Si le traitement réussit, EventBridge reprend l'interrogation jusqu'à ce qu'il reçoive plus d'enregistrements.

Par défaut, EventBridge invoque votre canal dès que des enregistrements sont disponibles. Si le lot qu'EventBridge lit à partir de la source ne comprend qu'un seul enregistrement, un seul événement est traité. Pour éviter de traiter un petit nombre d'enregistrements, vous pouvez indiquer au canal de les mettre en mémoire tampon pendant cinq minutes maximum en configurant une fenêtre de traitement par lots. Avant de traiter les événements, EventBridge continue de lire les enregistrements de la source jusqu'à ce qu'il ait rassemblé un lot complet, que la fenêtre de traitement par lot expire ou que le lot atteigne la limite de charge utile de 6 Mo.

Vous pouvez également augmenter la simultanéité en traitant plusieurs lots de chaque partition en parallèle. EventBridge peut traiter simultanément jusqu'à 10 lots dans chaque partition. Si vous

augmentez le nombre de lots simultanés par partition, EventBridge assure toujours un traitement dans l'ordre au niveau de la clé de partition.

Configurez le paramètre `ParallelizationFactor` pour traiter une partition d'un flux de données Kinesis ou DynamoDB avec plusieurs exécutions de canal simultanément. Vous pouvez spécifier le nombre de lots simultanés qu'EventBridge interroge à partir d'une partition via un facteur de parallélisation compris entre 1 (par défaut) et 10. Par exemple, lorsque vous définissez `ParallelizationFactor` sur 2, vous pouvez avoir jusqu'à 200 exécutions de canal EventBridge simultanées pour traiter 100 partitions de données Kinesis. Cela permet d'augmenter le débit de traitement quand le volume de données est volatil et que la valeur du paramètre `IteratorAge` est élevée. Notez que le facteur de parallélisation ne fonctionnera pas si vous utilisez l'agrégation Kinesis.

Position de départ du sondage et du stream

Sachez que l'interrogation des sources de flux lors de la création et des mises à jour du canal est finalement cohérente.

- Lors de la création du canal, le démarrage de l'interrogation des événements depuis le flux peut prendre plusieurs minutes.
- Lors des mises à jour du canal dans la configuration d'interrogation des sources, l'arrêt et le redémarrage de l'interrogation des événements depuis le flux peuvent prendre plusieurs minutes.

Cela signifie que si vous spécifiez `LATEST` comme position de départ du flux, le canal peut manquer des événements envoyés lors de la création ou des mises à jour du canal. Pour vous assurer de ne manquer aucun événement, définissez la position de départ du flux sur `TRIM_HORIZON`.

Signalement des échecs d'éléments par lot

Lorsqu'EventBridge utilise et traite des données de streaming à partir d'une source, par défaut, il effectue un point de contrôle sur le numéro de séquence le plus élevé d'un lot, mais uniquement si le lot est un succès complet. Pour éviter de retraiter les messages dont le traitement a réussi dans un lot ayant échoué, vous pouvez configurer votre enrichissement ou votre cible de sorte à renvoyer un objet en indiquant les messages qui ont réussi et ceux qui ont échoué. C'est ce que l'on appelle une réponse partielle de lot.

Pour de plus amples informations, veuillez consulter ???.

Conditions de réussite et d'échec

Si vous renvoyez l'un des éléments suivants, EventBridge traite un lot comme un succès complet :

- Une liste `batchItemFailure` vide
- Une liste `batchItemFailure` nulle
- Une `EventResponse` vide
- Une `EventResponse` nulle

Si vous renvoyez l'un des éléments suivants, EventBridge traite un lot comme un échec complet :

- Une chaîne `itemIdentifier` vide
- Un `itemIdentifier` nul
- Un `itemIdentifier` avec un nom de clé incorrect

EventBridge retente les échecs en fonction de votre politique de nouvelle tentative.

Flux Amazon Kinesis en tant que source

Vous pouvez utiliser EventBridge Pipes pour recevoir des enregistrements dans un flux de données Kinesis. Ensuite, vous pouvez éventuellement filtrer ou améliorer ces enregistrements avant de les envoyer à l'une des destinations disponibles pour être traités. Vous pouvez choisir des paramètres spécifiques à Kinesis lors de la configuration du canal. EventBridge Pipes conserve l'ordre des enregistrements du flux de données lors de l'envoi de ces données vers la destination.

Un flux de données Kinesis est un ensemble de partitions. Chaque partition contient une séquence d'enregistrements de données. Un consommateur est une application qui traite les données d'un flux de données Kinesis. Vous pouvez mapper un canal EventBridge à un consommateur à débit partagé (itérateur standard) ou à un consommateur à débit dédié avec diffusion améliorée.

Pour les itérateurs standard, EventBridge utilise le protocole HTTP pour interroger chaque partition de votre flux Kinesis à la recherche d'enregistrements. Le canal partage le débit de lecture avec d'autres consommateurs de la partition.

Pour réduire la latence et optimiser le débit en lecture, vous pouvez créer un consommateur de flux de données avec diffusion améliorée. Les consommateurs de flux obtiennent une connexion dédiée pour chaque partition qui n'a pas d'impact sur les autres applications lisant sur le flux. Le débit dédié

peut aider si vous avez de nombreuses applications lisant les mêmes données, ou si vous retraitez un flux avec de gros enregistrements. Kinesis envoie des enregistrements à EventBridge via HTTP/2. Pour en savoir plus sur les flux de données Kinesis, consultez Lecture de données à partir d'Amazon Kinesis Data Streams.

Exemple d'évènement

L'exemple d'événement suivant montre les informations reçues par le canal. Vous pouvez utiliser cet événement pour créer et filtrer vos modèles d'événements, ou pour définir la transformation d'entrée. Tous les champs ne peuvent pas être filtrés. Pour plus d'informations sur les champs que vous pouvez filtrer, consultez ???.


```
[
  {
    "kinesisSchemaVersion": "1.0",
    "partitionKey": "1",
    "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
    "data": "SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "approximateArrivalTimestamp": 1545084650.987
    "eventSource": "aws:kinesis",
    "eventVersion": "1.0",
    "eventID":
 "shardId-000000000006:49590338271490256608559692538361571095921575989136588898",
    "eventName": "aws:kinesis:record",
    "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
    "awsRegion": "us-east-2",
    "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
  },
  {
    "kinesisSchemaVersion": "1.0",
    "partitionKey": "1",
    "sequenceNumber": "49590338271490256608559692540925702759324208523137515618",
    "data": "VGhpcyBpcyBvbmx5IGEgdGVzdC4=",
    "approximateArrivalTimestamp": 1545084711.166
    "eventSource": "aws:kinesis",
    "eventVersion": "1.0",
    "eventID":
 "shardId-000000000006:49590338271490256608559692540925702759324208523137515618",
    "eventName": "aws:kinesis:record",
    "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
    "awsRegion": "us-east-2",
    "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
  }
```

```
]
```

Flux d'interrogation et de mise en lots

EventBridge interroge les partitions de votre flux Kinesis en quête d'enregistrements à une fréquence de base de quatre fois par seconde. Lorsque des enregistrements sont disponibles, EventBridge traite l'événement et attend le résultat. Si le traitement réussit, EventBridge reprend l'interrogation jusqu'à ce qu'il reçoive plus d'enregistrements.

Par défaut, EventBridge invoque votre canal dès que des enregistrements sont disponibles. Si le lot qu'EventBridge lit à partir de la source ne comprend qu'un seul enregistrement, un seul événement est traité. Pour éviter de traiter un petit nombre d'enregistrements, vous pouvez indiquer au canal de les mettre en mémoire tampon pendant cinq minutes maximum en configurant une fenêtre de traitement par lots. Avant de traiter les événements, EventBridge continue de lire les enregistrements de la source jusqu'à ce qu'il ait rassemblé un lot complet, que la fenêtre de traitement par lot expire ou que le lot atteigne la limite de charge utile de 6 Mo.

Vous pouvez également augmenter la simultanéité en traitant plusieurs lots de chaque partition en parallèle. EventBridge peut traiter simultanément jusqu'à 10 lots dans chaque partition. Si vous augmentez le nombre de lots simultanés par partition, EventBridge assure toujours un traitement dans l'ordre au niveau de la clé de partition.

Configurez le paramètre `ParallelizationFactor` pour traiter une partition d'un flux de données Kinesis ou DynamoDB avec plusieurs exécutions de canal simultanément. Vous pouvez spécifier le nombre de lots simultanés qu'EventBridge interroge à partir d'une partition via un facteur de parallélisation compris entre 1 (par défaut) et 10. Par exemple, lorsque vous définissez `ParallelizationFactor` sur 2, vous pouvez avoir jusqu'à 200 exécutions de canal EventBridge simultanées pour traiter 100 partitions de données Kinesis. Cela permet d'augmenter le débit de traitement quand le volume de données est volatil et que la valeur du paramètre `IteratorAge` est élevée. Notez que le facteur de parallélisation ne fonctionnera pas si vous utilisez l'agrégation Kinesis.

Position de départ du sondage et du stream

Sachez que l'interrogation des sources de flux lors de la création et des mises à jour du canal est finalement cohérente.

- Lors de la création du canal, le démarrage de l'interrogation des événements depuis le flux peut prendre plusieurs minutes.

- Lors des mises à jour du canal dans la configuration d'interrogation des sources, l'arrêt et le redémarrage de l'interrogation des événements depuis le flux peuvent prendre plusieurs minutes.

Cela signifie que si vous spécifiez `LATEST` comme position de départ du flux, le canal peut manquer des événements envoyés lors de la création ou des mises à jour du canal. Pour vous assurer de ne manquer aucun événement, spécifiez la position de départ du flux comme `TRIM_HORIZON` ou `AT_TIMESTAMP`.

Signalement des échecs d'éléments par lot

Lorsqu'EventBridge utilise et traite des données de streaming à partir d'une source, par défaut, il effectue un point de contrôle sur le numéro de séquence le plus élevé d'un lot, mais uniquement si le lot est un succès complet. Pour éviter de retraiter les messages dont le traitement a réussi dans un lot ayant échoué, vous pouvez configurer votre enrichissement ou votre cible de sorte à renvoyer un objet en indiquant les messages qui ont réussi et ceux qui ont échoué. C'est ce que l'on appelle une réponse partielle de lot.

Pour de plus amples informations, veuillez consulter ???.

Conditions de réussite et d'échec

Si vous renvoyez l'un des éléments suivants, EventBridge traite un lot comme un succès complet :

- Une liste `batchItemFailure` vide
- Une liste `batchItemFailure` nulle
- Une `EventResponse` vide
- Une `EventResponse` nulle

Si vous renvoyez l'un des éléments suivants, EventBridge traite un lot comme un échec complet :

- Une chaîne `itemIdentifier` vide
- Un `itemIdentifier` nul
- Un `itemIdentifier` avec un nom de clé incorrect

EventBridge retente les échecs en fonction de votre politique de nouvelle tentative.

Agent de messages Amazon MQ en tant que source

Vous pouvez utiliser EventBridge Pipes pour recevoir des enregistrements d'un courtier de messages Amazon MQ. Ensuite, vous pouvez éventuellement filtrer ou améliorer ces enregistrements avant de les envoyer à l'une des destinations disponibles pour être traités. Il existe des paramètres spécifiques à Amazon MQ que vous pouvez choisir lors de la configuration d'un canal. EventBridge Pipes conserve l'ordre des enregistrements provenant du courtier de messages lors de l'envoi de ces données à la destination.

Amazon MQ est un service d'agent de messages géré pour Apache ActiveMQ et RabbitMQ. Un agent de messages permet à des applications et composants logiciels de communiquer à l'aide de différents langages de programmation, systèmes d'exploitation et autres protocoles de messagerie formels avec des rubriques ou des files d'attente comme destinations d'événements.

Amazon MQ peut également gérer des instances Amazon Elastic Compute Cloud (Amazon EC2) en votre nom en installant des agents ActiveMQ ou RabbitMQ. Une fois qu'un agent est installé, il fournit différentes topologies de réseau et d'autres besoins en infrastructure à vos instances.

La source Amazon MQ est soumise aux restrictions de configuration suivantes :

- Compte croisé : EventBridge ne prend pas en charge le traitement multicompte. Vous ne pouvez pas l'utiliser EventBridge pour traiter les enregistrements d'un courtier de messages Amazon MQ qui se trouve sur un autre AWS compte.
- Authentification — Pour ActiveMQ, seul ActiveMQ est pris en charge. SimpleAuthenticationPlugin Pour RabbitMQ, seule l'authentification PLAIN est prise en charge. Pour gérer les informations d'identification, utilisez AWS Secrets Manager. Pour plus d'informations sur l'authentification ActiveMQ, consultez Intégration des agents ActiveMQ avec LDAP dans le Guide du développeur Amazon MQ.
- Quota de connexion : les agents ont un nombre maximal de connexions autorisées pour chaque protocole de niveau filaire. Ce quota est basé sur le type d'instance de l'agent. Pour plus d'informations, consultez la section Agents de Quotas dans Amazon MQ dans le Guide du développeur Amazon MQ.
- Connectivité : vous pouvez créer des agents dans un cloud privé virtuel (VPC) public ou privé. Pour les VPC privés, votre canal a besoin d'un accès au VPC pour recevoir des messages.
- Destinations d'événements : seules les destinations de file d'attente sont prises en charge. Toutefois, vous pouvez utiliser une rubrique virtuelle, qui se comporte comme une rubrique en interne et comme une file d'attente en externe, lorsqu'elle interagit avec vos canaux. Pour plus

d’informations, consultez Destinations virtuelles (langue française non garantie) sur le site web d’Apache ActiveMQ et Hôtes virtuels (langue française non garantie) sur le site web de RabbitMQ.

- Topologie réseau : pour ActiveMQ, une seule instance ou un seul agent en veille est pris en charge pour un canal. Pour RabbitMQ, un seul agent d’instance ou un seul déploiement de cluster est pris en charge pour chaque canal. Les agents à instance unique nécessitent un point de terminaison de basculement. Pour plus d’informations sur ces modes de déploiement de l’agent, consultez Architecture d’agent ActiveMQ et Architecture d’agent RabbitMQ dans le Guide du développeur Amazon MQ.
- Protocoles : les protocoles pris en charge dépendent de l’intégration Amazon MQ que vous utilisez.
  - Pour les intégrations ActiveMQ EventBridge , utilise OpenWire le protocole /Java Message Service (JMS) pour consommer les messages. La consommation de messages n’est prise en charge par aucun autre protocole. EventBridge prend uniquement en charge les BytesMessageopérations TextMessageet dans le cadre du protocole JMS. Pour plus d'informations sur le OpenWire protocole, consultez OpenWirele site Web d'Apache ActiveMQ.
  - Pour les intégrations RabbitMQ, EventBridge utilise le protocole AMQP 0-9-1 pour consommer les messages. Aucun autre protocole n'est pris en charge pour la consommation de messages. Pour plus d'informations sur l'implémentation par RabbitMQ du protocole AMQP 0-9-1, consultez Guide de référence complet AMQP 0-9-1 sur le site web de RabbitMQ.

EventBridge prend automatiquement en charge les dernières versions d'ActiveMQ et de RabbitMQ prises en charge par Amazon MQ. Pour connaître les dernières versions prises en charge, consultez Notes de mise à jour Amazon MQ dans le Guide du développeur Amazon MQ.

> **Note**
>
> Par défaut, Amazon MQ comporte une fenêtre de maintenance hebdomadaire pour les agents. Pendant cette période, les agents ne sont pas disponibles. Pour les courtiers qui ne sont pas en veille, ils EventBridge ne traiteront pas les messages avant la fin de la fenêtre.

Exemples d’événements

L’exemple d’événement suivant montre les informations reçues par le canal. Vous pouvez utiliser cet événement pour créer et filtrer vos modèles d’événements, ou pour définir la transformation d’entrée. Tous les champs ne peuvent pas être filtrés. Pour plus d’informations sur les champs que vous pouvez filtrer, consultez ???.

ActiveMQ

```
[
  {
    "eventSource": "aws:amq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
    "messageType": "jms/text-message",
    "data": "QUJDOkFBQUE=",
    "connectionId": "myJMSCoID",
    "redelivered": false,
    "destination": {
      "physicalname": "testQueue"
    },
    "timestamp": 1598827811958,
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  },
  {
    "eventSource": "aws:amq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
    "messageType": "jms/bytes-message",
    "data": "3DTOOW7crj51prgVLQaGQ82S48k=",
    "connectionId": "myJMSCoID1",
    "persistent": false,
    "destination": {
      "physicalname": "testQueue"
    },
    "timestamp": 1598827811958,
    "brokerInTime": 1598827811958,
    "brokerOutTime": 1598827811959
  }
]
```

RabbitMQ

```
[
  {
```

```
    "eventSource": "aws:rmq",
    "eventSourceArn": "arn:aws:mq:us-
west-2:111122223333:broker:pizzaBroker:b-9bcfa592-423a-4942-879d-eb284b418fc8",
    "eventSourceKey": "pizzaQueue::/",
    "basicProperties": {
      "contentType": "text/plain",
      "contentEncoding": null,
      "headers": {
        "header1": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            49
          ]
        },
        "header2": {
          "bytes": [
            118,
            97,
            108,
            117,
            101,
            50
          ]
        },
        "numberInHeader": 10
      },
      "deliveryMode": 1,
      "priority": 34,
      "correlationId": null,
      "replyTo": null,
      "expiration": "60000",
      "messageId": null,
      "timestamp": "Jan 1, 1970, 12:33:41 AM",
      "type": null,
      "userId": "AIDACKCEVSQ6C2EXAMPLE",
      "appId": null,
      "clusterId": null,
      "bodySize": 80
    },
    "redelivered": false,
```

```
    "data": "eyJ0aW1lb3V0IjowLCJkYXRhIjoiQ1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
  }
]
```

Groupe de consommateurs

Pour interagir avec Amazon MQ, EventBridge créez un groupe de consommateurs capable de lire les avis de vos courtiers Amazon MQ. Le groupe de consommateurs est créé avec le même ID que l'UUID du canal.

Pour les sources Amazon MQ, regroupe EventBridge les enregistrements et les envoie à votre fonction en une seule charge utile. Pour contrôler le comportement, vous pouvez configurer la fenêtre de traitement par lots et la taille du lot. EventBridge extrait les messages jusqu'à ce que l'une des situations suivantes se produise :

- Les enregistrements traités atteignent la taille de charge utile maximale de 6 Mo.
- La fenêtre de traitement par lots expire.
- Le nombre d'enregistrements atteint la taille totale du lot.

EventBridge convertit votre lot en une seule charge utile, puis invoque votre fonction. Les messages ne sont ni conservés ni désérialisés. Au lieu de cela, le groupe de consommateurs les récupère sous la forme d'un objet BLOB d'octets. Ensuite, il les code en base64 dans une charge utile JSON. Si le canal renvoie une erreur pour l'un des messages d'un lot, EventBridge réessaie l'ensemble du lot de messages jusqu'à ce que le traitement aboutisse ou que les messages expirent.

Configuration réseau

Par défaut, les agents Amazon MQ sont créés avec l'indicateur `PubliclyAccessible` défini sur false. Ce n'est que lorsque `PubliclyAccessible` est défini sur true que l'agent reçoit une adresse IP publique. Pour un accès complet avec votre canal, votre agent doit utiliser un point de terminaison public ou fournir l'accès au VPC.

Si votre courtier Amazon MQ n'est pas accessible au public, EventBridge il doit avoir accès aux ressources Amazon Virtual Private Cloud (Amazon VPC) associées à votre courtier.

- Pour accéder au VPC de vos courtiers Amazon MQ EventBridge , vous pouvez utiliser un accès Internet sortant pour les sous-réseaux de votre source. Pour les sous-réseaux publics, il doit s'agir d'une passerelle NAT gérée. Pour les sous-réseaux privés, il peut s'agir d'une passerelle NAT

ou de votre propre NAT. Assurez-vous que le NAT possède une adresse IP publique et peut se connecter à Internet.

- EventBridge Pipes prend également en charge la diffusion d'événements AWS PrivateLink, ce qui vous permet d'envoyer des événements depuis une source d'événements située dans un Amazon Virtual Private Cloud (Amazon VPC) vers une cible Pipes sans passer par l'Internet public. Vous pouvez utiliser Pipes pour interroger depuis Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogéré et des Amazon MQ sources résidant dans un sous-réseau privé sans avoir à déployer une passerelle Internet, à configurer des règles de pare-feu ou à configurer des serveurs proxy.

  Pour configurer un point de terminaison VPC, consultez la section Créer un point de terminaison VPC dans le guide de l'utilisateur.AWS PrivateLink Pour le nom du service, sélectionnez`com.amazonaws.region.pipes-data`.

Vous devez configurer vos groupes de sécurité Amazon VPC avec les règles suivantes (au minimum) :

- Règles de trafic entrant : autorisez tout le trafic sur le port du courtier Amazon MQ pour les groupes de sécurité spécifiés pour votre source.
- Règles sortantes : autorisent tout le trafic sur le port 443 pour toutes les destinations. Autorisez tout le trafic sur le port du courtier Amazon MQ pour les groupes de sécurité spécifiés pour votre source.

  Les ports Broker incluent :

  - 9092 pour le texte en clair
  - 9094 pour TLS
  - 9096 pour SASL
  - 9098 pour IAM

> **Note**
>
> Votre configuration Amazon VPC est découvrable via l'API Amazon MQ. Vous n'avez pas besoin de la définir pendant la configuration.

Rubrique Amazon Managed Streaming for Apache Kafka en tant que source

Vous pouvez utiliser EventBridge Pipes pour recevoir des enregistrements d'une rubrique Amazon Managed Streaming for Apache Kafka (Amazon MSK). Vous pouvez éventuellement filtrer ou améliorer ces enregistrements avant de les envoyer à l'une des destinations disponibles pour être traités. Il existe des paramètres spécifiques à Amazon MSK que vous pouvez choisir lors de la configuration d'un canal. EventBridge Pipes conserve l'ordre des enregistrements provenant du courtier de messages lors de l'envoi de ces données à la destination.

Amazon MSK est un service entièrement géré qui vous permet de créer et d'exécuter des applications qui utilisent Apache Kafka pour traiter les données en streaming. Amazon MSK simplifie la configuration, la mise à l'échelle et la gestion des clusters exécutant Apache Kafka. Avec Amazon MSK, vous pouvez configurer votre application pour plusieurs zones de disponibilité et pour des raisons de sécurité avec AWS Identity and Access Management (IAM). Amazon MSK prend en charge plusieurs versions open source de Kafka.

Amazon MSK en tant que source fonctionne de la même manière qu'Amazon Simple Queue Service (Amazon SQS) ou Amazon Kinesis. EventBridgeinterroge en interne les nouveaux messages provenant de la source, puis invoque la cible de manière synchrone. EventBridge lit les messages par lots et les fournit à votre fonction sous forme de charge utile d'événements. La taille de lot maximale est configurable. (par défaut, 100 messages).

Pour les sources basées sur Apache Kafka, EventBridge prend en charge les paramètres de contrôle du traitement, tels que les fenêtres de traitement par lots et la taille des lots.

EventBridge lit les messages de manière séquentielle pour chaque partition. Après EventBridge avoir traité chaque lot, il valide les décalages des messages de ce lot. Si la cible du canal renvoie une erreur pour l'un des messages d'un lot, EventBridge réessaie l'ensemble du lot de messages jusqu'à ce que le traitement aboutisse ou que les messages expirent.

EventBridge envoie le lot de messages lorsqu'il invoque la cible. La charge utile d'un événement contient un tableau de messages. Chaque élément de tableau contient des détails de la rubrique Amazon MSK et un identifiant de partition, ainsi qu'un horodatage et un message codé en base 64.

Exemples d'événements

L'exemple d'événement suivant montre les informations reçues par le canal. Vous pouvez utiliser cet événement pour créer et filtrer vos modèles d'événements, ou pour définir la transformation

d’entrée. Tous les champs ne peuvent pas être filtrés. Pour plus d’informations sur les champs que vous pouvez filtrer, consultez ???.

```
[
  {
    "eventSource": "aws:kafka",
    "eventSourceArn": "arn:aws:kafka:sa-east-1:123456789012:cluster/
vpc-2priv-2pub/751d2973-a626-431c-9d4e-d7975eb44dd7-2",
    "eventSourceKey": "mytopic-0",
    "topic": "mytopic",
    "partition": "0",
    "offset": 15,
    "timestamp": 1545084650987,
    "timestampType": "CREATE_TIME",
    "key":"abcDEFghiJKLmnoPQRstuVWXyz1234==",
    "value":"SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "headers": [
      {
        "headerKey": [
          104,
          101,
          97,
          100,
          101,
          114,
          86,
          97,
          108,
          117,
          101
        ]
      }
    ]
  }
]
```

Position de départ du sondage et du stream

Sachez que l’interrogation des sources de flux lors de la création et des mises à jour du canal est finalement cohérente.

- Lors de la création du canal, le démarrage de l’interrogation des événements depuis le flux peut prendre plusieurs minutes.

- Lors des mises à jour du canal dans la configuration d'interrogation des sources, l'arrêt et le redémarrage de l'interrogation des événements depuis le flux peuvent prendre plusieurs minutes.

Cela signifie que si vous spécifiez `LATEST` comme position de départ du flux, le canal peut manquer des événements envoyés lors de la création ou des mises à jour du canal. Pour vous assurer de ne manquer aucun événement, définissez la position de départ du flux sur `TRIM_HORIZON`.

Authentification du cluster MSK

EventBridge a besoin d'une autorisation pour accéder au cluster Amazon MSK, récupérer des enregistrements et effectuer d'autres tâches. Amazon MSK prend en charge plusieurs options de contrôle de l'accès client au cluster MSK. Pour plus d'informations sur la méthode d'authentification utilisée, consultez ???.

Options d'accès au cluster

- Accès non authentifié
- Authentification SASL/SCRAM
- Authentification basée sur les rôles IAM
- Authentification TLS mutuelle
- Configuration du secret mTLS
- Comment EventBridge choisir un courtier Bootstrap

Accès non authentifié

Nous recommandons d'utiliser uniquement un accès non authentifié pour le développement. L'accès non authentifié ne fonctionnera que si l'authentification basée sur les rôles IAM est désactivée pour le cluster.

Authentification SASL/SCRAM

Amazon MSK prend en charge l'authentification SASL/SCRAM (Simple Authentication and Security Layer/Salted Challenge Response Authentication Mechanism) avec chiffrement du protocole TLS (Transport Layer Security). EventBridge Pour vous connecter au cluster, vous devez enregistrer les informations d'authentification (informations de connexion) dans un AWS Secrets Manager secret.

Pour plus d'informations sur l'utilisation de Secrets Manager, consultez Authentification par nom d'utilisateur et mot de passe avec AWS Secrets Manager dans le Guide du développeur Amazon Managed Streaming for Apache Kafka.

Notez qu'Amazon MSK ne prend pas en charge l'authentification SASL/PLAIN.

Authentification basée sur les rôles IAM

Vous pouvez utiliser IAM pour authentifier l'identité des clients qui se connectent au cluster MSK. Si l'authentification IAM est active sur votre cluster MSK et que vous ne fournissez pas de secret pour l'authentification, l'authentification IAM est EventBridge automatiquement utilisée par défaut. Pour créer et déployer des stratégies utilisateur ou basées sur des rôles IAM, utilisez la console IAM ou l'API. Pour plus d'informations, consultez Contrôle d'accès IAM dans le Guide du développeur Amazon Managed Streaming for Apache Kafka.

EventBridge Pour permettre de se connecter au cluster MSK, de lire des enregistrements et d'effectuer d'autres actions requises, ajoutez les autorisations suivantes au rôle d'exécution de vos canaux.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:DescribeGroup",
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeTopic",
                "kafka-cluster:ReadData",
                "kafka-cluster:DescribeClusterDynamicConfiguration"
            ],
            "Resource": [
                "arn:aws:kafka:region:account-id:cluster/cluster-name/cluster-uuid",
                "arn:aws:kafka:region:account-id:topic/cluster-name/cluster-uuid/topic-
name",
                "arn:aws:kafka:region:account-id:group/cluster-name/cluster-
uuid/consumer-group-id"
            ]
        }
    ]
}
```

Vous pouvez étendre ces autorisations à un cluster, une rubrique et un groupe spécifiques. Pour plus d'informations, consultez Actions Amazon MSK Kafka dans le Guide du développeur Amazon Managed Streaming for Apache Kafka.

Authentification TLS mutuelle

Mutual TLS (mTLS) fournit une authentification bidirectionnelle entre le client et le serveur. Le client envoie un certificat au serveur pour que le serveur vérifie le client, et le serveur envoie un certificat au client pour que le client vérifie le serveur.

Pour Amazon MSK, EventBridge agit en tant que client. Vous configurez un certificat client (sous forme de secret dans Secrets Manager) pour vous authentifier EventBridge auprès des courtiers de votre cluster MSK. Le certificat client doit être signé par une autorité de certification (CA) située dans le magasin d'approbations du serveur. Le cluster MSK envoie un certificat de serveur EventBridge pour authentifier les courtiers. EventBridge Le certificat du serveur doit être signé par une autorité de certification figurant dans le AWS trust store.

Amazon MSK ne prend pas en charge les certificats de serveur autosignés, car tous les courtiers d'Amazon MSK utilisent des certificats publics signés par les autorités de certification Amazon Trust Services, qui font EventBridge confiance par défaut.

Pour plus d'informations sur le protocole mTLS pour Amazon MSK, consultez Authentification Mutual TLS dans le Guide du développeur Amazon Managed Streaming for Apache Kafka.

Configuration du secret mTLS

Le secret CLIENT_CERTIFICATE_TLS_AUTH nécessite un champ de certificat et un champ de clé privée. Pour une clé privée chiffrée, le secret nécessite un mot de passe de clé privée. Le certificat et la clé privée doivent être au format PEM.

> **Note**
>
> EventBridge prend en charge les algorithmes de chiffrement à clé privée PBES1 (mais pas PBES2).

Le champ de certificat doit contenir une liste de certificats, commençant par le certificat client, suivi de tous les certificats intermédiaires et se terminant par le certificat racine. Chaque certificat doit commencer sur une nouvelle ligne avec la structure suivante :

```
-----BEGIN CERTIFICATE-----
        <certificate contents>
```

```
-----END CERTIFICATE-----
```

Secrets Manager prend en charge les secrets jusqu'à 65 536 octets, ce qui offre suffisamment d'espace pour de longues chaînes de certificats.

La clé privée doit être au format PKCS #8, avec la structure suivante :

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Pour une clé privée chiffrée, utilisez la structure suivante :

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

L'exemple suivant affiche le contenu d'un secret pour l'authentification mTLS à l'aide d'une clé privée chiffrée. Pour une clé privée chiffrée, vous devez inclure le mot de passe de clé privée dans le secret.

```
{
 "privateKeyPassword": "testpassword",
 "certificate": "-----BEGIN CERTIFICATE-----
MIIE5DCCAsygAwIBAgIRAPJdwaFaNRrytHBto0j5BA0wDQYJKoZIhvcNAQELBQAw
...
j0Lh4/+1HfgyE2KlmII36dg4IMzNjAFEBZiCRoPimO40s1cRqtFHXoal0QQbIlxk
cmUuiAii9R0=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFgjCCA2qgAwIBAgIQdjNZd6uFf9hbNC5RdfmHrzANBgkqhkiG9w0BAQsFADBb
...
rQoiowbbk5wXCheYSANQIfTZ6weQTgiCHCCbuuMKNVS95FkXm0vqVD/YpXKwA/no
c8PH3PSoAaRwMMgOSA2ALJvbRz8mpg==
-----END CERTIFICATE-----",
 "privateKey": "-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUiAFcK5hT/X7Kjmgp
...
QrSekqF+kWzmB6nAfSzgO9IaoAaytLvNgGTckWeUkWn/V0Ck+LdGUXzAC4RxZnoQ
zp2mwJn2NYB7AZ7+imp0azDZb+8YG2aUCiyqb6PnnA==
-----END ENCRYPTED PRIVATE KEY-----"
}
```

Comment EventBridge choisir un courtier Bootstrap

EventBridge choisit un courtier bootstrap en fonction des méthodes d'authentification disponibles sur votre cluster et du fait que vous fournissez un secret pour l'authentification. Si vous fournissez un secret pour MTL ou SASL/SCRAM, choisissez EventBridge automatiquement cette méthode d'authentification. Si vous ne fournissez pas de secret, EventBridge choisissez la méthode d'authentification la plus puissante active sur votre cluster. Voici l'ordre de priorité dans lequel un courtier est EventBridge sélectionné, de l'authentification la plus forte à la plus faible :

- MTLs (secret fourni pour les mTLS)
- SASL/SCRAM (secret fourni pour SASL/SCRAM)
- SASL IAM (aucun secret fourni et authentification IAM active)
- Protocole TLS non authentifié (aucun secret fourni et authentification IAM non active)
- Texte brut (aucun secret fourni, l'authentification IAM et le protocole TLS non authentifié ne sont pas actifs)

> **Note**
>
> S'il ne EventBridge parvient pas à se connecter au type de courtier le plus sûr, il ne tente pas de se connecter à un autre type de courtier (plus faible). Si vous souhaitez EventBridge choisir un type de courtier plus faible, désactivez toutes les méthodes d'authentification renforcées sur votre cluster.

Configuration réseau

EventBridge doit avoir accès aux ressources Amazon Virtual Private Cloud (Amazon VPC) associées à votre cluster Amazon MSK.

- Pour accéder au VPC de votre cluster Amazon MSK, EventBridge vous pouvez utiliser un accès Internet sortant pour les sous-réseaux de votre source. Pour les sous-réseaux publics, il doit s'agir d'une passerelle NAT gérée. Pour les sous-réseaux privés, il peut s'agir d'une passerelle NAT ou de votre propre NAT. Assurez-vous que le NAT possède une adresse IP publique et peut se connecter à Internet.
- EventBridge Pipes prend également en charge la diffusion d'événements AWS PrivateLink, ce qui vous permet d'envoyer des événements depuis une source d'événements située dans un

Amazon Virtual Private Cloud (Amazon VPC) vers une cible Pipes sans passer par l'Internet public. Vous pouvez utiliser Pipes pour interroger depuis Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogéré et des Amazon MQ sources résidant dans un sous-réseau privé sans avoir à déployer une passerelle Internet, à configurer des règles de pare-feu ou à configurer des serveurs proxy.

Pour configurer un point de terminaison VPC, consultez la section Créer un point de terminaison VPC dans le guide de l'utilisateur.AWS PrivateLink Pour le nom du service, sélectionnez`com.amazonaws.region.pipes-data`.

Vous devez configurer vos groupes de sécurité Amazon VPC avec les règles suivantes (au minimum) :

- Règles de trafic entrant : autorisez tout le trafic sur le port du courtier Amazon MSK pour les groupes de sécurité spécifiés pour votre source.
- Règles sortantes : autorisent tout le trafic sur le port 443 pour toutes les destinations. Autorisez tout le trafic sur le port du courtier Amazon MSK pour les groupes de sécurité spécifiés pour votre source.

  Les ports Broker incluent :

  - 9092 pour le texte en clair
  - 9094 pour TLS
  - 9096 pour SASL
  - 9098 pour IAM

> **Note**
>
> Votre configuration Amazon VPC est détectable via l'API Amazon MSK. Vous n'avez pas besoin de la définir pendant la configuration.

Identifiant de groupe de consommateurs personnalisable

Lorsque vous configurez Apache Kafka en tant que source, vous pouvez spécifier un identifiant de groupe de consommateurs. Cet identifiant de groupe de consommateurs est un identifiant existant pour le groupe de consommateurs Apache Kafka auquel vous souhaitez rattacher votre canal.

Vous pouvez utiliser cette fonctionnalité pour migrer toutes les configurations de traitement des enregistrements Apache Kafka en cours depuis d'autres utilisateurs vers. EventBridge

Si vous spécifiez un identifiant de groupe de consommateurs et qu'il existe d'autres sondeurs actifs au sein de ce groupe de consommateurs, Apache Kafka distribue des messages à tous les consommateurs. En d'autres termes, EventBridge ne reçoit pas tous les messages relatifs au sujet Apache Kafka. Si vous EventBridge souhaitez gérer tous les messages du sujet, désactivez tous les autres sondeurs de ce groupe de consommateurs.

En outre, si vous spécifiez un identifiant de groupe de consommateurs et qu'Apache Kafka trouve un groupe de consommateurs valide portant le même identifiant, il EventBridge ignore le `StartingPosition` paramètre de votre canal. EventBridge Commence plutôt à traiter les enregistrements en fonction de la compensation engagée par le groupe de consommateurs. Si vous spécifiez un identifiant de groupe de consommateurs et qu'Apache Kafka ne trouve aucun groupe de consommateurs existant, EventBridge configure votre source avec l'identifiant spécifié. `StartingPosition`

L'identifiant du groupe de consommateurs que vous spécifiez doit être unique parmi toutes vos sources d'événements Apache Kafka. Après avoir créé un canal avec l'identifiant de groupe de consommateurs spécifié, vous ne pouvez plus mettre à jour cette valeur.

Autoscaling de la source Amazon MSK

Lorsque vous créez initialement une source Amazon MSK, EventBridge alloue un consommateur pour traiter toutes les partitions de la rubrique Apache Kafka. Chaque consommateur dispose de plusieurs processeurs exécutés en parallèle pour gérer des charges de travail accrues. En outre, EventBridge augmente ou diminue automatiquement le nombre de consommateurs en fonction de la charge de travail. Pour préserver l'ordre des messages dans chaque partition, le nombre maximum de consommateurs est de un par partition dans la rubrique.

EventBridge Évalue, à intervalles d'une minute, le décalage entre les utilisateurs et toutes les partitions du sujet. Si le décalage est trop élevé, la partition reçoit les messages plus rapidement qu'elle ne EventBridge peut les traiter. Si nécessaire, EventBridge ajoute ou supprime des consommateurs du sujet. Le processus de mise à l'échelle consistant à ajouter ou à supprimer des consommateurs a lieu dans les trois minutes suivant l'évaluation.

Si votre cible est surchargée, vous EventBridge réduisez le nombre de consommateurs. Cette action réduit la charge de travail du canal en diminuant le nombre de messages que les consommateurs peuvent échanger avec le canal.

Apache Kafka diffuse en tant que source

Apache Kafka est une plateforme open source de streaming d'événements qui prend en charge des charges de travail telles que les canaux de données et l'analytique de streaming. Vous pouvez utiliser Amazon Managed Streaming for Apache Kafka (Amazon MSK) ou un cluster Apache Kafka autogéré. En AWS termes terminologiques, un cluster autogéré fait référence à tout cluster Apache Kafka non hébergé par AWS. Cela inclut à la fois les clusters que vous gérez vous-même, ainsi que ceux hébergés par un fournisseur tiers Confluent Cloud, tel que CloudKarafka, ou Redpanda.

Pour plus d'informations sur les autres options d' AWS hébergement pour votre cluster, consultez la section Meilleures pratiques pour exécuter Apache Kafka AWS sur le blog AWS Big Data.

Apache Kafka en tant que source fonctionne de la même manière qu'Amazon Simple Queue Service (Amazon SQS) ou Amazon Kinesis. EventBridgeinterroge en interne les nouveaux messages provenant de la source, puis invoque la cible de manière synchrone. EventBridge lit les messages par lots et les fournit à votre fonction sous forme de charge utile d'événements. La taille de lot maximale est configurable. (par défaut, 100 messages).

Pour les sources basées sur Apache Kafka, EventBridge prend en charge les paramètres de contrôle du traitement, tels que les fenêtres de traitement par lots et la taille des lots.

EventBridge envoie le lot de messages dans le paramètre d'événement lorsqu'il invoque votre canal. La charge utile d'un événement contient un tableau de messages. Chaque élément de tableau contient les détails de la rubrique Apache Kafka et l'identifiant de partition Apache Kafka, ainsi qu'un horodatage et un message codé en base64.

Exemples d'événements

L'exemple d'événement suivant montre les informations reçues par le canal. Vous pouvez utiliser cet événement pour créer et filtrer vos modèles d'événements, ou pour définir la transformation d'entrée. Tous les champs ne peuvent pas être filtrés. Pour plus d'informations sur les champs que vous pouvez filtrer, consultez ???.

```
[
  {
    "eventSource": "SelfManagedKafka",
    "bootstrapServers": "b-2.demo-cluster-1.a1bcde.c1.kafka.us-
east-1.amazonaws.com:9092,b-1.demo-cluster-1.a1bcde.c1.kafka.us-
east-1.amazonaws.com:9092",
    "eventSourceKey": "mytopic-0",
    "topic": "mytopic",
```

```
    "partition": 0,
    "offset": 15,
    "timestamp": 1545084650987,
    "timestampType": "CREATE_TIME",
    "key":"abcDEFghiJKLmnoPQRstuVWXyz1234==",
    "value":"SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
    "headers": [
      {
        "headerKey": [
          104,
          101,
          97,
          100,
          101,
          114,
          86,
          97,
          108,
          117,
          101
        ]
      }
    ]
  }
]
```

Authentification de cluster Apache Kafka

EventBridge Pipes prend en charge plusieurs méthodes pour s'authentifier auprès de votre cluster Apache Kafka autogéré. Veillez à configurer le cluster Apache Kafka de sorte à utiliser l'une des méthodes d'authentification prises en charge suivantes : Pour plus d'informations sur la sécurité Apache Kafka, consultez la section Sécurité de la documentation d'Apache Kafka.

Accès VPC

Si vous utilisez un environnement Apache Kafka autogéré dans lequel seuls les utilisateurs d'Apache Kafka au sein de votre VPC ont accès à vos courtiers Apache Kafka, vous devez configurer Amazon Virtual Private Cloud (Amazon VPC) dans la source Apache Kafka.

Authentification SASL/SCRAM

EventBridge Pipes prend en charge l'authentification simple et l'authentification SASL/SCRAM (Security Layer/Salted Challenge Response Authentication Mechanism) avec le cryptage TLS

(Transport Layer Security). EventBridge Pipes envoie les informations d'identification cryptées pour s'authentifier auprès du cluster. Pour plus d'informations sur l'authentification SASL/SCRAM, consultez RFC 5802.

EventBridge Pipes prend en charge l'authentification SASL/PLAIN avec le cryptage TLS. Avec l'authentification SASL/PLAIN, EventBridge Pipes envoie les informations d'identification sous forme de texte clair (non crypté) au serveur.

Pour l'authentification SASL, vous devez stocker les informations d'identification en tant que secret dans AWS Secrets Manager.

Authentification TLS mutuelle

Mutual TLS (mTLS) fournit une authentification bidirectionnelle entre le client et le serveur. Le client envoie un certificat au serveur pour que le serveur vérifie le client, et le serveur envoie un certificat au client pour que le client vérifie le serveur.

Dans Apache Kafka autogéré, EventBridge Pipes agit en tant que client. Vous configurez un certificat client (en tant que secret dans Secrets Manager) pour authentifier EventBridge Pipes auprès de vos courtiers Apache Kafka. Le certificat client doit être signé par une autorité de certification (CA) située dans le magasin d'approbations du serveur.

Le cluster Apache Kafka envoie un certificat de serveur à EventBridge Pipes pour authentifier les courtiers Apache Kafka auprès de Pipes. EventBridge Le certificat de serveur peut être un certificat d'autorité de certification public ou un certificat CA/auto-signé privé. Le certificat d'autorité de certification public doit être signé par une autorité de certification qui se trouve dans le magasin de confiance de EventBridge Pipes. Pour un certificat privé ou auto-signé, vous configurez le certificat CA racine du serveur (en tant que secret dans Secrets Manager). EventBridge Pipes utilise le certificat racine pour vérifier les courtiers Apache Kafka.

Pour plus d'informations sur mTLS, consultez Présentation de l'authentification TLS mutuelle pour Amazon MSK en tant que source (langue française non garantie).

Configuration du secret du certificat client

Le secret CLIENT_CERTIFICATE_TLS_AUTH nécessite un champ de certificat et un champ de clé privée. Pour une clé privée chiffrée, le secret nécessite un mot de passe de clé privée. Le certificat et la clé privée doivent être au format PEM.

> **Note**
>
> EventBridge Pipes prend en charge les algorithmes de chiffrement à clé privée PBES1 (mais pas PBES2).

Le champ de certificat doit contenir une liste de certificats, commençant par le certificat client, suivi de tous les certificats intermédiaires et se terminant par le certificat racine. Chaque certificat doit commencer sur une nouvelle ligne avec la structure suivante :

```
-----BEGIN CERTIFICATE-----
        <certificate contents>
-----END CERTIFICATE-----
```

Secrets Manager prend en charge les secrets jusqu'à 65 536 octets, ce qui offre suffisamment d'espace pour de longues chaînes de certificats.

La clé privée doit être au format PKCS #8, avec la structure suivante :

```
-----BEGIN PRIVATE KEY-----
         <private key contents>
-----END PRIVATE KEY-----
```

Pour une clé privée chiffrée, utilisez la structure suivante :

```
-----BEGIN ENCRYPTED PRIVATE KEY-----
          <private key contents>
-----END ENCRYPTED PRIVATE KEY-----
```

L'exemple suivant affiche le contenu d'un secret pour l'authentification mTLS à l'aide d'une clé privée chiffrée. Pour une clé privée chiffrée, incluez le mot de passe de clé privée dans le secret.

```
{
 "privateKeyPassword": "testpassword",
 "certificate": "-----BEGIN CERTIFICATE-----
MIIE5DCCAsygAwIBAgIRAPJdwaFaNRrytHBto0j5BA0wDQYJKoZIhvcNAQELBQAw
...
j0Lh4/+1HfgyE2KlmII36dg4IMzNjAFEBZiCRoPimO40s1cRqtFHXoal0QQbIlxk
cmUuiAii9R0=
```

```
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFgjCCA2qgAwIBAgIQdjNZd6uFf9hbNC5RdfmHrzANBgkqhkiG9w0BAQsFADBb
...
rQoiowbbk5wXCheYSANQIfTZ6weQTgiCHCCbuuMKNVS95FkXm0vqVD/YpXKwA/no
c8PH3PSoAaRwMMgOSA2ALJvbRz8mpg==
-----END CERTIFICATE-----",
 "privateKey": "-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFKzBVBgkqhkiG9w0BBQ0wSDAnBgkqhkiG9w0BBQwwGgQUiAFcK5hT/X7Kjmgp
...
QrSekqF+kWzmB6nAfSzgO9IaoAaytLvNgGTckWeUkWn/V0Ck+LdGUXzAC4RxZnoQ
zp2mwJn2NYB7AZ7+imp0azDZb+8YG2aUCiyqb6PnnA==
-----END ENCRYPTED PRIVATE KEY-----"
}
```

Configuration du secret du certificat d'autorité de certification racine du serveur

Vous créez ce secret si vos agents Apache Kafka utilisent le chiffrement TLS avec des certificats signés par une CA privée. Vous pouvez utiliser le chiffrement TLS pour l'authentification VPC, SASL/SCRAM, SASL/PLAIN ou mTLS.

Le secret du certificat de CA racine du serveur requiert un champ contenant le certificat de CA racine de l'agent Apache Kafka au format PEM. La structure du secret est présentée dans l'exemple suivant.

```
{
     "certificate": "-----BEGIN CERTIFICATE-----
  MIID7zCCAtegAwIBAgIBADANBgkqhkiG9w0BAQsFADCBmDELMAkGA1UEBhMCVVMx
  EDAOBgNVBAgTB0FyaXpvbmExEzARBgNVBAcTClNjb3R0c2RhbGUxJTAjBgNVBAoT
  HFN0YXJmaWVsZCBUZWNobm9sb2dpZXMsIEluYy4xOzA5BgNVBAMTMlN0YXJmaWVs
  ZCBTZXJ2aWNlcyBSb290IENlcnRpZmljYXRlIEF1dG...
  -----END CERTIFICATE-----"
```

Configuration réseau

Si vous utilisez un environnement Apache Kafka autogéré qui utilise une connectivité VPC privée EventBridge , vous devez avoir accès aux ressources Amazon Virtual Private Cloud (Amazon VPC) associées à vos courtiers Apache Kafka.

- Pour accéder au VPC de votre cluster Apache Kafka, EventBridge vous pouvez utiliser un accès Internet sortant pour les sous-réseaux de votre source. Pour les sous-réseaux publics, il doit s'agir d'une passerelle NAT gérée. Pour les sous-réseaux privés, il peut s'agir d'une passerelle NAT

ou de votre propre NAT. Assurez-vous que le NAT possède une adresse IP publique et peut se connecter à Internet.

- EventBridge Pipes prend également en charge la diffusion d'événements AWS PrivateLink, ce qui vous permet d'envoyer des événements depuis une source d'événements située dans un Amazon Virtual Private Cloud (Amazon VPC) vers une cible Pipes sans passer par l'Internet public. Vous pouvez utiliser Pipes pour interroger depuis Amazon Managed Streaming for Apache Kafka (Amazon MSK), Apache Kafka autogéré et des Amazon MQ sources résidant dans un sous-réseau privé sans avoir à déployer une passerelle Internet, à configurer des règles de pare-feu ou à configurer des serveurs proxy.

  Pour configurer un point de terminaison VPC, consultez la section Créer un point de terminaison VPC dans le guide de l'utilisateur.AWS PrivateLink Pour le nom du service, sélectionnez`com.amazonaws.`*`region`*`.pipes-data`.

Vous devez configurer vos groupes de sécurité Amazon VPC avec les règles suivantes (au minimum) :

- Règles de trafic entrant : autorisez tout le trafic sur le port du broker Apache Kafka pour les groupes de sécurité spécifiés pour votre source.
- Règles sortantes : autorisent tout le trafic sur le port 443 pour toutes les destinations. Autorisez tout le trafic sur le port du broker Apache Kafka pour les groupes de sécurité spécifiés pour votre source.

  Les ports Broker incluent :

  - 9092 pour le texte en clair
  - 9094 pour TLS
  - 9096 pour SASL
  - 9098 pour IAM

Mise à l'échelle automatique des consommateurs avec les sources Apache Kafka

Lorsque vous créez une source Apache Kafka pour la première fois, EventBridge elle alloue un consommateur pour traiter toutes les partitions du sujet Kafka. Chaque consommateur dispose de plusieurs processeurs exécutés en parallèle pour gérer des charges de travail accrues. En outre, EventBridge augmente ou diminue automatiquement le nombre de consommateurs en fonction de la

charge de travail. Pour préserver l’ordre des messages dans chaque partition, le nombre maximum de consommateurs est de un par partition dans la rubrique.

EventBridge Évalue, à intervalles d'une minute, le décalage entre les utilisateurs et toutes les partitions du sujet. Si le décalage est trop élevé, la partition reçoit les messages plus rapidement qu'elle ne EventBridge peut les traiter. Si nécessaire, EventBridge ajoute ou supprime des consommateurs du sujet. Le processus de mise à l’échelle consistant à ajouter ou à supprimer des consommateurs a lieu dans les trois minutes suivant l’évaluation.

Si votre cible est surchargée, vous EventBridge réduisez le nombre de consommateurs. Cette action réduit la charge de travail de la fonction en diminuant le nombre de messages que les consommateurs peuvent échanger avec la fonction.

Amazon Simple Queue Service en tant que source

Vous pouvez utiliser EventBridge Pipes pour recevoir des enregistrements d'une file d'attente Amazon SQS. Ensuite, vous pouvez éventuellement filtrer ou améliorer ces enregistrements avant de les envoyer à une destination disponible pour être traités.

Vous pouvez utiliser un canal pour traiter les messages d'une file d'attente Amazon Simple Queue Service (Amazon SQS). EventBridge Les tuyaux supportent les files d'attente standard et les files d'attente du premier entré, premier sorti (FIFO). Avec Amazon SQS, vous pouvez décharger des tâches d’un composant de votre application en les envoyant à une file d’attente, puis en les traitant de manière asynchrone.

EventBridge interroge la file d'attente et invoque votre canal de manière synchrone avec un événement contenant des messages de file d'attente. EventBridge lit les messages par lots et invoque votre canal une fois pour chaque lot. Lorsque votre canal traite avec succès un lot, il EventBridge supprime ses messages de la file d'attente.

Par défaut, EventBridge interroge simultanément jusqu'à 10 messages dans votre file d'attente et envoie ce lot à votre canal. Pour éviter d’invoquer le canal avec un petit nombre d’enregistrements, vous pouvez indiquer à la source d’événement de les mettre en mémoire tampon pendant cinq minutes maximum en configurant une fenêtre de traitement par lots. Avant d'appeler le canal, EventBridge continue à interroger les messages de la file d'attente standard Amazon SQS jusqu'à ce que l'un des événements suivants se produise :

- La fenêtre de traitement par lots expire.
- Le quota de taille de la charge utile d’invocation est atteint.

- La taille de lot maximale configurée est atteinte.

> **Note**
>
> Si vous utilisez une fenêtre de traitement par lots et que votre file d'attente Amazon SQS contient peu de trafic, vous EventBridge pouvez attendre jusqu'à 20 secondes avant d'appeler votre canal. C'est le cas même si vous définissez une fenêtre de traitement par lots inférieure à 20 secondes. Pour les files d'attente FIFO, les enregistrements contiennent des attributs supplémentaires liés à la déduplication et au séquençage.

Lors de la EventBridge lecture d'un lot, les messages restent dans la file d'attente mais sont masqués pendant la durée du délai de visibilité de la file d'attente. Si votre canal traite le lot avec succès, EventBridge supprime les messages de la file d'attente. Par défaut, si votre canal rencontre une erreur lors du traitement d'un lot, tous les messages de ce lot redeviennent visibles dans la file d'attente. Pour cette raison, le code de votre canal doit pouvoir traiter le même message plusieurs fois sans effets secondaires involontaires. Vous pouvez modifier ce comportement de retraitement en incluant les défaillances d'éléments de lot dans la réponse de votre canal. L'exemple suivant présente un événement pour un lot de deux messages.

Exemples d'événements

L'exemple d'événement suivant montre les informations reçues par le canal. Vous pouvez utiliser cet événement pour créer et filtrer vos modèles d'événements, ou pour définir la transformation d'entrée. Tous les champs ne peuvent pas être filtrés. Pour plus d'informations sur les champs que vous pouvez filtrer, consultez ???.

File d'attente standard

```
[
  {
    "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
    "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1545082649183",
      "SenderId": "AIDAIENQZJOLO23YVJ4VO",
      "ApproximateFirstReceiveTimestamp": "1545082649185"
```

```
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:my-queue",
    "awsRegion": "us-east-2"
  },
  {
    "messageId": "2e1424d4-f796-459a-8184-9c92662be6da",
    "receiptHandle": "AQEBzWwaftRI0KuVm4tP+/7q1rGgNqicHq...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1545082650636",
      "SenderId": "AIDAIENQZJOLO23YVJ4VO",
      "ApproximateFirstReceiveTimestamp": "1545082650649"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:my-queue",
    "awsRegion": "us-east-2"
  }
]
```

File d’attente FIFO

```
[
  {
    "messageId": "11d6ee51-4cc7-4302-9e22-7cd8afdaadf5",
    "receiptHandle": "AQEBBX8nesZEXmkhsmZeyIE8iQAMig7qw...",
    "body": "Test message.",
    "attributes": {
      "ApproximateReceiveCount": "1",
      "SentTimestamp": "1573251510774",
      "SequenceNumber": "18849496460467696128",
      "MessageGroupId": "1",
      "SenderId": "AIDAIO23YVJENQZJOL4VO",
      "MessageDeduplicationId": "1",
      "ApproximateFirstReceiveTimestamp": "1573251510774"
    },
    "messageAttributes": {},
    "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
```

```
    "eventSource": "aws:sqs",
    "eventSourceARN": "arn:aws:sqs:us-east-2:123456789012:fifo.fifo",
    "awsRegion": "us-east-2"
  }
]
```

Dimensionnement et traitement

Pour les files d'attente standard, EventBridge utilise un long sondage pour interroger une file d'attente jusqu'à ce qu'elle devienne active. Lorsque des messages sont disponibles, EventBridge lit jusqu'à cinq lots et les envoie à votre canal. Si les messages sont toujours disponibles, EventBridge augmente le nombre de processus lisant des lots de 300 instances supplémentaires par minute. Le nombre maximal de lots qui peuvent être traités simultanément par un canal est de 1 000.

Pour les files d'attente FIFO, EventBridge envoie des messages à votre canal dans l'ordre dans lequel il les reçoit. Lorsque vous envoyez un message à une file d'attente FIFO, vous spécifiez un ID de groupe de messages. Amazon SQS facilite l'envoi de messages d'un même groupe à EventBridge, dans l'ordre. EventBridge trie les messages reçus en groupes et n'envoie qu'un seul lot à la fois pour un groupe. Si votre canal renvoie une erreur, le canal tente toutes les tentatives sur les messages concernés avant de EventBridge recevoir des messages supplémentaires du même groupe.

Configuration d'une file d'attente à utiliser avec EventBridge Pipes

Créez une file d'attente Amazon SQS à utiliser en tant que source pour votre canal. Configurez ensuite la file d'attente pour laisser le temps à votre canal de traiter chaque lot d'événements et de réessayer en réponse EventBridge à des erreurs de régulation à mesure qu'il prend de l'ampleur.

Pour laisser à votre canal le temps de traiter chaque lot d'enregistrements, définissez le délai de visibilité de la file d'attente source sur au moins six fois le temps d'exécution combiné de l'enrichissement du canal et des composants cibles. Le temps supplémentaire permet de EventBridge réessayer si votre tuyau est étranglé lors du traitement d'un lot précédent.

Si votre canal ne parvient pas à traiter un message plusieurs fois de suite, Amazon SQS peut envoyer celui-ci à une file d'attente de lettres mortes. Lorsque votre canal renvoie une erreur, il le EventBridge conserve dans la file d'attente. Une fois le délai de visibilité expiré, EventBridge reçoit à nouveau le message. Pour envoyer des messages à une deuxième file d'attente après plusieurs réceptions, configurez une file d'attente de lettres mortes sur votre file d'attente source.

> **Note**
>
> Assurez-vous que vous configurez la file d'attente de lettres mortes dans la file d'attente source, mais pas sur le canal. La file d'attente de lettres mortes que vous configurez sur un canal est utilisée pour la file d'attente d'invocation asynchrone du canal, mais pas pour les files d'attente sources.

Si votre canal renvoie une erreur ou ne peut pas être invoqué, car il a atteint le niveau de simultanéité maximal, le traitement peut aboutir avec des tentatives supplémentaires. Pour donner plus de chances aux messages d'être traités avant de les envoyer dans la file d'attente de lettres mortes, définissez `maxReceiveCount` sur 5 au minimum dans la stratégie de réacheminement de la file d'attente source.

Signalement des échecs d'articles par lots

Lorsqu'il EventBridge consomme et traite des données en streaming à partir d'une source, il vérifie par défaut le numéro de séquence le plus élevé d'un lot, mais uniquement lorsque le lot est totalement réussi. Pour éviter de retraiter les messages dont le traitement a réussi dans un lot ayant échoué, vous pouvez configurer votre enrichissement ou votre cible de sorte à renvoyer un objet en indiquant les messages qui ont réussi et ceux qui ont échoué. C'est ce que l'on appelle une réponse partielle de lot.

Pour de plus amples informations, veuillez consulter ???.

Conditions de réussite et d'échec

Si vous renvoyez l'un des éléments suivants, EventBridge considère un lot comme une réussite totale :

- Une liste `batchItemFailure` vide
- Une liste `batchItemFailure` nulle
- Une `EventResponse` vide
- Une `EventResponse` nulle

Si vous renvoyez l'un des éléments suivants, EventBridge considère un lot comme un échec total :

- Une chaîne `itemIdentifier` vide

- Un `itemIdentifier` nul
- Un `itemIdentifier` avec un nom de clé incorrect

EventBridge les nouvelles tentatives échouent en fonction de votre stratégie de réessai.

Filtrage Amazon EventBridge Pipes

Avec EventBridge Pipes, vous pouvez filtrer les événements d'une source donnée et n'en traiter qu'un sous-ensemble. Ce filtrage fonctionne de la même manière que le filtrage sur un bus d' EventBridge événements ou le mappage d'une source d'événements Lambda, en utilisant des modèles d'événements. Pour plus d'informations sur les modèles d'événements, consultez ???.

Un objet `FilterCriteria` de critères de filtre est une structure composée d'une liste de filtres (`Filters`). Chaque filtre est une structure qui définit un modèle de filtrage (`Pattern`). Un `Pattern` est une représentation sous forme de chaîne d'une règle de filtre JSON. Un objet `FilterCriteria` ressemble à l'exemple suivant :

```
{
  "Filters": [
    {"Pattern": "{ \"Metadata1\": [ rule1 ], \"data\": { \"Data1\": [ rule2 ] }}"
    }
  ]
}
```

Pour plus de clarté, voici la valeur du `Pattern` de filtre étendu en JSON simple :

```
{
  "Metadata1": [ pattern1 ],
  "data": {"Data1": [ pattern2 ]}
}
```

Les parties principales d'un objet `FilterCriteria` sont les propriétés de métadonnées et les propriétés de données.

- Les Propriétés des métadonnées sont les champs de l'objet événement. Dans l'exemple, `FilterCriteria.Metadata1` fait référence à une propriété de métadonnées.
- Les Propriétés de données sont les champs du corps de l'événement. Dans l'exemple, `FilterCriteria.Data1` fait référence à une propriété de données.

Supposons que votre flux Kinesis contienne l'événement suivant :

```
{
  "kinesisSchemaVersion": "1.0",
  "partitionKey": "1",
  "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
  "data": {"City": "Seattle",
    "State": "WA",
    "Temperature": "46",
    "Month": "December"
  },
  "approximateArrivalTimestamp": 1545084650.987
}
```

Lorsque l'événement passe par votre canal, il ressemble à ce qui suit avec le champ `data` codé en base64 :

```
{
  "kinesisSchemaVersion": "1.0",
  "partitionKey": "1",
  "sequenceNumber": "49590338271490256608559692538361571095921575989136588898",
  "data": "SGVsbG8sIHRoaXMgaXMgYSB0ZXN0Lg==",
  "approximateArrivalTimestamp": 1545084650.987,
  "eventSource": "aws:kinesis",
  "eventVersion": "1.0",
  "eventID":
 "shardId-000000000006:49590338271490256608559692538361571095921575989136588898",
  "eventName": "aws:kinesis:record",
  "invokeIdentityArn": "arn:aws:iam::123456789012:role/lambda-role",
  "awsRegion": "us-east-2",
  "eventSourceARN": "arn:aws:kinesis:us-east-2:123456789012:stream/lambda-stream"
}
```

Les propriétés de métadonnées de l'événement Kinesis correspondent à n'importe quel champ externe à l'objet `data`, tel que `partitionKey` ou `sequenceNumber`.

Les propriétés de données de l'événement Kinesis correspondent aux champs internes à l'objet `data`, tels que `City` ou `Temperature`.

Lorsque vous utilisez un filtre pour mettre en correspondance cet événement, vous pouvez utiliser des filtres sur les champs décodés. Par exemple, pour filtrer sur `partitionKey` et `City`, vous devez utiliser le filtre suivant :

```
{
  "partitionKey": [
    "1"
  ],
  "data": {
    "City": [
      "Seattle"
    ]
  }
}
```

Lorsque vous créez des filtres d'événements, EventBridge Pipes peut accéder au contenu des événements. Ce contenu est soit mis en échappement dans le code JSON, comme le champ body Amazon SQS, soit codé en base64, comme le champ `data` Kinesis. Si vos données sont au format JSON valide, vos modèles d'entrée ou vos chemins JSON pour les paramètres cibles peuvent référencer directement le contenu. Par exemple, si une source d'événement Kinesis est au format JSON valide, vous pouvez référencer une variable à l'aide de <$.data.someKey>.

Lorsque vous créez des modèles d'événements, vous pouvez filtrer en fonction des champs envoyés par l'API source, mais pas des champs ajoutés par l'opération d'interrogation. Les champs suivants ne peuvent pas être utilisés dans les modèles d'événements :

- `awsRegion`
- `eventSource`
- `eventSourceARN`
- `eventVersion`
- `eventID`
- `eventName`
- `invokeIdentityArn`
- `eventSourceKey`

Champs de message et de données

Chaque source EventBridge Pipe contient un champ qui contient le message ou les données de base. Nous les appelons champs de message ou champs de données. Ces champs sont spéciaux, car ils peuvent être mise en échappement dans le code JSON ou codés en base64, mais lorsqu'ils sont au format JSON valide, ils peuvent être filtrés avec des modèles JSON comme si le corps

n’était pas mis en échappement. Le contenu de ces champs peut également être utilisé de façon transparente dans les transformateurs d’entrée.

Filtrage correct des messages Amazon SQS

Si un message Amazon SQS ne répond pas à vos critères de filtrage, il est EventBridge automatiquement supprimé de la file d'attente. Vous n’avez pas besoin de supprimer manuellement ces messages dans Amazon SQS.

Pour Amazon SQS, le message `body` peut être n'importe quelle chaîne. Toutefois, cela peut être problématique si votre `FilterCriteria` s'attend à ce que `body` se présente dans un format JSON valide. Le scénario inverse est également vrai. Si le message entrant `body` est au format JSON valide, mais que votre critère de filtre s’attend à ce que `body` soit une chaîne de texte brut, cela peut entraîner un comportement inattendu.

Pour éviter ce problème, assurez-vous que le format de `body` dans `FilterCriteria` correspond au format attendu de `body` dans les messages que vous recevez de votre file d’attente. Avant de filtrer vos messages, évalue EventBridge automatiquement le format du message entrant `body` et de votre modèle de filtre pour`body`. S'il y a une incompatibilité, EventBridge supprime le message. Le tableau suivant résume cette évaluation :

Format du **body** du message entrant	Format du **body** du modèle de filtre	Action obtenue.
Chaîne de texte brut	Chaîne de texte brut	EventBridge filtres en fonction de vos critères de filtrage.
Chaîne de texte brut	Pas de modèle de filtre pour les propriétés des données	EventBridge filtres (sur les autres propriétés de métadonnées uniquement) en fonction de vos critères de filtre.
Chaîne de texte brut	JSON valide	EventBridge supprime le message.
JSON valide	Chaîne de texte brut	EventBridge supprime le message.

Format du **body** du message entrant	Format du **body** du modèle de filtre	Action obtenue.
JSON valide	Pas de modèle de filtre pour les propriétés des données	EventBridge filtres (sur les autres propriétés de métadonnées uniquement) en fonction de vos critères de filtre.
JSON valide	JSON valide	EventBridge filtres en fonction de vos critères de filtrage.

Si vous ne l'incluez `body` pas dans votre`FilterCriteria`, EventBridge ignore cette vérification.

Filtrage correct des messages Kinesis et DynamoDB

Une fois que vos critères de filtre traitent un enregistrement Kinesis ou DynamoDB, l'itérateur des flux passe au-delà de cet enregistrement. Si l'enregistrement ne répond pas à vos critères de filtre, vous n'avez pas besoin de supprimer manuellement l'enregistrement de la source de votre événement. Après la période de conservation, Kinesis et DynamoDB suppriment automatiquement ces anciens enregistrements. Si vous souhaitez que les enregistrements soient supprimés plus tôt, consultez Modification de la période de conservation des données.

Pour filtrer correctement les événements provenant de sources d'événements de flux, le champ de données et vos critères de filtre pour le champ de données doivent être au format JSON valide. (Pour Kinesis, le champ de données est `data`. Pour DynamoDB, le champ de données est`dynamodb`.) Si l'un des champs n'est pas dans un format JSON valide, EventBridge supprime le message ou génère une exception. Le tableau suivant résume le comportement spécifique :

Format des données entrantes (**`data`** ou **`dynamodb`**)	Pas de modèle de filtre pour les propriétés des données	Action obtenue.
JSON valide	JSON valide	EventBridge filtres en fonction de vos critères de filtrage.
JSON valide	Pas de modèle de filtre pour les propriétés des données	EventBridge filtres (sur les autres propriétés de métadonnées uniquement)

Format des données entrantes (**`data`** ou **`dynamodb`**)	Pas de modèle de filtre pour les propriétés des données	Action obtenue.
		en fonction de vos critères de filtre.
JSON valide	Non JSON	EventBridge lance une exception au moment du canal ou de la mise à jour. Le modèle de filtre des propriétés de données doit être au format JSON valide.
Non JSON	JSON valide	EventBridge fait tomber le record.
Non JSON	Pas de modèle de filtre pour les propriétés des données	EventBridge filtres (sur les autres propriétés de métadonnées uniquement) en fonction de vos critères de filtre.
Non JSON	Non JSON	EventBridge lance une exception au moment de la création ou de la mise à jour du canal. Le modèle de filtre des propriétés de données doit être au format JSON valide.

Filtrage approprié des messages Amazon Managed Streaming for Apache Kafka, Apache Kafka autogéré et Amazon MQ

Pour les sources Amazon MQ, le champ de message est `data`. Pour les sources Apache Kafka (Amazon MSK et Apache Kafka autogéré), il existe deux champs de message : `key` et `value`.

EventBridge supprime les messages qui ne correspondent pas à tous les champs inclus dans le filtre. Pour Apache Kafka, EventBridge valide les décalages pour les messages correspondants et non

correspondants après avoir invoqué la fonction avec succès. Pour Amazon MQ, EventBridge accuse réception des messages correspondants après avoir correctement invoqué la fonction et confirme les messages non correspondants lors du filtrage de ceux-ci.

Les messages Apache Kafka et Amazon MQ doivent être des chaînes codées en UTF-8, soit des chaînes en texte brut, soit au format JSON. En effet, il EventBridge décode les tableaux d'octets d'Apache Kafka et d'Amazon MQ en UTF-8 avant d'appliquer les critères de filtrage. Si vos messages utilisent un autre encodage, tel que UTF-16 ou ASCII, ou si le format du message ne correspond pas au `FilterCriteria` format, EventBridge traite uniquement les filtres de métadonnées. Le tableau suivant résume le comportement spécifique :

Format du message entrant (**`data`** ou **`key`** et **`value`**)	Modèle de filtre de format pour les propriétés des messages	Action obtenue.
Chaîne de texte brut	Chaîne de texte brut	EventBridge filtres en fonction de vos critères de filtrage.
Chaîne de texte brut	Pas de modèle de filtre pour les propriétés des données	EventBridge filtres (sur les autres propriétés de métadonnées uniquement) en fonction de vos critères de filtre.
Chaîne de texte brut	JSON valide	EventBridge filtres (sur les autres propriétés de métadonnées uniquement) en fonction de vos critères de filtre.
JSON valide	Chaîne de texte brut	EventBridge filtres (sur les autres propriétés de métadonnées uniquement) en fonction de vos critères de filtre.
JSON valide	Pas de modèle de filtre pour les propriétés des données	EventBridge filtres (sur les autres propriétés de métadonnées uniquement)

Format du message entrant (**`data`** ou **`key`** et **`value`**)	Modèle de filtre de format pour les propriétés des messages	Action obtenue.
		en fonction de vos critères de filtre.
JSON valide	JSON valide	EventBridge filtres en fonction de vos critères de filtrage.
Chaîne non codée UTF-8	JSON, chaîne de texte brut ou aucun modèle	EventBridge filtres (sur les autres propriétés de métadonnées uniquement) en fonction de vos critères de filtre.

Différences entre Lambda ESM et Pipes EventBridge

Lors du filtrage des événements, Lambda ESM et EventBridge Pipes fonctionnent généralement de la même manière. La principale différence réside dans le fait que le champ `eventSourceKey` n'est pas présent dans les charges utiles ESM.

Enrichissement des événements Amazon EventBridge Pipes

Avec l'étape d'enrichissement d'EventBridge Pipes, vous pouvez améliorer les données issues de la source avant de les envoyer à la cible. Par exemple, vous pouvez recevoir des événements de type Ticket créé qui n'incluent pas l'ensemble des données de ticket. Grâce à l'enrichissement, vous pouvez demander à une fonction Lambda d'appeler l'API `get-ticket` pour obtenir les détails complets du ticket. Les canaux peuvent ensuite envoyer ces informations à une cible.

Vous pouvez configurer les enrichissements suivants lorsque vous configurez un canal dans EventBridge :

- Destination d'API
- Amazon API Gateway
- Fonction Lambda
- Machine d'état Step Functions

> **Note**
>
> EventBridge Pipes prend uniquement en charge les flux de travaux express en tant qu'enrichissements.

EventBridge invoque les enrichissements de manière synchrone, car il doit attendre une réponse de l'enrichissement avant d'invoquer la cible.

Les réponses de l'enrichissement sont limitées à une taille maximale de 6 Mo.

Vous pouvez également transformer les données que vous recevez de la source avant de les envoyer pour amélioration. Pour de plus amples informations, veuillez consulter ???.

Filtrage des événements à l'aide de l'enrichissement

EventBridge Pipes transmet les réponses de l'enrichissement directement à la cible configurée. Cela inclut les réponses de type tableau pour les cibles qui prennent en charge les lots. Pour plus d'informations sur le comportement d'un lot, consultez ???. Vous pouvez également utiliser votre enrichissement comme filtre et transmettre moins d'événements que ceux reçus de la source. Si vous ne souhaitez pas invoquer la cible, renvoyez une réponse vide, telle que `""`, `{}` ou `[]`.

> **Note**
>
> Si vous souhaitez invoquer la cible avec une charge utile vide, renvoyez un tableau avec du code JSON vide `[{}]`.

Invocation d'enrichissements

EventBridge invoque les enrichissements de manière synchrone (type d'invocation défini sur `REQUEST_RESPONSE`), car il doit attendre une réponse de l'enrichissement avant d'invoquer la cible.

> **Note**
>
> Pour les machines d'état Step Functions, EventBridge prend uniquement en charge les flux de travaux express en tant qu'enrichissements, car ils peuvent être invoqués de manière synchrone.

Objectifs d'Amazon EventBridge Pipes

Vous pouvez envoyer les données de votre canal à une cible spécifique. Vous pouvez configurer les cibles suivantes lors de la configuration d'un canal dans EventBridge :

- Destination d'API
- API Gateway
- File d’attente des tâches de traitement par lot
- CloudWatch groupe de journaux
- Tâche ECS
- Bus d'événements dans le même compte et la même région
- Flux de diffusion Firehose
- Modèle d'évaluation Inspector
- Flux Kinesis
- Fonction Lambda (SYNC ou ASYNC)
- Requêtes d’API relatives aux données du cluster Redshift
- SageMaker Pipeline
- Rubrique Amazon SNS (rubriques FIFO SNS non prises en charge)
- File d’attente Amazon SQS
- Machine d’état Step Functions
  - Flux de travaux express (SYNC ou ASYNC)
  - Flux de travaux standard (ASYNC)
- Timestream pour LiveAnalytics table

Paramètres de cible

Certains services cibles n'envoient pas la charge utile de l'événement à la cible, mais traitent l'événement comme un déclencheur pour appeler une API spécifique. EventBridge utilise le `PipeTargetParameters`pour spécifier quelles informations sont envoyées à cette API. Tel est le cas des éléments suivants :

- Destinations d’API (Les données envoyées à une destination d’API doivent correspondre à la structure de l’API. Vous devez utiliser l’objet `InputTemplate` pour vous assurer que les données

sont correctement structurées. Si vous souhaitez inclure la charge utile de l'événement d'origine, référencez-la dans `InputTemplate`.)

- API Gateway (Les données envoyées à API Gateway doivent correspondre à la structure de l'API. Vous devez utiliser l'objet `InputTemplate` pour vous assurer que les données sont correctement structurées. Si vous souhaitez inclure la charge utile de l'événement d'origine, référencez-la dans `InputTemplate`.)
- `PipeTargetRedshiftDataParameters` (clusters d'API de données Amazon Redshift)
- `PipeTargetSageMakerPipelineParameters`(Pipelines de création de modèles Amazon SageMaker Runtime)
- `PipeTargetBatchJobParameters` (AWS Batch)

> **Note**
>
> EventBridge ne prend pas en charge toutes les syntaxes JSON Path et ne l'évalue pas lors de l'exécution. La syntaxe prise en charge inclut :
>
> - notation par points (par exemple, `$.detail`)
> - tirets
> - traits de soulignement
> - caractères alphanumériques
> - index de tableau
> - caractères génériques (*)

Paramètres de chemin dynamiques

EventBridge Les paramètres cibles des tuyaux prennent en charge la syntaxe de chemin dynamique JSON facultative. Vous pouvez utiliser cette syntaxe pour spécifier des chemins JSON au lieu de valeurs statiques (par exemple, `$.detail.state`). La valeur entière doit être un chemin JSON, pas seulement une partie de celui-ci. Par exemple, `RedshiftParameters.Sql` peut avoir la valeur `$.detail.state`, mais pas la valeur `"SELECT * FROM $.detail.state"`. Ces chemins sont remplacés de manière dynamique lors de l'exécution par des données provenant de la charge utile de l'événement elle-même au niveau du chemin spécifié. Les paramètres de chemin dynamiques ne peuvent pas faire référence à des valeurs nouvelles ou transformées résultant d'une transformation

d’entrée. La syntaxe prise en charge pour les chemins JSON de paramètres dynamiques est la même que lors de la transformation d’une entrée. Pour plus d’informations, consultez ???.

La syntaxe dynamique peut être utilisée sur tous les champs de chaîne, autres que les champs enum, de tous les paramètres d'enrichissement et de cible de EventBridge Pipes, sauf :

- `PipeTargetCloudWatchLogsParameters.LogStreamName`
- `PipeTargetEventBridgeEventBusParameters.EndpointId`
- `PipeEnrichmentHttpParameters.HeaderParameters`
- `PipeTargetHttpParameters.HeaderParameters`

Par exemple, pour définir la cible Kinesis `PartitionKey` d'un canal sur une clé personnalisée provenant de votre événement source, définissez le. KinesisTargetParameter PartitionKeypour :

- `"$.data.`*`someKey`*`"` pour une source Kinesis
- `"$.body.`*`someKey`*`"` pour une source Amazon SQS

Ensuite, si la charge utile de l'événement est une chaîne JSON valide, par exemple`{"`*`someKey`*`":"`*`someValue`*`"}`, EventBridge extrait la valeur du chemin JSON et l'utilise comme paramètre cible. Dans cet exemple, EventBridge définirait le Kinesis sur « `PartitionKey` *`SomeValue`* ».

Autorisations

Pour effectuer des appels d'API sur les ressources que vous possédez, EventBridge Pipes a besoin des autorisations appropriées. EventBridge PiPES utilise le rôle IAM que vous spécifiez sur le canal pour l'enrichissement et cible les appels à l'aide du principal IAM. `pipes.amazonaws.com`

Invocation de cibles

EventBridge propose les méthodes suivantes pour invoquer une cible :

- Synchrone (type d'invocation défini sur`REQUEST_RESPONSE`) : EventBridge attend une réponse de la cible avant de continuer.
- De manière asynchrone (type d'invocation défini sur`FIRE_AND_FORGET`) : EventBridge n'attend pas de réponse avant de continuer.

Par défaut, pour les canaux dont les sources sont ordonnées, EventBridge invoque les cibles de manière synchrone car une réponse de la cible est nécessaire avant de passer à l'événement suivant.

Si une source ne fait pas respecter l'ordre, telle qu'une file d'attente Amazon SQS standard, elle EventBridge peut invoquer une cible prise en charge de manière synchrone ou asynchrone.

Avec les fonctions Lambda et les machines d'état Step Functions, vous pouvez configurer le type d'invocation.

> **Note**
>
> Pour les machines d'état Step Functions, les flux de travaux standard doivent être invoqués de manière asynchrone.

EventBridge Les tuyaux ciblent les spécificités

AWS Batch files d'attente pour les emplois

Tous les AWS Batch `submitJob` paramètres sont configurés explicitement avec`BatchParameters`, et comme tous les paramètres Pipe, ils peuvent être dynamiques à l'aide d'un chemin JSON vers la charge utile de votre événement entrant.

CloudWatch Groupe de journaux

Que vous utilisiez un transformateur d'entrée ou non, la charge utile de l'événement est utilisée comme message du journal. Vous pouvez définir `Timestamp` (ou le nom explicite `LogStreamName` de votre destination) via `CloudWatchLogsParameters` dans `PipeTarget`. Comme pour tous les paramètres de canal, ces paramètres peuvent être dynamiques en utilisant un chemin JSON pointant vers la charge utile de votre événement entrant.

Tâches Amazon ECS

Tous les paramètres `runTask` Amazon ECS sont configurés de manière explicite via `EcsParameters`. Comme pour tous les paramètres de canal, ces paramètres peuvent être dynamiques en utilisant un chemin JSON pointant vers la charge utile de votre événement entrant.

Fonctions Lambda et flux de travaux Step Functions

Lambda et Step Functions ne disposent pas d'une API par lots. Pour traiter des lots d'événements provenant d'une source de canal, le lot est converti en tableau JSON et transmis en tant qu'entrée à la cible Lambda ou Step Functions. Pour plus d'informations, consultez ???.

Timestream pour LiveAnalytics table

Les considérations à prendre en compte lors de la spécification d'une LiveAnalytics table Timestream for comme cible de canal incluent :

- Les flux Apache Kafka (y compris ceux provenant de Amazon MSK fournisseurs tiers) ne sont actuellement pas pris en charge en tant que source de canaux.
- Si vous avez indiqué un DynamoDB flux Kinesis ou comme source de canal, vous devez spécifier le nombre de tentatives de nouvelle tentative.

  Pour plus d'informations, voir ???.

Traitement par lots et simultanéité d'Amazon EventBridge Pipes

Comportement de traitement par lots

EventBridge Pipes prend en charge le traitement par lots depuis la source et vers les cibles qui le supportent. En outre, le traitement par lots jusqu'à l'enrichissement est pris en charge pour AWS Lambda et AWS Step Functions. Étant donné que différents services prennent en charge différents niveaux de traitement par lots, vous ne pouvez pas configurer un canal avec une taille de lot supérieure à celle prise en charge par la cible. Par exemple, les sources de flux Amazon Kinesis prennent en charge une taille de lot maximale de 10 000 enregistrements, mais Amazon Simple Queue Service prend en charge un maximum de 10 messages par lot en tant que cible. Par conséquent, un canal partant d'un flux Kinesis jusqu'à une file d'attente Amazon SQS peut avoir une taille de lot maximale configurée sur la source de 10.

Si vous configurez un canal avec un enrichissement ou une cible qui ne prend pas en charge le traitement par lots, vous ne pourrez pas activer le traitement par lots sur la source.

Lorsque le traitement par lots est activé sur la source, des tableaux d'enregistrements JSON sont transmis par le canal, puis mappés à l'API par lots d'un enrichissement ou d'une cible pris(e) en charge. Les transformateurs d'entrée sont appliqués séparément sur chaque enregistrement

JSON individuel du tableau, et non sur le tableau dans son ensemble. Pour obtenir des exemples de tableaux, consultez ??? et sélectionnez une source spécifique. Pipes utilisera l'API par lots pour l'enrichissement ou la cible pris(e) en charge, même si la taille du lot est égale à 1. Si l'enrichissement ou la cible ne possède pas d'API par lots, mais reçoit des charges utiles JSON complètes, telles que Lambda et Step Functions, l'intégralité du tableau JSON est envoyée en une seule demande. La demande sera envoyée sous forme de tableau JSON même si la taille du lot est de 1.

Si un canal est configuré pour le traitement par lots au niveau de la source, et que la cible prend en charge le traitement par lots, vous pouvez renvoyer un tableau d'éléments JSON depuis votre enrichissement. Ce tableau peut inclure un tableau plus court ou plus long que la source d'origine. Toutefois, si la taille du tableau est supérieure à la taille de lot prise en charge par la cible, le canal n'invoquera pas la cible.

Cibles pouvant être traitées par lots prises en charge

Cible	Taille maximale du lot
CloudWatch Journaux	10 000
EventBridge bus événementiel	10
Stream Firehose	500
Flux Kinesis	500
Fonction Lambda	définie par le client
Machine d'état Step Functions	définie par le client
Rubrique Amazon SNS	10
File d'attente Amazon SQS	10

Les enrichissements et cibles suivants reçoivent la charge utile complète d'événements par lots à traiter et sont limités par la taille de la charge utile totale de l'événement, plutôt que par la taille du lot :

- Machine d'état Step Functions (262 144 caractères)

- Fonction Lambda (6 Mo)

Défaillance partielle d’un lot

Pour Amazon SQS et les sources de flux, telles que Kinesis et DynamoDB, Pipes prend en charge la gestion des défaillances partielles par lots en EventBridge cas de défaillance cible. Si la cible prend en charge le traitement par lots et que seule une partie du lot aboutit, réessaie EventBridge automatiquement de regrouper le reste de la charge utile. Pour le contenu le plus up-to-date enrichi, cette nouvelle tentative s'effectue sur l'ensemble du canal, y compris en réinvoquant tout enrichissement configuré.

La gestion des défaillances partielles de lot pour l’enrichissement n’est pas prise en charge.

Pour les cibles Lambda et Step Functions, vous pouvez également spécifier une défaillance partielle en renvoyant une charge utile avec une structure définie depuis la cible. Cela indique les événements qui doivent faire l’objet d’une nouvelle tentative.

Exemple de structure de charge utile en cas de défaillance partielle

```
{
  "batchItemFailures": [
    {
      "itemIdentifier": "id2"
    },
    {
      "itemIdentifier": "id4"
    }
]
```

Dans l’exemple, `itemIdentifier` correspond à l’identifiant des événements gérés par votre cible à partir de leur source d’origine. Pour Amazon SQS, il s’agit de `messageId`. Pour Kinesis et DynamoDB, il s’agit de `eventID`. Pour que EventBridge Pipes puisse gérer correctement les défaillances partielles des lots provenant des cibles, ces champs doivent être inclus dans toute charge utile du tableau renvoyée par l'enrichissement.

Comportement du débit et de la simultanéité

Chaque événement ou lot d’événements reçu par un canal en direction d’un enrichissement ou d’une cible est considéré comme une exécution de canal. Un canal à l’état `STARTED` interroge en

permanence les événements provenant de la source, en augmentant ou en diminuant en fonction du backlog disponible et des paramètres de traitement par lots configurés.

Pour en savoir plus sur les quotas pour les exécutions de canal simultanées et le nombre de canaux par compte et par région, consultez ???.

Par défaut, un seul canal est mis à l'échelle en fonction du nombre maximal d'exécutions simultanées suivant, selon la source :

- DynamoDB : les exécutions simultanées peuvent atteindre la valeur de `ParallelizationFactor` configurée sur le canal multipliée par le nombre de partitions dans le flux.
- Apache Kafka : les exécutions simultanées peuvent atteindre le nombre de partitions sur la rubrique, c'est-à-dire 1 000 au maximum.
- Kinesis : les exécutions simultanées peuvent atteindre la valeur de `ParallelizationFactor` configurée sur le canal multipliée par le nombre de partitions dans le flux.
- Amazon MQ : 5
- Amazon SQS : 1 250

Si vous devez augmenter le débit d'interrogation maximal ou les limites de simultanéité, contactez l'assistance.

> **Note**
>
> Les limites d'exécution sont considérées comme des restrictions optimales en matière de sécurité. Bien que l'interrogation ne soit pas limitée au-dessous de ces valeurs, un canal ou un compte peut dépasser ces valeurs recommandées.

Les exécutions de canal sont limitées à 5 minutes maximum, enrichissement et traitement de la cible inclus. Cette limite ne peut pas être augmentée pour le moment.

Les canaux dont les sources sont strictement ordonnées (telles que les files d'attente FIFO Amazon SQS, Kinesis et DynamoDB Streams, ou les rubriques Apache Kafka) sont également limités en termes de simultanéité par la configuration de la source, telle que le nombre d'ID de groupe de messages pour les files d'attente FIFO ou le nombre de partitions pour les files d'attente Kinesis. Étant donné que l'ordre est strictement garanti dans le cadre de ces contraintes, un canal dont la source est ordonnée ne peut pas dépasser ces limites de simultanéité.

Transformation d'entrée Amazon EventBridge Pipes

Amazon EventBridge Pipes prend en charge les transformateurs d'entrée facultatifs lors du transfert de données à l'enrichissement et à la cible. Vous pouvez utiliser des transformateurs d'entrée pour remodeler la charge utile d'entrée des événements JSON afin de répondre aux besoins du service d'enrichissement ou de cible. Pour Amazon API Gateway et les destinations d'API, voici comment modeler l'événement d'entrée dans le modèle RESTful de votre API. Les transformateurs d'entrée sont modélisés sous forme de paramètre `InputTemplate`. Il peut s'agir de texte libre, d'un chemin JSON vers la charge utile de l'événement ou d'un objet JSON qui inclut des chemins JSON en ligne vers la charge utile de l'événement. Pour l'enrichissement, la charge utile de l'événement provient de la source. Pour les cibles, la charge utile de l'événement correspond à ce qui est renvoyé par l'enrichissement, si une telle charge est configurée sur le canal. Outre les données propres au service incluses dans la charge utile de l'événement, vous pouvez utiliser des variables réservées dans `InputTemplate` pour référencer les données de votre canal.

Pour accéder aux éléments d'un tableau, utilisez la notation entre crochets.

> **Note**
>
> EventBridge ne prend pas en charge l'ensemble de la syntaxe du chemin JSON et l'évalue lors de l'exécution. La syntaxe prise en charge inclut :
>
> - notation par points (par exemple, `$.detail`)
> - tirets
> - traits de soulignement
> - caractères alphanumériques
> - index de tableau
> - caractères génériques (*)

Voici des exemples de paramètres `InputTemplate` faisant référence à la charge utile d'un événement Amazon SQS :

Chaîne statique

```
InputTemplate: "Hello, sender"
```

Chemin JSON

```
InputTemplate: <$.attributes.SenderId>
```

Chaîne dynamique

```
InputTemplate: "Hello, <$.attributes.SenderId>"
```

JSON statique

```
InputTemplate: >
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3",
}
```

JSON dynamique

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.key>,
  "d": <aws.pipes.event.ingestion-time>
}
```

Utilisation de la notation entre crochets pour accéder à un élément dans un tableau :

```
InputTemplate: >
{
  "key1": "value1"
  "key2": <$.body.Records[3]>,
  "d": <aws.pipes.event.ingestion-time>
}
```

> **Note**
>
> EventBridge remplace les transformateurs d'entrée lors de l'exécution pour garantir une sortie JSON valide. Pour cette raison, placez des guillemets autour des variables qui font référence aux paramètres de chemin JSON, mais ne placez pas de guillemets autour des variables qui font référence à des objets ou des tableaux JSON.

Variables réservées

Les modèles d'entrée peuvent utiliser les variables réservées suivantes :

- `<aws.pipes.pipe-arn>` : Amazon Resource Name (ARN) du canal.
- `<aws.pipes.pipe-name>` : nom du canal.
- `<aws.pipes.source-arn>` : ARN de la source d'événement du canal.
- `<aws.pipes.enrichment-arn>` : ARN de l'enrichissement du canal.
- `<aws.pipes.target-arn>` : ARN de la cible du canal.
- `<aws.pipes.event.ingestion-time>` : heure à laquelle l'événement a été reçu par le transformateur d'entrée. Il s'agit d'un horodatage ISO 8601. Cette durée est différente pour le transformateur d'entrée de l'enrichissement et le transformateur d'entrée de la cible, selon le moment où l'enrichissement a terminé de traiter l'événement.
- `<aws.pipes.event>` : événement tel qu'il a été reçu par le transformateur d'entrée.

  Pour un transformateur d'entrée d'enrichissement, il s'agit de l'événement provenant de la source. Il contient la charge utile d'origine de la source, ainsi que des métadonnées supplémentaires propres au service. Pour obtenir des exemples propres à un service, consultez ???.

  Pour un transformateur d'entrée de cible, il s'agit de l'événement renvoyé par l'enrichissement, s'il est configuré, sans métadonnées supplémentaires. En tant que telle, une charge utile renvoyée par l'enrichissement peut ne pas être au format JSON. Si aucun enrichissement n'est configuré sur le canal, il s'agit de l'événement provenant de la source avec des métadonnées.
- `<aws.pipes.event.json>` : identique à `aws.pipes.event`, mais la variable n'a de valeur que si la charge utile d'origine, provenant de la source ou renvoyée par l'enrichissement, est au format JSON. Si le canal contient un champ codé, tel que le champ `body` Amazon SQS ou `data` Kinesis, ces champs sont décodés et convertis au format JSON valide. Comme elle n'est pas mise en échappement, la variable ne peut être utilisée que comme valeur pour un champ JSON. Pour de plus amples informations, veuillez consulter ???.

Exemple de transformation d'entrée

Voici un exemple d'événement Amazon EC2 que nous pouvons utiliser comme exemple d'événement.

```
{
  "version": "0",
  "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-abcd1111"
  ],
  "detail": {
    "instance-id": "i-0123456789",
    "state": "RUNNING"
  }
}
```

Utilisons le code JSON suivant comme transformateur.

```
{
  "instance" : <$.detail.instance-id>,
  "state": <$.detail.state>,
  "pipeArn" : <aws.pipes.pipe-arn>,
  "pipeName" : <aws.pipes.pipe-name>,
  "originalEvent" : <aws.pipes.event.json>
}
```

La sortie générée est la suivante :

```
{
  "instance" : "i-0123456789",
  "state": "RUNNING",
  "pipeArn" : "arn:aws:pipe:us-east-1:123456789012:pipe/example",
  "pipeName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

Analyse implicite des données de corps

Les champs suivants de la charge utile entrante peuvent être mis en échappement dans le code JSON, tels que l'objet Amazon SQS `body`, ou codés en base64, tels que l'objet Kinesis `data`. Pour le filtrage et la transformation d'entrée, EventBridge transforme ces champs au format JSON valide afin que les sous-valeurs puissent être référencées directement. Par exemple, `<$.data.someKey>` pour Kinesis.

Pour que la cible reçoive la charge utile d'origine sans aucune métadonnée supplémentaire, utilisez un transformateur d'entrée avec ces données de corps, propres à la source. Par exemple, `<$.body>` pour Amazon SQS ou `<$.data>` pour Kinesis. Si la charge utile d'origine est une chaîne JSON valide (par exemple, `{"key": "value"}`), l'utilisation du transformateur d'entrée avec des données de corps propres à la source entraînera la suppression des guillemets contenus dans la charge utile source d'origine. Par exemple, `{"key": "value"}` deviendra `"{key: value}"` une fois livré à la cible. Si votre cible requiert des charges utiles JSON valides (par exemple, EventBridge Lambda ou Step Functions), cela entraînera un échec de livraison. Pour que la cible reçoive les données source d'origine sans générer de code JSON non valide, encapsulez le transformateur d'entrée des données de corps source au format JSON. Par exemple, `{"data": <$.data>}`.

L'analyse implicite du corps peut également être utilisée pour renseigner dynamiquement les valeurs de la plupart des paramètres de cible ou d'enrichissement du canal. Pour de plus amples informations, veuillez consulter ???.

> **Note**
>
> Si la charge utile d'origine est au format JSON valide, ce champ contient le code JSON non mis en échappement et non codé en base64. Toutefois, si la charge utile n'est pas au format JSON valide, EventBridge code en base64 les champs répertoriés ci-dessous, à l'exception d'Amazon SQS.

- Active MQ : `data`
- Kinesis : `data`
- Amazon MSK : `key` et `value`
- Rabbit MQ : `data`
- Apache Kafka autogéré : `key` et `value`
- Amazon SQS : `body`

Problèmes courants liés à la transformation d'entrée

Les problèmes courants qui se produisent lors de la transformation d'entrées dans EventBridge Pipes sont les suivants :

- Pour les chaînes, des guillemets sont requis.
- Il n'y a pas de validation lors de la création du chemin JSON pour votre modèle.
- Si vous spécifiez une variable à mettre en correspondance avec un chemin JSON qui n'existe pas dans l'événement, cette variable n'est pas créée et n'apparaîtra pas dans la sortie.
- Les propriétés JSON telles que `aws.pipes.event.json` ne peuvent être utilisées que comme valeur d'un champ JSON, et non en ligne dans d'autres chaînes.
- EventBridge ne met pas en échappement les valeurs extraites par le chemin d'entrée lors du remplissage du modèle d'entrée pour une cible.
- Si un chemin JSON fait référence à un objet ou un tableau JSON, mais que la variable est référencée dans une chaîne, EventBridge supprime tous les guillemets internes pour garantir la validité de la chaîne. Par exemple, "Body is **<**$.body**>**" obligerait EventBridge à supprimer les guillemets de l'objet.

  Par conséquent, si vous souhaitez générer un objet JSON basé sur une seule variable de chemin JSON, vous devez le placer sous forme de clé. Dans cet exemple, `{"body": <$.body>}`.
- Les guillemets ne sont pas obligatoires pour les variables qui représentent des chaînes. Ils sont autorisés, mais EventBridge Pipes ajoute automatiquement des guillemets aux valeurs des variables de chaîne pendant la transformation, afin de garantir que le résultat de la transformation est un code JSON valide. EventBridge Pipes n'ajoute pas de guillemets aux variables qui représentent des objets ou des tableaux JSON. N'ajoutez pas de guillemets pour les variables qui représentent des objets ou des tableaux JSON.

  Par exemple, le modèle d'entrée suivant inclut des variables qui représentent à la fois des chaînes et des objets JSON :

```
{
  "pipeArn" : <aws.pipes.pipe-arn>,
  "pipeName" : <aws.pipes.pipe-name>,
  "originalEvent" : <aws.pipes.event.json>
}
```

  Le résultat est un code JSON valide avec des guillemets appropriés :

```
{
  "pipeArn" : "arn:aws:events:us-east-2:123456789012:pipe/example",
  "pipeName" : "example",
  "originalEvent" : {
    ... // commented for brevity
  }
}
```

- Pour les enrichissements ou les cibles Lambda ou Step Functions, les lots sont livrés à la cible sous forme de tableaux JSON, même si la taille du lot est de 1. Cependant, les transformateurs d'entrée seront toujours appliqués aux enregistrements individuels du tableau JSON, et non au tableau dans son ensemble. Pour de plus amples informations, veuillez consulter ???.

Connectez-vous à Amazon EventBridge Pipes

EventBridge La journalisation des canaux vous permet de demander à EventBridge Pipes d'envoyer des enregistrements détaillant les performances des canaux aux AWS services pris en charge. Utilisez les journaux pour en savoir plus sur les performances d'exécution de votre canal et pour faciliter le dépannage et le débogage.

Vous pouvez sélectionner les AWS services suivants comme destinations de log vers lesquelles EventBridge Pipes livre des enregistrements :

- CloudWatch Journaux

  EventBridge fournit des enregistrements de journaux au groupe de CloudWatch journaux de journaux spécifié.

  Utilisez CloudWatch les journaux pour centraliser les journaux de tous les systèmes, applications et AWS services que vous utilisez, au sein d'un seul service hautement évolutif. Pour plus d'informations, consultez la section Utilisation des groupes de journaux et des flux de CloudWatch journaux dans le guide de l'utilisateur Amazon Logs.
- Journaux de diffusion de Firehose

  EventBridge fournit des enregistrements de journal à un flux de diffusion Firehose.

  Amazon Data Firehose est un service entièrement géré qui fournit des données de streaming en temps réel à des destinations telles que certains AWS services, ainsi qu'à tout point de terminaison HTTP personnalisé ou à tout point de terminaison HTTP appartenant à des fournisseurs de

services tiers pris en charge. Pour plus d'informations, consultez la section Création d'un flux de diffusion Amazon Data Firehose dans le guide de l'utilisateur d'Amazon Data Firehose.

- Journaux Amazon S3

  EventBridge fournit les enregistrements de journal sous forme d'objets Amazon S3 au compartiment spécifié.

  Amazon S3 est un service de stockage d'objets qui offre une évolutivité, une disponibilité des données, une sécurité et des performances de pointe. Pour plus d'informations, consultez Chargement, téléchargement et utilisation des objets dans Amazon S3 dans le Guide de l'utilisateur Amazon Simple Storage Service.

Comment fonctionne la journalisation sur Amazon EventBridge Pipes

Une exécution de canal est un événement ou un lot d'événements reçu par un canal en direction d'un enrichissement et/ou d'une cible. Si cette option est activée, EventBridge génère un enregistrement de journal pour chaque étape d'exécution effectuée lors du traitement du lot d'événements. Les informations contenues dans l'enregistrement s'appliquent au lot d'événements, qu'il s'agisse d'un événement unique ou de 10 000 événements au maximum.

Vous pouvez configurer la taille du lot d'événements sur la source et la cible du canal. Pour plus d'informations, consultez ???.

Les données d'enregistrement envoyées à chaque destination de journal sont les mêmes.

Si une destination Amazon CloudWatch Logs est configurée, les enregistrements de journal envoyés à toutes les destinations sont limités à 256 Ko. Les champs seront tronqués si nécessaire.

Vous pouvez personnaliser les enregistrements EventBridge envoyés aux destinations de journal sélectionnées de la manière suivante :

- Vous pouvez spécifier le niveau de journalisation, qui détermine les étapes d'exécution pour lesquelles les EventBridge enregistrements sont envoyés aux destinations de journalisation sélectionnées. Pour plus d'informations, consultez ???.
- Vous pouvez spécifier si EventBridge Pipes inclut les données d'exécution dans les enregistrements des étapes d'exécution lorsque cela est pertinent. Ces données comprennent :
  - La charge utile du lot d'événements
  - La demande envoyée au service d' AWS enrichissement ou au service cible

- La réponse renvoyée par le service AWS d'enrichissement ou le service cible

Pour plus d’informations, consultez ???.

Spécification du niveau de journalisation des EventBridge tuyaux

Vous pouvez spécifier les types d'étapes d'exécution pour lesquelles des EventBridge enregistrements sont envoyés aux destinations de journal sélectionnées.

Choisissez parmi les niveaux de détail suivants à inclure dans les enregistrements de journal. Le niveau de journalisation s’applique à toutes les destinations de journal spécifiées pour le canal. Chaque niveau de journalisation inclut les étapes d’exécution des niveaux de journalisation précédents.

- OFF — EventBridge n'envoie aucun enregistrement vers les destinations de journal spécifiées. Il s’agit du paramètre par défaut.
- ERREUR — EventBridge envoie tous les enregistrements relatifs aux erreurs générées lors de l'exécution du canal vers les destinations de journal spécifiées.
- INFO — EventBridge envoie tous les enregistrements relatifs aux erreurs, et sélectionne les autres étapes effectuées lors de l'exécution du canal vers les destinations de journal spécifiées.
- TRACE — EventBridge envoie tous les enregistrements générés au cours de n'importe quelle étape de l'exécution du canal vers les destinations de journal spécifiées.

Dans la EventBridge console, CloudWatch les journaux sont sélectionnés comme destination par défaut, tout comme le niveau du `ERROR` journal. Ainsi, par défaut, EventBridge Pipes crée un nouveau groupe de CloudWatch journaux auquel il envoie des enregistrements contenant le `ERROR` niveau de détail. Aucune valeur par défaut n’est sélectionnée lorsque vous configurez les journaux par programmation.

Le tableau suivant répertorie les étapes d’exécution incluses dans chaque niveau de journalisation.

Étape	TRACE	INFO	ERROR	OFF
Échec de l'exécution	x	h/24, j/7	x	
Échec partiel de l’exécution	x	h/24, j/7	x	

Étape	TRACE	INFO	ERROR	OFF
Exécution commencée	x	x		
Réussite de l'exécution	x	x		
Exécution limitée	x	h/24, j/7	x	
Execution Timeout (Délai d'exécution)	x	h/24, j/7	x	
Échec de l’invocation d’enrichissement	x	h/24, j/7	x	
Invocation d’enrichissement ignorée	x	x		
L’invocation d’enrichissement a démarré	x			
Réussite de l’invocation d’enrichissement	x			
Début de l’étape d’enrichissement	x	x		
Échec de l’étape d’enrichissement	x	h/24, j/7	x	
Réussite de l’étape d’enrichissement	x	x		
Echec de la transformation d’enrichissement	x	h/24, j/7	x	
La transformation d’enrichissement a démarré	x			
Réussite de la transformation d’enrichissement	x			

Étape	TRACE	INFO	ERROR	OFF
Échec de l’invocation de la cible	x	h/24, j/7	x	
Échec partiel de l’invocation de la cible	x	h/24, j/7	x	
Invocation de la cible ignorée	x			
L’invocation de la cible a démarré	x			
Réussite de l’invocation de la cible	x			
Début de l’étape de la cible	x	x		
Échec de l’étape de la cible	x	h/24, j/7	x	
Échec partiel de l’étape de la cible	x	h/24, j/7	x	
Étape de la cible ignorée	x			
Réussite de l’étape de la cible	x	x		
Échec de la transformation de la cible	x	h/24, j/7	x	
La transformation de la cible a démarré	x			
Réussite de la transformation de la cible	x			

Inclure les données d'exécution dans les logs de EventBridge Pipes

Vous pouvez spécifier EventBridge pour inclure les données d'exécution dans les enregistrements qu'il génère. Les données d’exécution incluent des champs représentant la charge utile du lot

d’événements, ainsi que la demande envoyée à l’enrichissement et à la cible et la réponse de ces derniers.

Les données d’exécution sont utiles pour le dépannage et le débogage. Le champ `payload` contient le contenu réel de chaque événement inclus dans le lot, ce qui vous permet de corréler des événements individuels à une exécution de canal spécifique.

Si vous choisissez d’inclure les données d’exécution, elles sont incluses pour toutes les destinations de journal spécifiées pour le canal.

> **Important**
>
> Ces champs peuvent contenir des informations sensibles. EventBridge ne tente pas de supprimer le contenu de ces champs lors de la journalisation.

Lorsque vous incluez des données d'exécution, EventBridge ajoutez les champs suivants aux enregistrements concernés :

- **`payload`**

  Représente le contenu du lot d’événements traité par le canal.

  EventBridge inclut le `payload` champ dans les enregistrements générés aux étapes où le contenu du lot d'événements peut avoir été mis à jour. Ces étapes sont les suivantes :

  - `EXECUTION_STARTED`
  - `ENRICHMENT_TRANSFORMATION_SUCCEEDED`
  - `ENRICHMENT_STAGE_SUCCEEDED`
  - `TARGET_TRANSFORMATION_SUCCEEDED`
  - `TARGET_STAGE_SUCCEEDED`
- **`awsRequest`**

  Représente la demande envoyée à l’enrichissement ou à la cible sous forme de chaîne JSON. Pour les demandes envoyées à une destination d’API, il s’agit de la requête HTTP envoyée à ce point de terminaison.

  EventBridge inclut le `awsRequest` champ dans les enregistrements générés lors des dernières étapes d'enrichissement et de ciblage, c'est-à-dire après avoir EventBridge exécuté ou tenté

d'exécuter la demande par rapport à l'enrichissement ou au service cible spécifié. Ces étapes sont les suivantes :

- `ENRICHMENT_INVOCATION_FAILED`
- `ENRICHMENT_INVOCATION_SUCCEEDED`
- `TARGET_INVOCATION_FAILED`
- `TARGET_INVOCATION_PARTIALLY_FAILED`
- `TARGET_INVOCATION_SUCCEEDED`

- **`awsResponse`**

  Représente la réponse renvoyée par l'enrichissement ou la cible, au format JSON. Pour les demandes envoyées à une destination d'API, il s'agit de la réponse HTTP renvoyée par ce point de terminaison.

  De mêmeawsRequest, EventBridge inclut le `awsResponse` champ dans les enregistrements générés lors des dernières étapes de l'enrichissement et du ciblage, c'est-à-dire après avoir EventBridge exécuté ou tenté d'exécuter une demande concernant le service d'enrichissement ou le service cible spécifié et reçu une réponse. Ces étapes sont les suivantes :

  - `ENRICHMENT_INVOCATION_FAILED`
  - `ENRICHMENT_INVOCATION_SUCCEEDED`
  - `TARGET_INVOCATION_FAILED`
  - `TARGET_INVOCATION_PARTIALLY_FAILED`
  - `TARGET_INVOCATION_SUCCEEDED`

Pour obtenir une description des étapes d'exécution de canal, consultez ???.

Tronquer les données d'exécution dans les enregistrements du journal EventBridge Pipes

Si vous choisissez d' EventBridge inclure les données d'exécution dans les enregistrements du journal d'un canal, il est possible qu'un enregistrement dépasse la limite de 256 Ko. Pour éviter cela, tronque EventBridge automatiquement les champs de données d'exécution, dans l'ordre suivant. EventBridge tronque entièrement chaque champ avant de tronquer le champ suivant. EventBridge tronque les données des champs simplement en supprimant des caractères à la fin de la chaîne de données ; aucune tentative n'est faite pour tronquer en fonction de l'importance des données, et la troncature invalidera le formatage JSON.

- `payload`
- `awsRequest`
- `awsResponse`

Si EventBridge des champs sont tronqués dans l'événement, le `truncatedFields` champ inclut une liste des champs de données tronqués.

Signalement d'erreurs dans les enregistrements du journal EventBridge Pipes

EventBridge inclut également les données d'erreur, lorsqu'elles sont disponibles, dans les étapes d'exécution du canal qui représentent les états de défaillance. Les étapes sont les suivantes :

- `ExecutionThrottled`
- `ExecutionTimeout`
- `ExecutionFailed`
- `ExecutionPartiallyFailed`
- `EnrichmentTransformationFailed`
- `EnrichmentInvocationFailed`
- `EnrichmentStageFailed`
- `TargetTransformationFailed`
- `TargetInvocationFailed`
- `TargetInvocationPartiallyFailed`
- `TargetStageFailed`
- `TargetStagePartiallyFailed`

EventBridge Étapes d'exécution des tuyaux

En comprenant le déroulement des étapes d’exécution d’un canal, vous pourrez mieux résoudre les problèmes liés aux performances de votre canal ou les déboguer à l’aide de journaux.

Une exécution de canal est un événement ou un lot d’événements reçu par un canal en direction d’un enrichissement ou d’une cible. Si cette option est activée, EventBridge génère un enregistrement de journal pour chaque étape d'exécution effectuée lors du traitement du lot d'événements.

Globalement, l'exécution comporte deux étapes ou ensemble d'étapes : l'enrichissement et la cible. Chacune de ces étapes comprend des étapes de transformation et d'invocation.

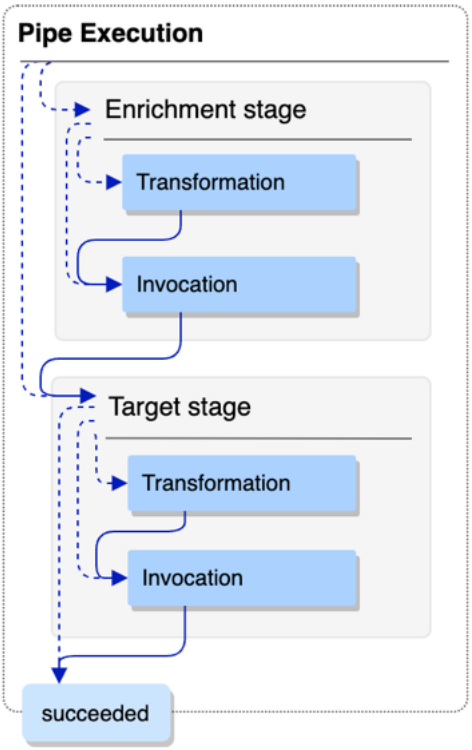
Les principales étapes d'une exécution de canal réussie sont les suivantes :

- L'exécution du canal démarre.
- L'exécution passe à l'étape de l'enrichissement si vous avez spécifié un enrichissement pour les événements. Si vous n'avez pas spécifié d'enrichissement, l'exécution passe à l'étape de la cible.

  Au cours de l'étape d'enrichissement, le canal effectue la transformation que vous avez spécifiée, puis invoque l'enrichissement.
- Au cours de l'étape de cible, le canal effectue la transformation que vous avez spécifiée, puis invoque la cible.

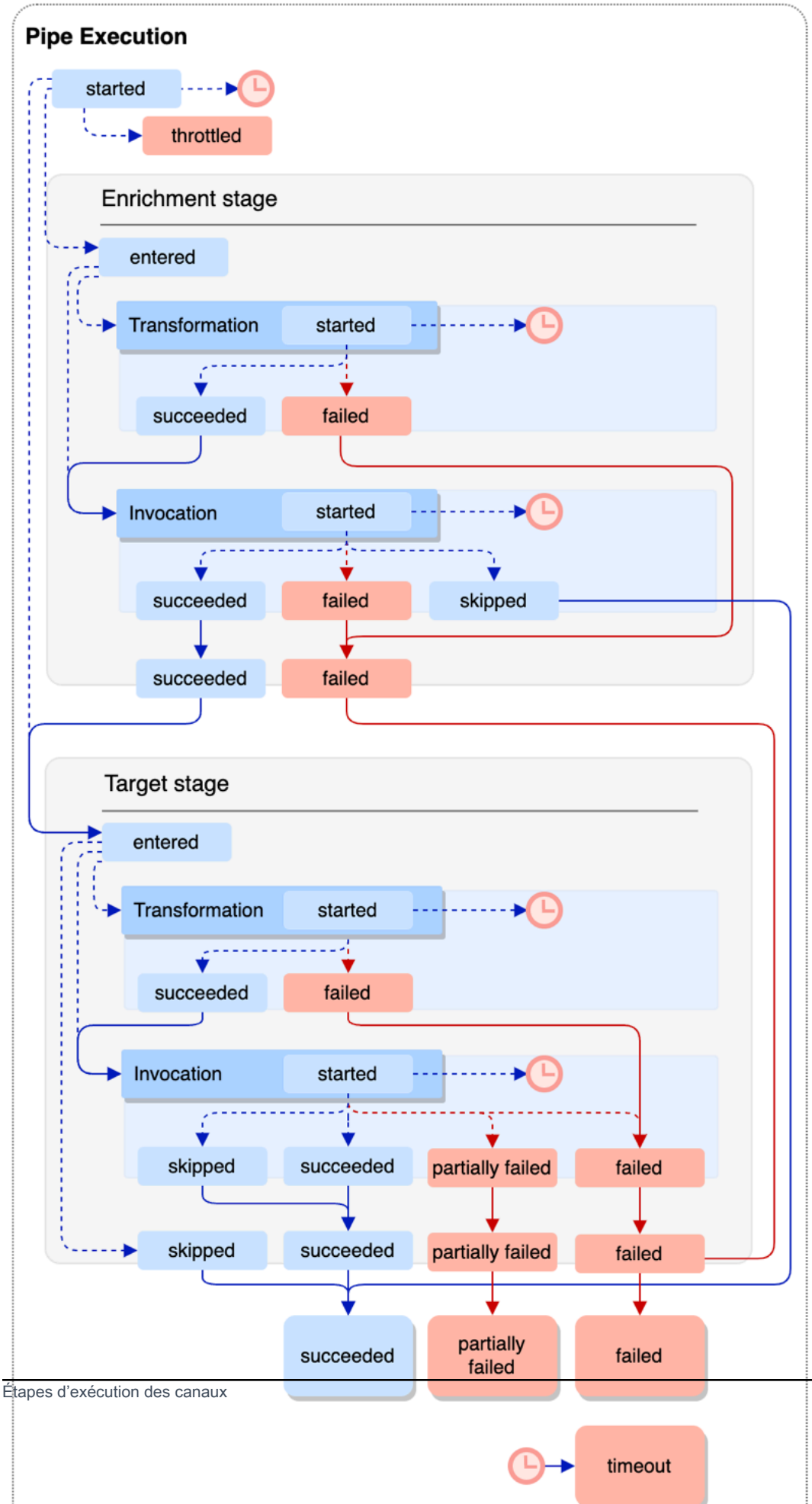
  Si vous n'avez pas spécifié de transformation ou de cible, l'exécution ignore l'étape de la cible.
- L'exécution du canal se termine avec succès.

Le schéma ci-dessous illustre le déroulement des étapes. Les lignes pointillées représentent les chemins divergents.



Le schéma ci-dessous présente une vue détaillée du flux d’exécution d’un canal, avec toutes les étapes d’exécution possibles représentées. Les lignes pointillées représentent à nouveau les chemins divergents.

Pour obtenir la liste complète des étapes d’exécution d’un canal, consultez ???.


Pipe Execution
started
throttled
Enrichment stage
entered
Transformation
started
succeeded
failed
Invocation
started
succeeded
failed
skipped
succeeded
failed
Target stage
entered
Transformation
started
succeeded
failed
Invocation
started
skipped
succeeded
partially failed
failed
skipped
succeeded
partially failed
failed
succeeded
partially failed
failed
timeout

Notez que l'invocation de la cible peut entraîner une défaillance partielle du lot. Pour plus d'informations, consultez ???.

EventBridge Référence du schéma du journal des tuyaux

La référence suivante détaille le schéma des enregistrements du journal EventBridge Pipes.

Chaque enregistrement de journal représente une étape d'exécution du canal et peut contenir jusqu'à 10 000 événements si la source et la cible du canal ont été configurées pour le traitement par lots.

Pour plus d'informations, consultez ???.

```
{
    "executionId": "guid",
    "timestamp": "date_time",
    "messageType": "execution_step",
    "resourceArn": "arn:aws:pipes:region:account:pipe/pipe-name",
    "logLevel": "TRACE | INFO | ERROR",
    "payload": "{}",
    "awsRequest": "{}"
    "awsResponse":"{}"
    "truncatedFields": ["awsRequest","awsResponse","payload"],
    "error": {
        "httpStatusCode": code,
        "message": "error_message",
        "details": "",
        "awsService": "service_name",
        "requestId": "service_request_id"
    }
}
```

executionId

ID de l'exécution de canal.

Une exécution de canal est un événement ou un lot d'événements reçu par un canal en direction d'un enrichissement ou d'une cible. Pour plus d'informations, consultez ???.

timestamp

Date et heure auxquelles l'événement de journal a été émis.

Unité : milliseconde

messageType

Étape d’exécution du canal pour laquelle l’enregistrement a été généré.

Pour plus d’informations sur les étapes d’exécution d’un canal, consultez ???.

resourceArn

Amazon Resource Name (ARN) du canal.

logLevel

Niveau de détail spécifié pour le journal du canal.

Valeurs valides : `ERROR` | `INFO` | `TRACE`

Pour plus d’informations, consultez ???.

payload

Contenu du lot d’événements traité par le canal.

EventBridge inclut ce champ uniquement si vous avez spécifié d'inclure les données d'exécution dans les journaux de ce canal. Pour plus d’informations, consultez ???.

> **Important**
>
> Ces champs peuvent contenir des informations sensibles. EventBridge ne tente pas de supprimer le contenu de ces champs lors de la journalisation.

Pour plus d’informations, consultez ???.

awsRequest

Demande envoyée à l’enrichissement ou à la cible, au format JSON. Pour les demandes envoyées à une destination d’API, il s’agit de la requête HTTP envoyée à ce point de terminaison.

EventBridge inclut ce champ uniquement si vous avez spécifié d'inclure les données d'exécution dans les journaux de ce canal. Pour plus d’informations, consultez ???.

> **Important**
>
> Ces champs peuvent contenir des informations sensibles. EventBridge ne tente pas de supprimer le contenu de ces champs lors de la journalisation.

Pour plus d’informations, consultez ???.

awsResponse

Réponse renvoyée par l’enrichissement ou la cible, au format JSON. Pour les demandes envoyées à une destination d’API, il s’agit de la réponse HTTP renvoyée par ce point de terminaison, pas la réponse renvoyée par le service de destination d’API lui-même.

EventBridge inclut ce champ uniquement si vous avez spécifié d'inclure les données d'exécution dans les journaux de ce canal. Pour plus d’informations, consultez ???.

> **Important**
>
> Ces champs peuvent contenir des informations sensibles. EventBridge ne tente pas de supprimer le contenu de ces champs lors de la journalisation.

Pour plus d’informations, consultez ???.

truncatedFields

La liste de tous les champs de données d'exécution EventBridge a été tronquée pour maintenir l'enregistrement en dessous de la limite de taille de 256 Ko.

S'il EventBridge n'est pas nécessaire de tronquer aucun des champs de données d'exécution, ce champ est présent mais`null`.

Pour plus d’informations, consultez ???.

error

Contient des informations relatives aux erreurs générées lors de cette étape d’exécution du canal.

Si aucune erreur n’a été générée lors de cette étape d’exécution du canal, ce champ est présent mais `null`.

httpStatusCode

Code de statut HTTP renvoyé par le service appelé.

message

Message d’erreur renvoyé par le service appelé.

details

Toute information d’erreur détaillée renvoyée par le service appelé.

awsService

Nom du service appelé.

requestId

ID de cette demande provenant du service appelé.

Enregistrement et surveillance d'Amazon EventBridge Pipes à l'aide AWS CloudTrail d'Amazon CloudWatch Logs

Vous pouvez enregistrer EventBridge les appels de Pipes, utiliser CloudTrail et surveiller l'état de vos canaux à l'aide CloudWatch de métriques.

CloudWatch métriques

EventBridge Pipes envoie des métriques à Amazon CloudWatch toutes les minutes, qu'il s'agisse d'une exécution limitée d'un canal ou d'une cible invoquée avec succès.

Métrique	Description	Dimensions	Unités
`Concurrency`	Nombre d’exécutions simultanées d’un canal.	AwsAccountId	Aucun
`Duration`	Durée d’exécution du canal.	PipeName	Millisecondes
`EventCount`	Nombre d’événements traités par un canal.	PipeName	Aucun
`EventSize`	Taille de la charge utile de l’événement qui a invoqué le canal.	PipeName	Octets
`ExecutionThrottled`	Nombre d’exécutions d’un canal qui ont été limitées. **Note** Cette valeur est de `0` si aucune exécution n’a été limitée.	AwsAccountId, PipeName	Aucun

Métrique	Description	Dimensions	Unités
`ExecutionTimeout`	Nombre d'exécutions d'un canal qui ont expiré avant la fin de l'exécution. **Note** Cette valeur est de `0` si aucune exécution n'a expiré.	PipeName	Aucun
`ExecutionFailed`	Nombre d'exécutions d'un canal ayant échoué. **Note** Cette valeur est de `0` si aucune exécution n'a échoué.	PipeName	Aucun
`ExecutionPartiallyFailed`	Nombre d'exécutions d'un canal ayant partiellement échoué. **Note** Cette valeur est de `0` si aucune exécution n'a partiellement échoué.	PipeName	Aucun
`EnrichmentStageDuration`	Durée de l'étape d'enrichissement.	PipeName	Millisecondes
`EnrichmentStageFailed`	Nombre d'exécutions de l'étape d'enrichissement d'un canal ayant échoué. **Note** Cette valeur est de `0` si aucune exécution n'a échoué.	PipeName	Aucun

Métrique	Description	Dimensions	Unités
`Invocations`	Nombre total d'invocations.	AwsAccountId, PipeName	Aucun
`TargetStageDuration`	Durée de l'étape de la cible.	PipeName	Millisecondes
`TargetStageFailed`	Nombre d'exécutions de l'étape de la cible d'un canal ayant échoué. **Note** Cette valeur est de `0` si aucune exécution n'a échoué.	PipeName	Aucun
`TargetStagePartiallyFailed`	Nombre d'exécutions de l'étape de la cible d'un canal ayant partiellement échoué. **Note** Cette valeur est de `0` si aucune exécution de l'étape de la cible n'a partiellement échoué.	PipeName	Aucun
`TargetStageSkipped`	Nombre d'exécutions de l'étape de la cible d'un canal qui ont été ignorées (en raison du renvoi d'une charge utile vide par l'enrichissement, par exemple).	PipeName	Nombre

Dimensions pour les CloudWatch métriques

CloudWatch les métriques ont des dimensions, ou des attributs triables, qui sont répertoriés ci-dessous.

Dimension	Description
`AwsAccountId`	Filtre les métriques disponibles par ID de compte.
`PipeName`	Filtre les métriques disponibles par nom de canal.

Gestion EventBridge des erreurs et résolution des problèmes liés à Amazon Pipes

Comportement de nouvelle tentative et gestion des erreurs

EventBridge Pipes réessaie automatiquement l'enrichissement et l'invocation de la cible en cas d' AWS échec réessayable avec le service source, le service d'enrichissement ou le service cible, ou. EventBridge Toutefois, en cas d'échec dû à la mise en œuvre de l'enrichissement ou du client cible, le débit d'interrogation du canal diminuera progressivement. En cas d'erreurs 4xx quasi continues (telles que les problèmes d'autorisation avec des ressources IAM ou manquantes), le canal peut être automatiquement désactivé et `StateReason` peut comporter un message explicatif.

Erreurs d'invocation du canal et comportement de nouvelle tentative

Lorsque vous invoquez un canal, deux principaux types d'erreurs peuvent se produire : les erreurs internes au canal et les erreurs d'invocation du client.

Erreurs internes au canal

Les erreurs internes à Pipes sont des erreurs résultant d'aspects de l'invocation gérés par le service EventBridge Pipes.

Ces types d'erreurs peuvent inclure les problèmes suivants :

- Échec de la connexion HTTP lors d'une tentative d'invocation du service client cible
- Baisse temporaire de la disponibilité sur le service de canal lui-même.

En général, EventBridge Pipes réessaie les erreurs internes un nombre indéfini de fois et ne s'arrête que lorsque l'enregistrement expire dans la source.

Pour les canaux dotés d'une source de flux, EventBridge Pipes ne compte pas les tentatives pour des erreurs internes par rapport au nombre maximum de tentatives spécifié dans la politique de nouvelles tentatives pour la source de flux. Pour les canaux contenant une source Amazon SQS, EventBridge Pipes ne compte pas les tentatives pour des erreurs internes par rapport au nombre maximal de réceptions pour la source Amazon SQS.

Erreurs d’invocation du client

Les erreurs d’invocation du client sont des erreurs résultant de la configuration ou du code géré par l’utilisateur.

Ces types d’erreurs peuvent inclure les problèmes suivants :

- Autorisations insuffisantes sur le canal pour invoquer la cible.
- Erreur logique dans un point de terminaison Lambda, Step Functions, de destination d’API ou API Gateway client invoqué de manière synchrone.

Pour les erreurs d'invocation du client, EventBridge Pipes procède comme suit :

- Pour les canaux dotés d'une source de flux, EventBridge Pipes réessaie jusqu'à la durée maximale de tentative définie dans la politique de relance des canaux ou jusqu'à l'expiration de l'âge maximum d'enregistrement, selon la première éventualité.
- Pour les canaux contenant une source Amazon SQS, EventBridge Pipes tente à nouveau de corriger une erreur client jusqu'au nombre maximal de destinataires dans la file d'attente source.
- Pour les canaux contenant une source Apache Kafka ou Amazon MQ EventBridge , réessaie les erreurs du client de la même manière que les erreurs internes.

Pour les canaux dotés de cibles de calcul, vous devez appeler le canal de manière synchrone pour que EventBridge Pipes soit conscient des erreurs d'exécution générées par la logique de calcul du client et réessaie de corriger ces erreurs. Pipes ne peut pas effectuer de nouvelle tentative en cas d’erreur émise par la logique d’un flux de travaux standard Step Functions, car cette cible doit être invoquée de manière asynchrone.

Pour Amazon SQS et les sources de flux, telles que Kinesis et DynamoDB, Pipes prend en charge la gestion des défaillances partielles par lots en EventBridge cas de défaillance cible. Pour plus d’informations, consultez Défaillance partielle d’un lot.

Comportement d’une DLQ de canal

Un canal hérite du comportement de file d’attente de lettres mortes (DLQ) de la source :

- Si la file d’attente Amazon SQS source comporte une DLQ configurée, les messages y sont automatiquement livrés une fois le nombre de tentatives spécifié atteint.
- Pour les sources de flux, telles que les flux DynamoDB et Kinesis, vous pouvez configurer une DLQ pour les événements de canal et de routage. Les sources de flux DynamoDB et Kinesis prennent en charge les files d’attente Amazon SQS et les rubriques Amazon SNS en tant que cibles de DLQ.

Si vous spécifiez `DeadLetterConfig` pour un canal avec une source Kinesis ou DynamoDB, assurez-vous que la propriété `MaximumRecordAgeInSeconds` du canal est inférieure à la propriété `MaximumRecordAge` de l’événement source. `MaximumRecordAgeInSeconds` contrôle le moment où l’interrogateur du canal abandonne l’événement et le livre à la DLQ. `MaximumRecordAge` contrôle la durée pendant laquelle le message reste visible dans le flux source avant d’être supprimé. Par conséquent, définissez `MaximumRecordAgeInSeconds` sur une valeur inférieure à la propriété `MaximumRecordAge` source afin que le délai entre le moment où l’événement est envoyé à la DLQ et le moment où il est automatiquement supprimé par la source soit suffisant pour que vous puissiez déterminer pourquoi l’événement a été envoyé à la DLQ.

Pour les sources Amazon MQ, la DLQ peut être configurée directement sur l’agent de messages.

EventBridge Pipes ne prend pas en charge les DLQ « premier entré, premier sorti » (FIFO) pour les sources de flux.

EventBridge Pipes ne prend pas en charge le DLQ pour les sources de flux Amazon MSK et les sources de flux Apache Kafka autogérées.

États de défaillance d’un canal

La création, la suppression et la mise à jour de canaux sont des opérations asynchrones qui peuvent entraîner un état de défaillance. De même, un canal peut être automatiquement désactivé en raison de défaillances. Dans tous les cas, la propriété `StateReason` du canal fournit des informations permettant de résoudre la défaillance.

Voici des exemples de valeurs possibles pour `StateReason` :

- Flux introuvable. Pour reprendre le traitement, supprimez le canal et créez-en un nouveau.

- Pipes ne dispose pas des autorisations requises pour effectuer des opérations de file d'attente (sqs :ReceiveMessage, sqs : DeleteMessage et sqs :) GetQueueAttributes
- Erreur de connexion. Votre VPC doit être capable de se connecter à des canaux. Vous pouvez fournir un accès en configurant une passerelle NAT ou un point de terminaison VPC pour canaliser les données. Pour savoir comment configurer une passerelle NAT ou un point de terminaison VPC pour canaliser les données, consultez la documentation. AWS
- Aucun groupe de sécurité n'est associé au cluster MSK.

Un canal peut être automatiquement arrêté avec la propriété `StateReason` mise à jour. Les raisons possibles sont les suivantes :

- Un flux de travaux standard Step Functions configuré en tant qu'enrichissement.
- Un flux de travaux standard Step Functions configuré en tant que cible à invoquer de manière synchrone.

Défaillances de chiffrement personnalisées

Si vous configurez une source pour utiliser une clé de chiffrement AWS KMS personnalisée (CMK) plutôt qu'une AWS KMS clé AWS gérée, vous devez explicitement autoriser le déchiffrement du rôle d'exécution de votre canal. Pour ce faire, incluez l'autorisation supplémentaire suivante dans la politique de clé CMK personnalisée :

```
  {
      "Sid": "Allow Pipes access",
      "Effect": "Allow",
      "Principal": {
          "AWS": "arn:aws:iam::01234567890:role/service-role/
Amazon_EventBridge_Pipe_DDBStreamSourcePipe_12345678"
      },
      "Action": "kms:Decrypt",
      "Resource": "*"
  }
```

Remplacez le rôle ci-dessus par le rôle d'exécution de votre canal.

Cela est vrai pour toutes les sources de tuyauterie dotées de la AWS KMS technologie CMK, notamment :

- Amazon DynamoDB Streams
- Amazon Kinesis Data Streams
- Amazon MQ
- Amazon MSK
- Amazon SQS

Didacticiel : création d'un canal EventBridge qui filtre les événements sources

Dans ce didacticiel, vous allez créer un canal qui connecte une source de flux DynamoDB à une cible de file d'attente Amazon SQS. Cela inclut la spécification d'un modèle d'événement pour le canal à utiliser lors du filtrage des événements destinés à être livrés à la file d'attente. Vous allez ensuite tester le canal pour vérifier que seuls les événements souhaités sont livrés.

Prérequis : créer la source et la cible

Avant de créer le canal, vous devez créer la source et la cible auxquelles le canal doit se connecter. Dans ce cas, un flux de données Amazon DynamoDB sert de source pour le canal et une file d'attente Amazon SQS sert de cible pour le canal.

Pour simplifier cette étape, vous pouvez utiliser AWS CloudFormation pour allouer les ressources de la source et de la cible. Pour ce faire, vous allez créer un modèle CloudFormation définissant les ressources suivantes :

- La source du canal

  Une table Amazon DynamoDB nommée `pipe-tutorial-source`, avec un flux permettant de fournir un flux ordonné d'informations sur les modifications apportées aux éléments de la table DynamoDB.
- La cible du canal

  Une file d'attente Amazon SQS nommée `pipe-tutorial-target`, destinée à recevoir le flux DynamoDB d'événements provenant de votre canal.

Pour créer le modèle CloudFormation pour allouer les ressources du canal

1. Copiez le modèle de texte JSON dans la section ??? ci-dessous.

2. Enregistrez le modèle sous forme de fichier JSON (par exemple, `~/pipe-tutorial-resources.json`).

Utilisez ensuite le modèle de fichier que vous venez de créer pour allouer une pile CloudFormation.

> **Note**
>
> Une fois que vous avez créé votre pile CloudFormation, les ressources AWS qu'elle alloue vous seront facturées.

Allocation des prérequis du didacticiel à l'aide de l'interface de ligne de commande AWS

- Exécutez la commande d'interface de ligne de commande suivante, où `--template-body` spécifie l'emplacement de votre modèle de fichier :

```
aws cloudformation create-stack --stack-name pipe-tuturial-resources --template-body file://~/pipe-tutorial-resources.json
```

Allocation des prérequis du didacticiel à l'aide de la console CloudFormation

1. Ouvrez la console AWS CloudFormation à l'adresse https://console.aws.amazon.com/cloudformation.
2. Sélectionnez Piles, puis Créer une pile et choisissez Avec de nouvelles ressources (standard).

   CloudFormation affiche l'assistant Créer une pile.
3. Pour Prérequis - Préparer le modèle, laissez Le modèle est prêt sélectionné par défaut.
4. Sous Spécifier un modèle, sélectionnez Charger un fichier de modèle, puis choisissez le fichier et sélectionnez Suivant.
5. Configurez la pile et les ressources qu'elle allouera :
   - Dans le champ Nom de la pile, saisissez `pipe-tuturial-resources`.
   - Pour Paramètres, conservez les noms par défaut de la table DynamoDB et de la file d'attente Amazon SQS.
   - Choisissez Next (Suivant).
6. Choisissez Suivant, puis Soumettre.

CloudFormation crée la pile et alloue les ressources définies dans le modèle.

Pour plus d'informations sur CloudFormation, consultez Qu'est-ce qu'AWS CloudFormation ? dans le Guide de l'utilisateur AWS CloudFormation.

Étape 1 : Créer le canal

Une fois la source et la cible du canal allouées, vous pouvez créer le canal pour connecter les deux services.

Création du canal à l'aide de la console EventBridge

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Pipelines.
3. Choisissez Créer un pipeline.
4. Pour Nom, nommez votre canal `pipe-tutorial`.
5. Spécifiez la source du flux de données DynamoDB :
   a. Sous Informations, dans Source, sélectionnez Flux de données DynamoDB.

      EventBridge affiche les paramètres de configuration de la source spécifiques à DynamoDB.
   b. Pour Flux DynamoDB, sélectionnez `pipe-tutorial-source`.

      Laissez Position de départ définie sur la valeur par défaut, `Latest`.
   c. Choisissez Next (Suivant).
6. Spécifiez et testez un modèle d'événement pour filtrer les événements :

   Le filtrage vous permet de contrôler les événements que les canaux envoient à l'enrichissement ou à la cible. Le canal envoie uniquement les événements qui correspondent au modèle d'événement à l'enrichissement ou à la cible.

   Pour de plus amples informations, veuillez consulter ???.

   > **Note**
   >
   > Seuls les événements envoyés à l'enrichissement ou à la cible vous sont facturés.

a. Sous Exemple d'événement - facultatif, laissez Événements AWS sélectionné et assurez-vous que Exemple d'événement de flux DynamoDB 1 est sélectionné.

   Il s'agit de l'exemple d'événement que vous allez utiliser pour tester notre modèle d'événement.

b. Sous Modèle d'événement, entrez le modèle d'événement suivant :

```
{
  "eventName": ["INSERT", "MODIFY"]
}
```

c. Choisissez Modèle de test.

   EventBridge affiche un message indiquant que l'exemple d'événement correspond au modèle d'événement. En effet, la valeur `eventName` de l'exemple d'événement est définie sur `INSERT`.

d. Choisissez Next (Suivant).

7. Choisissez Suivant pour ne pas spécifier d'enrichissement.

   Dans cet exemple, vous ne sélectionnerez pas d'enrichissement. Les enrichissements vous permettent de sélectionner un service pour améliorer les données provenant de la source avant de les envoyer à la cible. Pour en savoir plus, consultez ???.

8. Spécifiez votre file d'attente Amazon SQS en tant que cible de canal :

   a. Sous Informations, pour Service cible, sélectionnez File d'attente Amazon SQS.

   b. Pour File d'attente, sélectionnez `pipe-tutorial-target`.

   c. Laissez la section Transformateur d'entrée cible vide.

      Pour de plus amples informations, veuillez consulter ???.

9. Choisissez Créer un pipeline.

   EventBridge crée le canal et affiche la page détaillée du canal. Le canal est prêt une fois que son statut passe à `Running`.

Étape 2 : Confirmer que le canal filtre les événements

Le canal est configuré, mais n'a pas encore reçu les événements de la table.

Pour tester le canal, vous allez mettre à jour les entrées de la table DynamoDB. Chaque mise à jour génère des événements que le flux DynamoDB envoie à notre canal. Certains correspondront au modèle d'événement que vous avez spécifié, d'autres non. Vous pouvez ensuite examiner la file d'attente Amazon SQS pour vous assurer que le canal n'a livré que les événements correspondant à notre modèle d'événement.

Mise à jour des éléments de table pour générer des événements

1. Ouvrez la console DynamoDB à l'adresse https://console.aws.amazon.com/dynamodb/.
2. Dans le volet de navigation de gauche, sélectionnez Tables. Sélectionnez la table `pipe-tutorial-source`.

   DynamoDB affiche la page détaillée de la table `pipe-tutorial-source`.
3. Sélectionnez Explorer les éléments de table, puis choisissez Créer un élément.

   DynamoDB affiche la page Créer un élément.
4. Sous Attributs, créez un nouvel élément de table :
   a. Pour Album, entrez `Album A`.
   b. Pour Artiste, entrez `Artist A`.
   c. Choisissez Créer un élément.
5. Mettez à jour l'élément de table :
   a. Sous Éléments retournés, choisissez Album A.
   b. Sélectionnez Ajouter un nouvel attribut, puis Chaîne.
   c. Entrez une nouvelle valeur pour `Song`, `Song A`.
   d. Choisissez Enregistrer les modifications.
6. Supprimez l'élément de table :
   a. Sous Éléments retournés, cochez Album A.
   b. Dans le menu Actions, sélectionnez Supprimer des éléments.

Vous avez effectué trois mises à jour sur l'élément de table ; cela génère trois événements pour le flux de données DynamoDB :

- Un événement `INSERT` correspondant à la création de l'élément.
- Un événement `MODIFY` correspondant à l'ajout d'un attribut à l'élément.
- Un événement `REMOVE` correspondant à la suppression de l'élément.

Cependant, le modèle d'événement que vous avez spécifié pour le canal doit filtrer les événements qui ne sont pas des événements `INSERT` ou `MODIFY`. Vérifiez ensuite que le canal a livré les événements attendus à la file d'attente.

Vérification que les événements attendus ont été livrés à la file d'attente

1. Ouvrez la console Amazon SQS à l'adresse https://console.aws.amazon.com/sqs/.
2. Choisissez la file d'attente `pipe-tutorial-target`.

   Amazon SQS affiche la page détaillée de la file d'attente.
3. Sélectionnez Envoyer et recevoir des messages, puis sous Recevoir des messages, choisissez Rechercher des messages.

   La file d'attente interroge le canal, puis répertorie les événements qu'elle reçoit.
4. Choisissez le nom de l'événement pour voir le code JSON de l'événement qui a été livré.

La file d'attente doit comporter deux événements : un pour lequel `eventName` est défini sur `INSERT` et un autre pour lequel `eventName` est défini sur `MODIFY`. Cependant, le canal n'a pas livré l'événement de suppression de l'élément de table, car le paramètre `eventName` de cet événement était défini sur `REMOVE`, ce qui ne correspondait pas au modèle d'événement que vous avez spécifié dans le canal.

Étape 3 : Nettoyer vos ressources

Commencez par supprimer le canal lui-même.

Suppression du canal à l'aide de la console EventBridge

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Pipelines.

3. Sélectionnez le canal `pipe-tutorial` et choisissez Supprimer.

Supprimez ensuite la pile CloudFormation pour éviter que l'utilisation continue des ressources qu'elle contient ne vous soit facturée.

Suppression des prérequis du didacticiel à l'aide de l'interface de ligne de commande AWS

- Exécutez la commande d'interface de ligne de commande suivante, où `--stack-name` spécifie le nom de votre pile :

```
aws cloudformation delete-stack --stack-name pipe-tuturial-resources
```

Suppression des prérequis du didacticiel à l'aide de la console AWS CloudFormation

1. Ouvrez la console AWS CloudFormation à l'adresse https://console.aws.amazon.com/cloudformation.
2. Sur la page Piles, sélectionnez la pile, puis Supprimer.
3. Sélectionnez Supprimer pour confirmer votre action.

Modèle AWS CloudFormation pour la génération des prérequis

Utilisez le code JSON ci-dessous pour créer un modèle CloudFormation afin d'allouer les ressources de la source et de la cible nécessaires à ce didacticiel.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",

 "Description" : "Provisions resources to use with the EventBridge Pipes tutorial. You
 will be billed for the AWS resources used if you create a stack from this template.",

  "Parameters" : {
    "SourceTableName" : {
      "Type" : "String",
      "Default" : "pipe-tutorial-source",
      "Description" : "Specify the name of the table to provision as the pipe source,
 or accept the default."
    },
  "TargetQueueName" : {
```

```
    "Type" : "String",
    "Default" : "pipe-tutorial-target",
    "Description" : "Specify the name of the queue to provision as the pipe target, or
 accept the default."
  }
},
  "Resources": {
    "PipeTutorialSourceDynamoDBTable": {
      "Type": "AWS::DynamoDB::Table",
      "Properties": {
        "AttributeDefinitions": [{
            "AttributeName": "Album",
            "AttributeType": "S"
          },
          {
            "AttributeName": "Artist",
            "AttributeType": "S"
          }

        ],
        "KeySchema": [{
            "AttributeName": "Album",
            "KeyType": "HASH"

          },
          {
            "AttributeName": "Artist",
            "KeyType": "RANGE"
          }
        ],
        "ProvisionedThroughput": {
          "ReadCapacityUnits": 10,
          "WriteCapacityUnits": 10
        },
        "StreamSpecification": {
          "StreamViewType": "NEW_AND_OLD_IMAGES"
        },
        "TableName": { "Ref" : "SourceTableName" }
      }
    },
    "PipeTutorialTargetQueue": {
      "Type": "AWS::SQS::Queue",
      "Properties": {
        "QueueName": { "Ref" : "TargetQueueName" }
```

```
      }
    }
  }
}
```

Générer un AWS CloudFormation modèle à partir de EventBridge Pipes

AWS CloudFormation vous permet de configurer et de gérer vos AWS ressources sur l'ensemble des comptes et des régions de manière centralisée et reproductible en traitant l'infrastructure comme du code. CloudFormation pour ce faire, vous pouvez créer des modèles qui définissent les ressources que vous souhaitez approvisionner et gérer.

EventBridge vous permet de générer des modèles à partir des canaux existants de votre compte, afin de vous aider à démarrer le développement CloudFormation de modèles. Vous pouvez sélectionner un ou plusieurs canaux à inclure dans le modèle. Vous pouvez ensuite utiliser ces modèles comme base pour créer des piles de ressources à CloudFormation gérer.

Pour plus d'informations CloudFormation, consultez le guide de AWS CloudFormation l'utilisateur.

Pour les bus d'événements, vous pouvez générer des CloudFormation modèles à partir des bus d'événements et des règles des bus d'événements.

Ressources incluses dans les modèles EventBridge Pipe

Lors EventBridge de la CloudFormation génération du modèle, il crée une AWS::Pipes::Piperessource pour chaque canal sélectionné. En outre, EventBridge inclut les ressources suivantes dans les conditions décrites :

- AWS::Events::ApiDestination

  Si vos canaux incluent des destinations d'API, sous forme d'enrichissements ou de cibles, EventBridge incluez-les dans le CloudFormation modèle en tant que AWS::Events::ApiDestination ressources.
- AWS::Events::EventBus

  Si vos canaux incluent un bus d'événements comme cible, EventBridge incluez-le dans le CloudFormation modèle en tant que AWS::Events::EventBus ressource.
- AWS::IAM::Role

Si vous aviez EventBridge créé un nouveau rôle d'exécution lors de la configuration du canal, vous pouvez choisir d' EventBridge inclure ce rôle dans le modèle en tant que AWS::IAM::Role ressource. EventBridge n'inclut pas les rôles que vous créez. (Dans les deux cas, la `RoleArn` propriété de la AWS::Pipes::Pipe ressource contient l'ARN du rôle.)

Considérations relatives à l'utilisation CloudFormation de modèles générés à partir de EventBridge Pipes

Tenez compte des facteurs suivants lorsque vous utilisez un CloudFormation modèle à partir duquel vous l'avez généré EventBridge :

- EventBridge n'inclut aucun mot de passe dans le modèle généré.

  Vous pouvez modifier le modèle pour y inclure des paramètres qui permettent aux utilisateurs de spécifier des mots de passe ou d'autres informations sensibles lorsqu'ils utilisent le modèle pour créer ou mettre à jour une CloudFormation pile.

  En outre, les utilisateurs peuvent utiliser Secrets Manager pour créer un secret dans la région souhaitée, puis modifier le modèle généré pour utiliser des paramètres dynamiques.
- Les cibles du modèle généré restent exactement telles qu'elles ont été spécifiées dans le canal d'origine. Cela peut entraîner des problèmes entre régions si vous ne modifiez pas correctement le modèle avant de l'utiliser pour créer des piles dans d'autres régions.

  De plus, le modèle généré ne crée pas automatiquement les cibles en aval.

Génération d'un CloudFormation modèle à partir de EventBridge Pipes

Pour générer un CloudFormation modèle à partir d'un ou de plusieurs canaux à l'aide de la EventBridge console, procédez comme suit :

Pour générer un CloudFormation modèle à partir d'un ou de plusieurs canaux

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Pipelines.
3. Sous Tuyaux, choisissez un ou plusieurs tuyaux que vous souhaitez inclure dans le CloudFormation modèle généré.

Pour un canal unique, vous pouvez également choisir le nom du canal pour afficher la page de détails du canal.

4. Choisissez CloudFormation Modèle, puis choisissez le format dans lequel vous EventBridge souhaitez générer le modèle : JSON ou YAML.

   EventBridge affiche le modèle, généré dans le format sélectionné.

5. Si vous avez EventBridge créé un nouveau rôle d'exécution pour l'un des canaux sélectionnés et que vous EventBridge souhaitez inclure ces rôles dans le modèle, choisissez Inclure IAM les rôles créés par la console en votre nom.
6. EventBridge vous permet de télécharger le fichier modèle ou de le copier dans le presse-papiers.

   - Pour télécharger le fichier modèle, choisissez Télécharger.
   - Pour copier le modèle dans le presse-papiers, choisissez Copier.

7. Pour quitter le modèle, choisissez Annuler.

Rendre les applications tolérantes aux pannes régionales avec des points de terminaison globaux et une réplication d'événements

Vous pouvez améliorer la disponibilité de votre application grâce aux points de terminaison EventBridge mondiaux Amazon. Les points de terminaison globaux contribuent à rendre votre application tolérante aux pannes régionales sans frais supplémentaires. Pour commencer, affectez une surveillance d'état Amazon Route 53 au point de terminaison. Lorsque le basculement est lancé, la surveillance d'état indique un état « Unhealthy » (défectueux). Quelques minutes après le lancement du basculement, tous les événements personnalisés sont routés vers un bus d'événements dans la région secondaire et sont traités par ce bus d'événements. Une fois que la surveillance d'état indique un état « Healthy » (sain), les événements sont traités par le bus d'événements dans la région principale.

Lorsque vous utilisez des points de terminaison globaux, vous pouvez activer la réplication d'événements. La réplication d'événements envoie tous les événements personnalisés aux bus d'événements des régions principale et secondaire à l'aide de règles gérées.

> **Note**
>
> Si vous utilisez des bus personnalisés, vous aurez besoin d'un bus personnalisé dans chaque région portant le même nom et dans le même compte pour que le basculement fonctionne correctement.

Rubriques

Objectifs de délai de reprise et de point de reprise

L'objectif de délai de reprise (RTO) correspond au temps nécessaire à la région secondaire pour commencer à recevoir des événements après une panne. Pour le RTO, le délai inclut la période de déclenchement des CloudWatch alarmes et de mise à jour des statuts pour les bilans de santé de Route 53. L'objectif de point de reprise (RPO) correspond à la mesure des données qui ne seront pas traitées en cas de panne. Pour le RPO, cela inclut les événements qui ne sont pas répliqués dans la région secondaire et qui sont bloqués dans la région principale jusqu'à ce que le service ou la région soit rétabli(e). Avec les points de terminaison globaux, si vous suivez nos recommandations en matière de configuration des alarmes, vous pouvez vous attendre à ce que le RTO et le RPO soient de 360 secondes avec un maximum de 420 secondes.

Réplication des événements

Les événements sont traités de manière asynchrone dans la région secondaire. Cela signifie qu'il n'est pas garanti que les événements seront traités en même temps dans les deux régions. Lorsque le basculement est déclenché, les événements sont traités par la région secondaire et seront traités par la région principale lorsqu'elle sera disponible. L'activation de la réplication des événements augmentera vos coûts mensuels. Pour plus d'informations, consultez les EventBridgetarifs Amazon

Nous recommandons d'activer la réplication des événements lors de la configuration des points de terminaison globaux pour les raisons suivantes :

- La réplication d'événements vous permet de vérifier que vos points de terminaison globaux sont correctement configurés. Cela permet de garantir que vous serez couvert en cas de basculement.
- La réplication des événements est requise pour effectuer une récupération automatique suite à un événement de basculement. Si la réplication des événements n'est pas activée, vous devrez réinitialiser manuellement la surveillance d'état Route 53 sur « Healthy » (sain) avant que les événements ne retournent dans la région principale.

Charge utile des événements répliqués

Voici un exemple de charge utile d'événements répliqués :

> **Note**
>
> L'élément `region` indique la région à partir de laquelle l'événement a été répliqué.

```
{
    "version": "0",
    "id": "a908baa3-65e5-ab77-367e-527c0e71bbc2",
    "detail-type": "Test",
    "source": "test.service.com",
    "account": "0123456789",
    "time": "1900-01-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:events:us-east-1:0123456789:endpoint/MyEndpoint"
    ],
    "detail": {
        "a": "b"
    }
}
```

Création d’un point de terminaison global

Pour configurer un point de terminaison global, procédez comme suit :

1. Assurez-vous que les bus d’événements et les règles sont identiques dans la région principale et la région secondaire.
2. Créez une surveillance d’état Route 53 pour surveiller vos bus d’événements. Pour obtenir de l’aide lors de la création de votre surveillance d’état, choisissez Nouvelle surveillance de l’état lors de la création de votre point de terminaison global.
3. Créez votre point de terminaison global.

Une fois que vous avez configuré la surveillance d’état Route 53, vous pouvez créer un point de terminaison global.

Pour créer un point de terminaison global à l’aide de la console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Points de terminaison globaux.
3. Choisissez Créer un point de terminaison.
4. Entrez un nom et une description pour le point de terminaison.

5. Pour Bus d’événements dans la région principale, choisissez le bus d’événements auquel vous souhaitez associer le point de terminaison.
6. Pour Région secondaire, choisissez la région vers laquelle vous souhaitez diriger les événements en cas de basculement.

> **Note**
>
> Le champ Bus d’événements dans la région secondaire est rempli automatiquement et n’est pas modifiable.

7. Pour Surveillance de l’état Route 53 destinée à déclencher le basculement et la récupération, choisissez la surveillance d’état que le point de terminaison va surveiller. Si vous n'avez pas encore effectué de bilan de santé, choisissez New Health check pour ouvrir la AWS CloudFormation console et créer un bilan de santé à l'aide d'un CloudFormation modèle.

> **Note**
>
> Les données manquantes entraîneront l’échec de la surveillance d’état. Si vous n'avez besoin d'envoyer des événements que par intermittence, envisagez d'utiliser un envoi plus long MinimumEvaluationPeriodou considérez les données manquantes comme « manquantes » plutôt que comme des « violations ».

8. (Facultatif) Pour la réplication d’événements, procédez comme suit :
   a. Sélectionnez Réplication des événements activée.
   b. Pour Rôle d’exécution, choisissez de créer un nouveau rôle AWS Identity and Access Management ou d’utiliser un rôle existant. Procédez comme suit :
      - Choisissez Create a new role for this specific resource. Vous pouvez éventuellement mettre à jour le Nom du rôle pour créer un nouveau rôle.
      - Choisissez Utiliser le rôle existant. Ensuite, pour Rôle d’exécution, choisissez le rôle à utiliser.
9. Choisissez Créer.

Pour créer un point de terminaison global à l’aide de l’API

Pour créer un point de terminaison global à l'aide de l' EventBridge API, consultez CreateEndpointle Amazon EventBridge API Reference.

Pour créer un point de terminaison global à l’aide d’ AWS CloudFormation

Pour créer un point de terminaison global à l'aide de l' AWS CloudFormation API, consultez AWS::Events::Endpointsle guide de AWS CloudFormation l'utilisateur.

Utilisation de points de terminaison globaux à l'aide d'un SDK AWS

> **Note**
>
> La prise en charge de C++ sera bientôt disponible.

Lorsque vous utilisez un AWS SDK pour travailler avec des points de terminaison globaux, gardez les points suivants à l'esprit :

- La bibliothèque AWS Common Runtime (CRT) doit être installée pour votre SDK spécifique. Si elle n’est pas installée, vous recevrez un message d’exception indiquant ce qui doit être installé. Pour plus d’informations, consultez les ressources suivantes :
  - Bibliothèques CRT (Common Runtime)AWS
  - guênes/ aws-crt-java
  - guênes/ aws-crt-nodejs
  - guênes/ aws-crt-python
- Une fois que vous avez créé un point de terminaison global, vous devez ajouter `endpointId` et `EventBusName` à tous les appels `PutEvents` que vous utilisez.
- Les points de terminaison globaux prennent en charge Signature Version 4A. Cette version de SigV4 permet de signer des requêtes pour plusieurs Régions AWS. Ceci est utile dans les opérations d'API qui peuvent entraîner un accès aux données à partir d'une de plusieurs régions. Lorsque vous utilisez le AWS SDK, vous fournissez vos informations d'identification et les demandes adressées aux points de terminaison globaux utiliseront la version 4A de Signature sans configuration supplémentaire. Pour en savoir plus sur SigV4A, consultez Signature AWS de demandes d'API dans les AWS Références générales .

Si vous demandez des informations d'identification temporaires au point de AWS STS terminaison global (sts.amazonaws.com), des informations d'identification qui, par défaut, ne sont pas compatibles avec le protocole AWS STS SigV4A. Voir Gestion AWS STS dans une AWS région dans le guide de AWS Identity and Access Management l'utilisateur pour plus d'informations.

Régions disponibles

Les régions suivantes prennent en charge les points de terminaison globaux.

- USA Est (Virginie du Nord)
- USA Est (Ohio)
- USA Ouest (Californie du Nord)
- US West (Oregon)
- Canada (Centre)
- Europe (Francfort)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Osaka)
- Asia Pacific (Seoul)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asia Pacific (Tokyo)
- Amérique du Sud (São Paulo)

Bonnes pratiques pour l'utilisation des points de terminaison globaux Amazon EventBridge

Les bonnes pratiques suivantes sont recommandées lorsque vous configurez des points de terminaison globaux.

Rubriques

Activation de la réplication des événements

Nous vous recommandons vivement d'activer la réplication et de traiter vos événements dans la région secondaire que vous affectez à votre point de terminaison global. Cela garantit que votre application dans la région secondaire est correctement configurée. Vous devriez également activer la réplication pour garantir la récupération automatique dans la région principale une fois le problème résolu.

Les ID d'événement peuvent changer en fonction des appels d'API. Pour corréler les événements entre les régions, vous devez donc disposer d'un identifiant unique et immuable. Les consommateurs devraient également être conçus en tenant compte de l'idempotence. Ainsi, si vous répliquez des événements ou si vous les relisez à partir d'archives, le traitement des événements dans les deux régions n'aura aucun effet secondaire.

Prévention de la limitation des événements

Pour éviter que les événements ne soient limités, nous vous recommandons de mettre à jour vos limites relatives aux actions `PutEvents` et aux cibles afin qu'elles soient cohérentes d'une région à l'autre.

Utilisation des métriques d'abonné dans les surveillances d'état Amazon Route 53

Évitez d'inclure des métriques d'abonné dans vos surveillances d'état Amazon Route 53. L'inclusion de ces métriques peut entraîner le basculement de votre diffuseur de publication vers les régions secondaires si un abonné rencontre un problème alors que tous les autres abonnés restent sains

dans la région principale. Si l'un de vos abonnés ne parvient pas à traiter les événements dans la région principale, vous devriez activer la réplication pour que votre abonné de la région secondaire puisse traiter les événements avec succès.

Modèle AWS CloudFormation pour configurer la surveillance d'état Route 53

Lorsque vous utilisez des points de terminaison globaux, vous devez effectuer une surveillance d'état Route 53 pour surveiller l'état de vos régions. Le modèle suivant définit une alarme Amazon CloudWatch et l'utilise pour définir une surveillance d'état Route 53.

Rubriques

- Modèle AWS CloudFormation pour définir une surveillance d'état Route 53
- Propriétés du modèle d'alarme CloudWatch
- Propriétés du modèle de surveillance d'état Route 53

Modèle AWS CloudFormation pour définir une surveillance d'état Route 53

Utilisez le modèle suivant pour définir votre surveillance d'état Route 53.

```
Description: |-
  Global endpoints health check that will fail when the average Amazon EventBridge
  latency is above 30 seconds for a duration of 5 minutes. Note, missing data will
  cause the health check to fail, so if you only send events intermittently, consider
  changing the heath check to use a longer evaluation period or instead treat missing
  data as 'missing' instead of 'breaching'.

Metadata:
  AWS::CloudFormation::Interface:
    ParameterGroups:
      - Label:
          default: "Global endpoint health check alarm configuration"
        Parameters:
          - HealthCheckName
          - HighLatencyAlarmPeriod
          - MinimumEvaluationPeriod
          - MinimumThreshold
          - TreatMissingDataAs
    ParameterLabels:
```

```
      HealthCheckName:
        default: Health check name
      HighLatencyAlarmPeriod:
        default: High latency alarm period
      MinimumEvaluationPeriod:
        default: Minimum evaluation period
      MinimumThreshold:
        default: Minimum threshold
      TreatMissingDataAs:
        default: Treat missing data as

Parameters:
  HealthCheckName:
    Description: Name of the health check
    Type: String
    Default: LatencyFailuresHealthCheck
  HighLatencyAlarmPeriod:
    Description: The period, in seconds, over which the statistic is applied. Valid
 values are 10, 30, 60, and any multiple of 60.
    MinValue: 10
    Type: Number
    Default: 60
  MinimumEvaluationPeriod:
    Description: The number of periods over which data is compared to the specified
 threshold. You must have at least one evaluation period.
    MinValue: 1
    Type: Number
    Default: 5
  MinimumThreshold:
    Description: The value to compare with the specified statistic.
    Type: Number
    Default: 30000
  TreatMissingDataAs:
    Description: Sets how this alarm is to handle missing data points.
    Type: String
    AllowedValues:
      - breaching
      - notBreaching
      - ignore
      - missing
    Default: breaching

Mappings:
  "InsufficientDataMap":
```

```
    "missing":
      "HCConfig": "LastKnownStatus"
    "breaching":
      "HCConfig": "Unhealthy"

Resources:
  HighLatencyAlarm:
      Type: AWS::CloudWatch::Alarm
      Properties:
        AlarmDescription: High Latency in Amazon EventBridge
        MetricName: IngestionToInvocationStartLatency
        Namespace: AWS/Events
        Statistic: Average
        Period: !Ref HighLatencyAlarmPeriod
        EvaluationPeriods: !Ref MinimumEvaluationPeriod
        Threshold: !Ref MinimumThreshold
        ComparisonOperator: GreaterThanThreshold
        TreatMissingData: !Ref TreatMissingDataAs

  LatencyHealthCheck:
      Type: AWS::Route53::HealthCheck
      Properties:
        HealthCheckTags:
          - Key: Name
            Value: !Ref HealthCheckName
        HealthCheckConfig:
          Type: CLOUDWATCH_METRIC
          AlarmIdentifier:
            Name:
              Ref: HighLatencyAlarm
            Region: !Ref AWS::Region
          InsufficientDataHealthStatus: !FindInMap [InsufficientDataMap, !Ref
 TreatMissingDataAs, HCConfig]

Outputs:
  HealthCheckId:
    Description: The identifier that Amazon Route 53 assigned to the health check when
 you created it.
    Value: !GetAtt LatencyHealthCheck.HealthCheckId
```

Les ID d'événement peuvent changer en fonction des appels d'API. Pour corréler les événements entre les régions, vous devez donc disposer d'un identifiant unique et immuable. Les consommateurs devraient également être conçus en tenant compte de l'idempotence. Ainsi, si vous répliquez des

événements ou si vous les relisez à partir d'archives, le traitement des événements dans les deux régions n'aura aucun effet secondaire.

Propriétés du modèle d'alarme CloudWatch

> **Note**
>
> Pour tous les champs **`editable`**, tenez compte de votre débit par seconde. Si vous n'envoyez des événements que par intermittence, envisagez de modifier la surveillance d'état de sorte à utiliser une période d'évaluation plus longue ou à traiter les données manquantes en tant que `missing` plutôt que `breaching`.

Les propriétés suivantes sont utilisées dans la section d'alarme CloudWatch du modèle :

Métrique	Description
`AlarmDescription`	Description de l'alarme. Par défaut : **`High Latency in Amazon EventBridge`**
`MetricName`	Nom de la métrique associée à l'alarme. Il est obligatoire pour une alarme basée sur une métrique. Pour une alarme basée sur une expression mathématique, vous utilisez `Metrics` à la place et vous ne pouvez pas spécifier `MetricName` . Valeur par défaut : IngestionToInvocationStartLatency
`Namespace`	Espace de noms de la métrique associée à l'alarme. Il est obligatoire pour une alarme basée sur une métrique. Pour une alarme basée sur une expression mathématique, vous ne pouvez pas spécifier `Namespace` et vous utilisez `Metrics` à la place. Par défaut : `AWS/Events`
`Statistic`	Statistique pour la métrique associée à l'alarme, autre que sur les centiles. Valeur par défaut : Average

Métrique	Description
`Period`	La période, en secondes, au cours de laquelle la statistique est appliquée. Il est obligatoire pour une alarme basée sur une métrique. Les valeurs valides sont 10, 30, 60 et n'importe quel multiple de 60. Par défaut : **`60`**
`EvaluationPeriods`	Nombre de périodes au cours desquelles les données sont comparées au seuil défini. Si vous définissez une alarme qui nécessite qu'un certain nombre de points de données consécutifs soient atteints pour déclencher l'alarme, cette valeur spécifie ce nombre. Si vous définissez une alarme « M sur N », cette valeur correspond au N et `DatapointsToAlarm` correspond au M. Par défaut : **`5`**
`Threshold`	Valeur à comparer à la statistique spécifiée. Par défaut : **`30,000`**
`ComparisonOperator`	Opération arithmétique à utiliser lors de la comparaison de la statistique et du seuil spécifiés. La valeur de statistique spécifiée est utilisée comme premier opérateur. Par défaut : `GreaterThanThreshold`
`TreatMissingData`	Définit la façon dont cette alarme doit gérer les points de données manquants. Valeurs valides : `breaching` , `notBreaching` , `ignore` et `missing` Par défaut : `breaching`

Propriétés du modèle de surveillance d'état Route 53

> **Note**
>
> Pour tous les champs **`editable`**, tenez compte de votre débit par seconde. Si vous n'envoyez des événements que par intermittence, envisagez de modifier la surveillance d'état de sorte à utiliser une période d'évaluation plus longue ou à traiter les données manquantes en tant que `missing` plutôt que `breaching`.

Les propriétés suivantes sont utilisées dans la section de surveillance d'état Route 53 du modèle :

Métrique	Description
`HealthCheckName`	Nom de la surveillance d'état. Par défaut : **`LatencyFailuresHealthCheck`**
`InsufficientDataHealthStatus`	Quand CloudWatch ne dispose pas de suffisamment de données sur la métrique pour déterminer l'état de l'alarme, il s'agit du statut qu'Amazon Route 53 doit affecter à la surveillance d'état. Valeurs valides : • `Healthy` : Route 53 considère la vérification de l'état comme étant saine. • `Unhealthy` : Route 53 considère la vérification de l'état comme étant non saine. • `LastKnownStatus` : Route 53 utilise le statut de la vérification de l'état constaté la dernière fois où CloudWatch avait suffisamment de données pour déterminer l'état de l'alarme. Pour les nouvelles surveillances de l'état n'ayant aucun dernier statut connu, le statut par défaut de la surveillance de l'état est « sain ». Valeur par défaut : Unhealthy

Métrique	Description
	Note Ce champ est mis à jour en fonction des données entrées dans le champ `TreatMissingData` . Si `TreatingMissingData` est défini sur `Missing`, il sera mis à jour sur `LastKnownStatus` . Si `TreatingMissingData` est défini sur `Breaching` , il sera mis à jour sur `Unhealthy` .

EventBridge Schémas Amazon

Un schéma définit la structure des événements envoyés à EventBridge. EventBridge fournit des schémas pour tous les événements générés par les AWS services. Vous pouvez également créer ou charger des schémas personnalisés ou inférer des schémas directement à partir d'événements sur un bus d'événements. Une fois que vous disposez d'un schéma pour un événement, vous pouvez télécharger des liaisons de code pour les langages de programmation usuels et accélérer le développement. Vous pouvez utiliser des liaisons de code pour les schémas et gérer les schémas depuis la EventBridge console, à l'aide de l'API, ou directement dans votre IDE à l'aide des boîtes à outils. AWS Pour créer des applications sans serveur qui utilisent des événements, utilisez AWS Serverless Application Model.

> **Note**
>
> Lorsque vous utilisez la fonctionnalité de transformation d'entrée, l'événement d'origine est inféré par la découverte du schéma, et non l'événement transformé envoyé à la cible.

EventBridge supporte les formats OpenAPI 3 et JSONSchema Draft4.

Pour AWS Toolkit for JetBrains et AWS Toolkit for VS Code, vous pouvez parcourir ou rechercher des schémas et télécharger des liaisons de code pour les schémas directement dans votre IDE.

La vidéo suivante donne un aperçu des schémas et des registres de schémas : Utilisation du registre des schémas

Rubriques

- Masquage des valeurs de propriété pour l'API du registre des schémas
- Trouver un EventBridge schéma Amazon
- Registres EventBridge de schémas Amazon
- Création d'un EventBridge schéma Amazon
- liaisons EventBridge de code Amazon

Masquage des valeurs de propriété pour l’API du registre des schémas

Certaines valeurs de propriété des événements utilisés pour créer un registre de schémas peuvent contenir des informations sensibles sur le client. Pour protéger les informations du client, les valeurs seront masquées par des astérisques (*). Comme nous masquons ces valeurs, il est EventBridge recommandé de ne pas créer d'applications qui dépendent explicitement des propriétés suivantes ou de leurs valeurs :

- CreateSchema— Les `Content` propriétés du `requestParameters` corps
- GetDiscoveredSchema— La `Events` propriété du `requestParameters` corps et la `Content` propriété du `responseElements` corps
- SearchSchemas— La `keywords` propriété du `requestParameters`
- UpdateSchema— La `Content` propriété du `requestParameters`

Trouver un EventBridge schéma Amazon

EventBridge inclut des schémas pour tous les AWS services qui génèrent des événements. Vous pouvez trouver ces schémas dans la EventBridge console ou vous pouvez les trouver en utilisant l'action `SearchSchemas`API.

Pour rechercher des schémas de AWS services dans la console EventBridge

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schemas (Schémas).
3. Sur la page Schémas, sélectionnez Registre des schémas d’événements AWS .
   <result>

   La première page des schémas disponibles s'affiche.
   </result>
4. Pour rechercher un schéma, dans Rechercher des schémas AWS d'événements, entrez un terme de recherche.

   Une recherche renvoie les correspondances pour le nom et le contenu des schémas disponibles, puis affiche les versions du schéma qui contiennent les correspondances.
5. Ouvrez un schéma d'événements en sélectionnant le nom du schéma.

Registres EventBridge de schémas Amazon

Les registres de schémas sont des conteneurs pour les schémas. Les registres de schémas collectent et organisent les schémas de telle sorte que vos schémas soient dans des groupes logiques. Les registres de schémas par défaut sont les suivants :

- Tous les schémas : tous les schémas issus des registres des AWS événements, des schémas découverts et des registres de schémas personnalisés.
- AWS registre des schémas d'événements — Les schémas intégrés.
- Registre de schéma découvert : schémas découverts par la découverte de schéma.

Vous pouvez créer des registres personnalisés pour organiser les schémas que vous créez ou téléchargez.

Pour créer un registre personnalisé

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schémas, puis choisissez Créer un registre.
3. Sur la page Informations de registre, entrez un Nom.
4. (Facultatif) Entrez une description pour votre nouveau registre.
5. Choisissez Créer.

Pour créer un schéma personnalisé dans votre nouveau registre, sélectionnez Créer un schéma personnalisé. Pour ajouter un schéma à votre registre, sélectionnez ce dernier lorsque vous créez un nouveau schéma.

Pour créer un registre à l'aide de l'API, utilisez `CreateRegistry`. Pour plus d'informations, consultez le manuel Amazon EventBridge Schema Registry API Reference.

Pour plus d'informations sur l'utilisation du registre des EventBridge schémas via AWS CloudFormation, consultez la section EventSchemas Resource Type Reference in AWS CloudFormation.

Création d'un EventBridge schéma Amazon

Vous créez des schémas en utilisant des fichiers JSON avec la spécification OpenAPI ou la spécification JSONSchema Draft4. Vous pouvez créer ou télécharger vos propres schémas en EventBridge utilisant un modèle ou en générant un schéma basé sur le JSON d'un événement. Vous pouvez également inférer le schéma à partir d'événements sur un bus d'événements. Pour créer un schéma à l'aide de l'API EventBridge Schema Registry, utilisez l'action `CreateSchema`API.

Lorsque vous choisissez entre les formats OpenAPI 3 et JSONSchema Draft4, tenez compte des différences suivantes :

- Le format JSONSchema prend en charge les mots-clés supplémentaires qui ne sont pas pris en charge dans OpenAPI, tels que `$schema, additionalItems`.
- Il existe des différences mineures dans la façon dont les mots-clés sont traités, telles que le `type` et le `format`.
- OpenAPI ne prend pas en charge les liens hypertexte JSONSchema Hyper-Schema dans les documents JSON.
- Les outils pour OpenAPI ont tendance à se concentrer sur le moment de la construction, tandis que les outils pour JSONSchema ont tendance à se concentrer sur les opérations d'exécution, tels que les outils clients pour la validation des schémas.

Nous recommandons d'utiliser le format JSONSchema pour implémenter la validation côté client afin que les événements envoyés EventBridge soient conformes au schéma. Vous pouvez utiliser JSONSchema pour définir un contrat pour les documents JSON valides, puis utiliser un outil de validation de schéma JSON avant d'envoyer les événements associés.

Une fois que vous disposez d'un nouveau schéma, vous pouvez télécharger des liaisons de code afin de créer des applications pour les événements utilisant ce schéma.

Rubriques

- Création d'un schéma à l'aide d'un modèle
- Modification d'un modèle de schéma directement dans la console
- Création d'un schéma à partir du code JSON d'un événement
- Création d'un schéma à partir d'événements sur un bus d'événements

Création d’un schéma à l’aide d’un modèle

Vous pouvez créer un schéma à partir d'un modèle ou en modifiant un modèle directement dans la EventBridge console. Pour obtenir le modèle, vous le téléchargez à partir de la console. Vous pouvez modifier le modèle afin que le schéma corresponde à vos événements. Chargez ensuite votre nouveau modèle via la console.

Pour télécharger le modèle de schéma

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schema registry (Registre de schémas).
3. Dans la section Getting Started (Mise en route), sous Schema template (Modèle de schéma), choisissez Download (Télécharger).

Vous pouvez également copier le modèle JSON à partir de l’exemple de code suivant.

```
{
    "openapi": "3.0.0",
    "info": {
      "version": "1.0.0",
      "title": "Event"
    },
    "paths": {},
    "components": {
      "schemas": {
        "Event": {
          "type": "object",
          "properties": {
            "ordinal": {
              "type": "number",
              "format": "int64"
            },
            "name": {
              "type": "string"
            },
            "price": {
              "type": "number",
              "format": "double"
            },
            "address": {
              "type": "string"
```

```
            },
            "comments": {
              "type": "array",
              "items": {
                "type": "string"
              }
            },
            "created_at": {
              "type": "string",
              "format": "date-time"
            }
          }
        }
      }
    }
  }
```

Pour charger un modèle de schéma

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schémas, puis choisissez Créer un schéma.
3. (Facultatif) Sélectionnez ou créez un registre de schémas.
4. Sous Informations de schéma, entrez un nom pour votre schéma.
5. (Facultatif) Entrez une description pour votre schéma.
6. Pour Type de schéma, choisissez OpenAPI 3.0 ou JSON Schema Draft 4.
7. Dans la zone de texte de l'onglet Créer, faites glisser votre fichier de schéma vers la zone de texte ou collez la source du schéma.
8. Sélectionnez Créer.

Modification d'un modèle de schéma directement dans la console

Pour modifier un schéma dans la console

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schémas, puis choisissez Créer un schéma.
3. (Facultatif) Sélectionnez ou créez un registre de schémas.
4. Sous Informations de schéma, entrez un nom pour votre schéma.

5. Pour Type de schéma, choisissez OpenAPI 3.0 ou JSON Schema Draft 4.
6. (Facultatif) Entrez une description du schéma à créer.
7. Dans l'onglet Créer, choisissez Charger un modèle.
8. Dans la zone de texte, modifiez le modèle afin que le schéma corresponde à vos événements.
9. Sélectionnez Créer.

Création d'un schéma à partir du code JSON d'un événement

Si vous disposez du code JSON d'un événement, vous pouvez créer automatiquement un schéma pour ce type d'événement.

Pour créer un schéma en fonction du code JSON d'un événement

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schémas, puis choisissez Créer un schéma.
3. (Facultatif) Sélectionnez ou créez un registre de schémas.
4. Sous Schema details (Détails du schéma), entrez un nom pour votre schéma.
5. (Facultatif) Entrez une description du schéma que vous avez créé.
6. Pour Type de schéma, choisissez OpenAPI 3.0.

   Vous ne pouvez pas utiliser JSONSchema lorsque vous créez un schéma à partir du code JSON d'un événement.
7. Sélectionnez Discover from JSON (Découvrir à partir de JSON)
8. Dans la zone de texte sous JSON, collez ou faites glisser la source JSON d'un événement.

   Par exemple, vous pouvez coller la source de cet AWS Step Functions événement en cas d'échec d'exécution.

```
{
    "version": "0",
    "id": "315c1398-40ff-a850-213b-158f73e60175",
    "detail-type": "Step Functions Execution Status Change",
    "source": "aws.states",
    "account": "012345678912",
    "time": "2019-02-26T19:42:21Z",
    "region": "us-east-1",
    "resources": [
```

```
      "arn:aws:states:us-east-1:012345678912:execution:state-machine-
name:execution-name"
    ],
    "detail": {
        "executionArn": "arn:aws:states:us-east-1:012345678912:execution:state-
machine-name:execution-name",
        "stateMachineArn": "arn:aws:states:us-
east-1:012345678912:stateMachine:state-machine",
        "name": "execution-name",
        "status": "FAILED",
        "startDate": 1551225146847,
        "stopDate": 1551225151881,
        "input": "{}",
        "output": null
    }
}
```

9. Choisissez Discover schema (Découvrir le schéma).
10. EventBridge génère un schéma OpenAPI pour l'événement. Par exemple, le schéma suivant est généré pour l’événement Step Functions précédent.

```
{
  "openapi": "3.0.0",
  "info": {
    "version": "1.0.0",
    "title": "StepFunctionsExecutionStatusChange"
  },
  "paths": {},
  "components": {
    "schemas": {
      "AWSEvent": {
        "type": "object",
        "required": ["detail-type", "resources", "detail", "id", "source", "time",
 "region", "version", "account"],
        "x-amazon-events-detail-type": "Step Functions Execution Status Change",
        "x-amazon-events-source": "aws.states",
        "properties": {
          "detail": {
            "$ref": "#/components/schemas/StepFunctionsExecutionStatusChange"
          },
          "account": {
            "type": "string"
          },
```

```
          "detail-type": {
            "type": "string"
          },
          "id": {
            "type": "string"
          },
          "region": {
            "type": "string"
          },
          "resources": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "source": {
            "type": "string"
          },
          "time": {
            "type": "string",
            "format": "date-time"
          },
          "version": {
            "type": "string"
          }
        }
      },
      "StepFunctionsExecutionStatusChange": {
        "type": "object",
        "required": ["output", "input", "executionArn", "name", "stateMachineArn",
"startDate", "stopDate", "status"],
        "properties": {
          "executionArn": {
            "type": "string"
          },
          "input": {
            "type": "string"
          },
          "name": {
            "type": "string"
          },
          "output": {},
          "startDate": {
            "type": "integer",
```

```
            "format": "int64"
          },
          "stateMachineArn": {
            "type": "string"
          },
          "status": {
            "type": "string"
          },
          "stopDate": {
            "type": "integer",
            "format": "int64"
          }
        }
      }
    }
  }
}
```

11. Une fois le schéma généré, choisissez Créer.

Création d’un schéma à partir d’événements sur un bus d’événements

EventBridge peut déduire des schémas en découvrant des événements. Pour inférer des schémas, vous activez la découverte d’événement sur un bus d’événements et chaque schéma unique est ajouté au registre des schémas, y compris ceux relatifs aux événements entre comptes. Les schémas découverts par EventBridge apparaissent dans le registre des schémas découverts sur la page Schémas.

Si le contenu des événements sur le bus d'événements change, EventBridge crée de nouvelles versions du EventBridge schéma associé.

> **Note**
>
> L’activation de la découverte d’événement sur un bus d’événements peut entraîner des frais. Les cinq premiers millions d’événements traités chaque mois sont gratuits.

> **Note**
>
> EventBridge déduit des schémas à partir d'événements entre comptes par défaut, mais vous pouvez le désactiver en mettant à jour la propriété. `cross-account` Pour plus

d'informations, consultez Discoverers dans le document de référence de l'API EventBridge Schema Registry.

Pour activer la découverte de schéma sur un bus d'événements

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Effectuez l’une des actions suivantes :
   - Pour activer la découverte sur le bus d’événements par défaut, choisissez Démarrer la découverte.
   - Pour activer la découverte sur un bus d’événements personnalisé, sélectionnez le bouton radio correspondant au bus d’événements personnalisé, puis choisissez Démarrer la découverte.

liaisons EventBridge de code Amazon

Vous pouvez générer des liaisons de code pour les schémas d'événements afin d'accélérer le développement dans Golang, Java, Python et. TypeScript Les liaisons de code sont disponibles pour les événements de service AWS , les schémas que vous créez et les schémas que vous générez en fonction d'événements sur un bus d'événements. Vous pouvez générer des liaisons de code pour un schéma à l'aide de la EventBridge console, de l'API EventBridge Schema Registry ou d'un AWS kit d'outils dans votre IDE.

Pour générer des liaisons de code à partir d'un schéma EventBridge

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schemas (Schémas).
3. Recherchez un schéma pour lequel vous souhaitez des liaisons de code, soit en parcourant les registres de schémas, soit en recherchant un schéma.
4. Sélectionnez le nom du schéma.
5. Sur la page de Informations de schéma, dans la section Version, choisissez Télécharger les liaisons de code.
6. Sur la page Download code bindings (Télécharger les liaisons de code), sélectionnez le langage des liaisons de code que vous souhaitez télécharger.
7. Sélectionnez Download (Télécharger).

   Quelques secondes peuvent être nécessaires avant que votre téléchargement commence. Le fichier téléchargé est un fichier zip contenant des liaisons de code pour le langage que vous avez sélectionné.

Services et outils associés à Amazon EventBridge

Amazon EventBridge fait appel à d'autres services et outils AWS pour traiter les événements ou invoquer une ressource en tant que cible d'une règle. Pour plus d'informations sur les intégrations d'EventBridge à d'autres services et outils AWS, consultez les sections suivantes :

Rubriques

- Utilisation d'Amazon EventBridge avec des points de terminaison de VPC d'interface
- Intégration d'Amazon EventBridge à AWS X-Ray
- Utilisation EventBridge avec le kit de test d'application AWS intégré
- Inclure les EventBridge ressources Amazon dans des AWS CloudFormation piles

Utilisation d'Amazon EventBridge avec des points de terminaison de VPC d'interface

Si vous utilisez Amazon Virtual Private Cloud (Amazon VPC) pour héberger vos ressources AWS, vous pouvez établir une connexion privée entre votre VPC et EventBridge. Les ressources de votre VPC peuvent utiliser cette connexion pour communiquer avec EventBridge.

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 EventBridge, vous définissez un point de terminaison de VPC d'interface pour EventBridge. Le point de terminaison assure une connectivité scalable et fiable à EventBridge, sans qu'une passerelle Internet, une instance NAT (Network Address Translation) ou une connexion VPN ne soit nécessaire. Pour de plus amples informations, consultez 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, 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 la section PrivateLink AWS et points de terminaison d'un VPC.

Lorsque vous utilisez un point de terminaison de VPC d'interface privée, les événements personnalisés que votre VPC envoie à EventBridge utilisent ce point de terminaison. EventBridge envoie ensuite ces événements à d'autres services AWS en fonction des règles et des cibles que vous avez configurées. Une fois les événements envoyés à un autre service, vous pouvez les recevoir via le point de terminaison public ou un point de terminaison de VPC pour ce service. Par exemple, si vous créez une règle pour envoyer des événements à une file d'attente Amazon SQS, vous pouvez configurer un point de terminaison de VPC d'interface pour qu'Amazon SQS reçoive des messages de cette file d'attente dans votre VPC sans utiliser le point de terminaison public.

Disponibilité

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

- USA Est (Ohio)
- USA Est (Virginie du Nord)
- USA Ouest (Californie du Nord)

- USA Ouest (Oregon)
- Afrique (Le Cap)
- Asie-Pacifique (Mumbai)
- Asie-Pacifique (Hyderabad)
- Asie-Pacifique (Hong Kong)
- Asie-Pacifique (Singapour)
- Asie-Pacifique (Sydney)
- Asie-Pacifique (Jakarta)
- Asie-Pacifique (Melbourne)
- Asie-Pacifique (Tokyo)
- Asie-Pacifique (Séoul)
- Asie-Pacifique (Osaka)
- Canada (Centre)
- Canada Ouest (Calgary)
- Chine (Beijing)
- China (Ningxia)
- Europe (Francfort)
- Europe (Zurich)
- Europe (Irlande)
- Europe (Londres)
- Europe (Milan)
- Europe (Espagne)
- Europe (Paris)
- Europe (Stockholm)
- Moyen-Orient (EAU)
- Moyen-Orient (Bahreïn)
- Amérique du Sud (Sao Paulo)
- Israël (Tel Aviv)
- AWS GovCloud (US-West)
- AWS GovCloud (US, côte est)

Création d'un point de terminaison de VPC pour EventBridge

Pour utiliser EventBridge avec votre VPC, créez un point de terminaison de VPC d'interface pour EventBridge et choisissez com.amazonaws.***région***.events comme nom de service. Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Amazon VPC Guide de l'utilisateur.

Particularités pour EventBridge Pipes

La prise en charge complète d'EventBridge Pipes pour les points de terminaison de VPC d'interface n'est pas disponible. Pour utiliser les sources suivantes dans un VPC avec EventBridge Pipes, consultez les rubriques suivantes :

- Configuration réseau pour Amazon MSK
- Configuration réseau pour Apache Kafka autogéré
- Configuration réseau pour Amazon MQ

Intégration d'Amazon EventBridge à AWS X-Ray

Vous pouvez utiliser AWS X-Ray pour suivre les événements qui transitent par EventBridge. EventBridge transmet l'en-tête de suivi d'origine à la cible afin que les services cibles puissent suivre, analyser et déboguer les événements.

EventBridge peut transmettre un en-tête de suivi pour un événement uniquement si ce dernier provient d'une demande `PutEvents` qui a transmis le contexte de suivi. X-Ray ne suit pas les événements provenant de partenaires tiers, d'événements planifiés ou de services AWS et ces sources d'événements n'apparaissent pas sur la cartographie des services X-Ray.

X-Ray valide les en-têtes de suivi et ceux qui ne sont pas valides sont supprimés. Cependant, l'événement est toujours traité.

> **Important**
>
> L'en-tête de suivi n'est pas disponible sur l'événement livré à la cible d'invocation.
>
> - Si vous avez une archive d'événement, l'en-tête de suivi n'est pas disponible pour les événements archivés. Si vous relisez des événements archivés, l'en-tête de suivi n'est pas inclus.
> - Si vous avez une file d'attente de lettres mortes (DLQ), l'en-tête de suivi est inclus dans la demande `SendMessage` qui envoie l'événement à la DLQ. Si vous récupérez des événements (messages) de la DLQ en utilisant `ReceiveMessage`, l'en-tête de suivi associé à l'événement est inclus dans l'attribut de message Amazon SQS, mais il n'est pas inclus dans le message d'événement.

Pour en savoir plus sur la manière dont un nœud d'événement EventBridge connecte les services sources et cibles, consultez Affichage de la source et des cibles dans la cartographie des services X-Ray dans le Guide du développeur AWS X-Ray.

Vous pouvez transmettre les informations d'en-tête de suivi suivantes via EventBridge :

- En-tête HTTP par défaut : le kit SDK X-Ray remplit automatiquement l'en-tête de suivi en tant qu'en-tête HTTP `X-Amzn-Trace-Id` pour toutes les cibles d'invocation. Pour en savoir plus sur l'en-tête HTTP par défaut, consultez En-tête de suivi dans le Guide du développeur AWS X-Ray.

- Attribut système **`TraceHeader`** : `TraceHeader` est un attribut PutEventsRequestEntry réservé par EventBridge pour transporter l'en-tête de suivi X-Ray vers une cible. Si vous utilisez également `PutEventsRequestEntry`, `PutEventsRequestEntry` remplace l'en-tête de suivi HTTP.

> **Note**
>
> L'en-tête de suivi n'est pas pris en compte dans la taille de l'événement `PutEventsRequestEntry`. Pour de plus amples informations, veuillez consulter Calcul de la taille de EventBridge PutEvents l'entrée d'un événement Amazon.

La vidéo suivante illustre l'utilisation conjointe de X-Ray et d'EventBridge : Utilisation d'AWS X-Ray pour le suivi

Utilisation EventBridge avec le kit de test d'application AWS intégré

Lorsque vous créez des applications composées de services sans serveur tels que Lambda ou Step Functions EventBridge, de nombreux composants de votre architecture ne peuvent pas être déployés sur votre poste de travail, mais existent uniquement dans AWS le cloud. Contrairement à l'utilisation d'applications déployées localement, ces types d'applications bénéficient de stratégies basées sur le cloud pour effectuer des tests automatisés. AWS Le kit de test d'applications intégré (AWS IATK) vous aide à mettre en œuvre certaines de ces stratégies pour vos applications.

AWS IATK est une bibliothèque logicielle qui vous aide à écrire des tests automatisés pour les applications basées sur le cloud.

EventBridge intégration avec AWS IATK

Vous pouvez utiliser EventBridge des événements et des bus d'événements avec AWS IATK pour implémenter vos tests automatisés, notamment :

Implémentation de harnais de test

Pour écrire des tests d'intégration pour les architectures pilotées par des événements, établissez des limites logiques en divisant votre application en sous-systèmes. Une technique utile pour tester les sous-systèmes consiste à créer des harnais de test, c'est-à-dire des ressources que vous créez spécifiquement pour tester les sous-systèmes.

Par exemple, un test d'intégration peut démarrer un processus de sous-système en lui transmettant un événement de test d'entrée. AWS L'IATK peut créer pour vous un faisceau de test qui écoute les événements en sortie EventBridge . (En fait, le harnais est composé d'une EventBridge règle qui transmet l'événement de sortie à Amazon SQS.) Votre test d'intégration interroge ensuite le harnais de test pour examiner la sortie et déterminer si le test a réussi ou échoué.

Génération d'événements fictifs

AWS IATK vous permet de générer des événements fictifs à partir d'un schéma stocké dans le registre des EventBridge schémas. Cela vous permet de générer un événement fictif et d'invoquer n'importe quel consommateur (comme une fonction Lambda ou une machine d'état Step Functions) avec l'événement généré.

Pour plus d'informations, voir Présentation du kit de test d'application AWS intégré sur GitHub.

Inclure les EventBridge ressources Amazon dans des AWS CloudFormation piles

AWS CloudFormation vous permet de configurer et de gérer vos AWS ressources sur l'ensemble des comptes et des régions de manière centralisée et reproductible en traitant l'infrastructure comme du code. CloudFormation pour ce faire, vous pouvez créer des modèles qui définissent les ressources que vous souhaitez approvisionner et gérer. Ces ressources peuvent inclure EventBridge des artefacts tels que des bus et des règles d'événements, des canaux, des schémas et des horaires, entre autres. Utilisez ces ressources pour intégrer des EventBridge fonctionnalités aux ensembles technologiques que vous fournissez et gérez. CloudFormation

EventBridge Ressources Amazon disponibles dans AWS CloudFormation

EventBridge fournit des ressources à utiliser dans les CloudFormation modèles dans les espaces de noms de ressources suivants :

- AWS::Events

  Les exemples de modèles incluent :

  - Créez une destination d'API pour PagerDuty
  - Créer une destination d'API pour Slack
  - Création d'une connexion avec des paramètres ApiKey d'autorisation

  - Créer une connexion avec les paramètres d’autorisation OAuth
  - Créer un point de terminaison global avec la réplication d’événements
  - Refuser la politique à l’aide de plusieurs principaux et actions
  - Accorder une autorisation à une organisation en utilisant un bus d’événements personnalisé
  - Créer une règle entre régions
  - Créer une règle qui inclut une file d’attente de lettres mortes pour une cible
  - Invoquer régulièrement une fonction Lambda
  - Invoquer une fonction Lambda en réponse à un événement
  - Notifier une rubrique en réponse à une entrée de journal
- AWS::EventSchémas
- AWS::Pipes

  Les exemples de modèles incluent :

  - Créer un canal avec un filtre d’événements
- AWS::Scheduler

Génération de définitions EventBridge de ressources Amazon pour les AWS CloudFormation modèles

Pour vous aider à démarrer rapidement le développement de CloudFormation modèles, la EventBridge console vous permet de créer des CloudFormation modèles à partir des bus, règles et canaux d'événements existants dans votre compte.

- ???
- ???
- ???

AWS CloudFormation Gestion du bus d'événements par défaut

Comme le EventBridge bus d'événements par défaut est automatiquement intégré à votre compte, vous ne pouvez pas le créer à l'aide d'un CloudFormation modèle, comme vous le feriez normalement pour toute ressource que vous souhaitez inclure dans une CloudFormation pile. Pour inclure le bus d'événements par défaut dans une CloudFormation pile, vous devez d'abord l'importer

dans une pile. Une fois que vous avez importé le bus d'événements par défaut dans une pile, vous pouvez mettre à jour les propriétés du bus d'événements comme vous le souhaitez.

Pour plus d’informations, consultez ???.

Gestion des événements de AWS CloudFormation pile à l'aide de EventBridge

En plus d'inclure EventBridge des ressources dans vos CloudFormation piles, vous pouvez les utiliser EventBridge pour gérer les événements générés par les CloudFormation piles elles-mêmes. CloudFormation envoie des événements à EventBridge chaque fois qu'une opération de création, de mise à jour, de suppression ou de détection de dérive est effectuée sur une pile. CloudFormation envoie également des événements EventBridge pour modifier le statut des ensembles de piles et des instances d'ensembles de piles. Vous pouvez utiliser des EventBridge règles pour acheminer les événements vers les cibles que vous avez définies.

Pour plus d'informations, consultez la section Gestion des CloudFormation événements EventBridge à l'aide du Guide de AWS CloudFormation l'utilisateur.

Didacticiels Amazon EventBridge

EventBridge s'intègre avec un certain nombre de services AWS et de partenaires SaaS. Ces didacticiels sont conçus pour vous aider à vous familiariser avec les principes de base d'EventBridge et à la façon dont ce service peut être intégré à votre architecture sans serveur.

Didacticiels:

- Didacticiels de démarrage avec Amazon EventBridge
- Didacticiels Amazon EventBridge pour une intégration avec d'autres services AWS
- Didacticiels Amazon EventBridge pour une intégration avec des fournisseurs SaaS

Didacticiels de démarrage avec Amazon EventBridge

Les didacticiels suivants vous permettent d'explorer les fonctionnalités d'EventBridge et la façon de les utiliser.

Didacticiels:

- Archivage-relecture des événements Amazon EventBridge
- Création d'un exemple d'application Amazon EventBridge
- Didacticiel : Téléchargement de liaisons de code pour des événements à l'aide du registre de schémas EventBridge
- Didacticiel : Utilisation du transformateur d'entrée pour personnaliser les éléments qu'EventBridge transmet à la cible d'événement

Archivage-relecture des événements Amazon EventBridge

Vous pouvez utiliser EventBridge pour router des événements vers des fonctions AWS Lambda spécifiques à l'aide de règles.

Dans ce didacticiel, vous allez créer une fonction qui fera office de cible pour la règle EventBridge à l'aide de la console Lambda. Vous créerez ensuite une archive et une règle qui archivera des événements de test à l'aide de la console EventBridge. Dès qu'il y aura des événements dans cette archive, vous les relirez.

Étapes :

Étape 1 : Créer une fonction Lambda

Pour commencer, créez une fonction Lambda afin de journaliser les événements.

Pour créer une fonction Lambda :

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Sélectionnez Create function (Créer une fonction).
3. Choisissez Créer à partir de zéro.
4. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction `LogScheduledEvent`.
5. Gardez les autres options comme valeurs par défaut et choisissez Créer une fonction.
6. Dans l'onglet Code de la page de fonction, double-cliquez sur index.js.
7. Remplacez le code JavaScript existant par le code suivant :

```
'use strict';
```

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

8. Choisissez Deploy (Déployer).

Étape 2 : Créer l'archive

À présent, créez l'archive où seront conservés tous les événements de test.

Pour créer une archive

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le panneau de navigation, choisissez Archives.
3. Choisissez Créer une archive.
4. Entrez un nom et une description pour l'archive. Par exemple, nommez l'archive `ArchiveTest`.
5. Gardez les autres options comme valeurs par défaut et choisissez Suivant.
6. Choisissez Créer une archive.

Étape 3 : Créer une règle

Créez une règle pour archiver les événements qui sont envoyés au bus d'événements.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `ARTestRule`.

   Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un service AWS

de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.

6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. Pour Modèle d'événement, entrez ce qui suit :

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

10. Choisissez Next (Suivant).
11. Pour Types de cibles, choisissez service AWS.
12. Pour Sélectionner une cible, choisissez Fonction Lambda dans la liste déroulante.
13. Pour Fonction, sélectionnez la fonction Lambda que vous avez créée dans la section Étape 1 : Créer une fonction Lambda. Dans cet exemple, sélectionnez `LogScheduledEvent`.
14. Choisissez Next (Suivant).
15. Choisissez Next (Suivant).
16. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 4 : Envoyer des événements de test

Maintenant que vous avez configuré l'archive et la règle, nous allons envoyer des événements de test pour vérifier que l'archive fonctionne correctement.

> **Note**
>
> Les événements peuvent mettre un certain temps à parvenir à l'archive.

Pour envoyer des événements de test (console)

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.

2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Dans la vignette Bus d'événements par défaut, choisissez Actions, Envoyer des événements.
4. Entrez une source d'événements. Par exemple, `TestEvent`.
5. Pour Type de détails, entrez `customerCreated`.
6. Pour Détails de l'événement, entrez `{}`.
7. Sélectionnez Send (Envoyer).

Étape 5 : Relire les événements

Dès lors que les événements de test se trouvent dans l'archive, vous pouvez les relire.

Pour relire les événements archivés (console)

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le panneau de navigation, choisissez Relectures.
3. Choisissez Démarrer une nouvelle lecture.
4. Entrez un nom et une description pour le relecture. Par exemple, nommez la relecture `ReplayTest`.
5. Pour Source, sélectionnez l'archive que vous avez créée dans la section Étape 2 : Créer l'archive.
6. Pour Période de relecture, procédez comme suit.
    a. Pour Heure de début, sélectionnez la date à laquelle vous avez envoyé les événements de test et l'heure à laquelle vous les avez envoyés. Par exemple : `2021/08/11` et `08:00:00`.
    b. Pour Heure de fin, sélectionnez la date et l'heure actuelles. Par exemple : `2021/08/11` et `09:15:00`.
7. Choisissez Démarrer la relecture.

Étape 6 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les fonctions Lambda

1. Ouvrez la page Functions (Fonctions) de la console Lambda.
2. Sélectionnez la ou les fonctions que vous avez créées.
3. Sélectionnez Actions, Delete (Supprimer).
4. Choisissez Supprimer.

Pour supprimer la ou les archives EventBridge

1. Ouvrez la page Archives de la console EventBridge.
2. Sélectionnez la ou les archives que vous avez créées.
3. Choisissez Supprimer.
4. Entrez le nom de l'archive et choisissez Supprimer.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Sélectionnez Delete.

Création d'un exemple d'application Amazon EventBridge

Vous pouvez utiliser EventBridge pour router des événements vers des fonctions Lambda spécifiques à l'aide de règles.

Dans ce didacticiel, vous allez utiliser l'interface AWS CLI, Node.js et le code disponible dans le référentiel GitHub pour créer les éléments suivants :

- Une fonction AWS Lambda qui génère des événements pour les opérations de distributeur automatique de billets (ATM).
- Trois fonctions Lambda à utiliser comme cibles d'une règle EventBridge.
- La règle qui route les événements créés vers la fonction en aval appropriée sur la base d'un modèle d'événement.

Cet exemple utilise des modèles AWS SAM pour définir les règles EventBridge. Pour en savoir plus sur l'utilisation de modèles AWS SAM avec EventBridge, consultez ???.

Dans le référentiel, le sous-répertoire atmProducer contient `handler.js`, qui représente le service ATM qui génère les événements. Ce code est un gestionnaire Lambda écrit dans Node.js et qui publie les événements sur EventBridge via le kit AWS SDK en utilisant cette ligne de code JavaScript.

```
const result = await eventbridge.putEvents(params).promise()
```

Ce répertoire contient également `events.js`, qui répertorie plusieurs opérations de test dans un tableau d'entrées. Voici comment un événement unique est défini en JavaScript :

```
{
  // Event envelope fields
  Source: 'custom.myATMapp',
  EventBusName: 'default',
  DetailType: 'transaction',
  Time: new Date(),

  // Main event body
  Detail: JSON.stringify({
    action: 'withdrawal',
    location: 'MA-BOS-01',
    amount: 300,
    result: 'approved',
```

```
    transactionId: '123456',
    cardPresent: true,
    partnerBank: 'Example Bank',
    remainingFunds: 722.34
  })
}
```

La section Detail de l'événement spécifie les attributs d'opération. Il s'agit notamment de l'emplacement du distributeur automatique, du montant, de la banque partenaire et du résultat de l'opération.

Le fichier `handler.js` contenu dans le sous-répertoire atmConsumer contient trois fonctions :

```
exports.case1Handler = async (event) => {
  console.log('--- Approved transactions ---')
  console.log(JSON.stringify(event, null, 2))
}

exports.case2Handler = async (event) => {
  console.log('--- NY location transactions ---')
  console.log(JSON.stringify(event, null, 2))
}

exports.case3Handler = async (event) => {
  console.log('--- Unapproved transactions ---')
  console.log(JSON.stringify(event, null, 2))
}
```

Chaque fonction reçoit les événements d'opération, qui sont journalisés dans Amazon CloudWatch Logs via les instructions `console.log`. Les fonctions consommateur fonctionnent indépendamment du producteur et n'ont pas connaissance de la source des événements.

La logique de routage est contenue dans les règles EventBridge déployées par le modèle AWS SAM de l'application. Les règles évaluent le flux d'événements entrant et routent les événements correspondants vers les fonctions Lambda cibles.

Les règles utilisent des modèles d'événements qui sont des objets JSON dont la structure est la même que celle des événements correspondants. Voici le modèle d'événement pour l'une des règles.

```
{
  "detail-type": ["transaction"],
  "source": ["custom.myATMapp"],
```

```
  "detail": {
    "location": [{
      "prefix": "NY-"
    }]
  }
}
```

Étapes :

- Prérequis
- Étape 1 : Créer l'application
- Étape 2 : Exécuter l'application
- Étape 3 : Consulter les journaux et vérifier que l'application fonctionne
- Étape 4 : Nettoyer vos ressources

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des ressources suivantes :

- Un compte AWS. Créez un compte AWS si vous n'en avez pas déjà un.
- L'interface AWS CLI installée. Pour installer l'interface AWS CLI, consultez Installation, mise à jour et désinstallation de l'interface AWS CLI version 2.
- Node.js 12.x installé. Pour installer Node.js, accédez à la page de téléchargement.

Étape 1 : Créer l'application

Pour configurer l'exemple d'application, vous allez utiliser l'interface AWS CLI et Git pour créer les ressources AWS dont vous aurez besoin.

Pour créer l'application

1. Connectez-vous à AWS.
2. Installez Git et installez l'interface CLI AWS Serverless Application Model sur votre machine locale.
3. Créez un répertoire, puis accédez-y dans un terminal.
4. Dans la ligne de commande, entrez `git clone https://github.com/aws-samples/amazon-eventbridge-producer-consumer-example`.

5. Dans la ligne de commande, exécutez la commande suivante :

```
cd ./amazon-eventbridge-producer-consumer-example
sam deploy --guided
```

6. Dans le terminal, procédez comme suit :

   a. Dans **`Stack Name`**, entrez un nom pour la pile. Par exemple, nommez-la `Test`.
   b. Dans **`AWS Region`**, entrez la région. Par exemple, `us-west-2`.
   c. Pour **`Confirm changes before deploy`**, saisissez Y.
   d. Pour **`Allow SAM CLI IAM role creation`**, entrez Y.
   e. Pour **`Save arguments to configuration file`**, entrez Y.
   f. Pour **`SAM configuration file`**, saisissez `samconfig.toml`.
   g. Pour **`SAM configuration environment`**, saisissez `default`.

Étape 2 : Exécuter l'application

Maintenant que vous avez configuré les ressources, vous allez utiliser la console pour tester les fonctions.

Pour exécuter l'application

1. Ouvrez la console Lambda dans la région où vous avez déployé l'application AWS SAM.
2. Il existe quatre fonctions Lambda avec le préfixe atm-demo. Sélectionnez la fonction atmProducerFn, puis choisissez Actions, Tester.
3. Entrez `Test` en guise de Nom.
4. Choisissez Test (Tester).

Étape 3 : Consulter les journaux et vérifier que l'application fonctionne

Maintenant que vous avez exécuté l'application, vous allez utiliser la console pour consulter les journaux CloudWatch Logs.

Pour consulter les journaux

1. Ouvrez la console CloudWatch dans la région où vous avez exécuté l'application AWS SAM.
2. Choisissez Journaux, puis groupe de journaux.

3. Sélectionnez le groupe de journaux contenant atmConsumerCase1. Vous constatez la présence de deux flux représentant les deux opérations approuvées par le distributeur automatique de billets. Choisissez un flux de journaux pour examiner la sortie.
4. Revenez à la liste des groupes de journaux, puis sélectionnez le groupe de journaux contenant atmConsumerCase2. Vous y trouvez deux flux représentant les deux opérations correspondant au filtre de localisation New York.
5. Revenez à la liste des groupes de journaux et sélectionnez le groupe de journaux contenant atmConsumerCase3. Ouvrez le flux pour voir les opérations refusées.

Étape 4 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Choisissez Supprimer.

Pour supprimer la ou les fonctions Lambda

1. Ouvrez la page Functions (Fonctions) de la console Lambda.
2. Sélectionnez la ou les fonctions que vous avez créées.
3. Sélectionnez Actions, Delete (Supprimer).
4. Choisissez Supprimer.

Pour supprimer le ou les groupes de journaux CloudWatch Logs

1. Ouvrez la console CloudWatch.
2. Choisissez Journaux, Groupes de journaux.
3. Sélectionnez le ou les groupes de journaux qui ont été créés dans ce didacticiel.
4. Sélectionnez Actions, Delete log group(s) (Supprimer le ou les groupes de journaux).

5. Sélectionnez Delete.

Didacticiel : Téléchargement de liaisons de code pour des événements à l'aide du registre de schémas EventBridge

Vous pouvez générer des liaisons de code pour des schémas d'événements dans le but d'accélérer le développement pour Golang, Java, Python et TypeScript. Vous pouvez obtenir des liaisons de code pour les services AWS existants, les schémas que vous créez et les schémas que vous générez en fonction des événements d'un bus d'événements. Vous pouvez générer des liaisons de code pour un schéma en utilisant l'un des outils suivants :

- Console EventBridge
- API de registre de schémas EventBridge
- Votre environnement IDE avec une boîte à outils AWS Toolkit

Dans ce didacticiel, vous allez générer et télécharger des liaisons de code à partir d'un schéma EventBridge pour les événements d'un service AWS.

Pour générer des liaisons de code à partir d'un schéma EventBridge

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Schemas (Schémas).
3. Sélectionnez l'onglet Registre des schémas d'événements AWS.
4. Recherchez le schéma du service AWS pour lequel vous voulez générer des liaisons de code, soit en parcourant le registre de schémas, soit en effectuant une recherche.
5. Sélectionnez le nom du schéma.
6. Sur la page Informations de schéma, dans la section Version, sélectionnez Télécharger les liaisons de code.
7. Sur la page Download code bindings (Télécharger les liaisons de code), sélectionnez le langage des liaisons de code que vous souhaitez télécharger.
8. Sélectionnez Download (Télécharger).

   Quelques secondes peuvent être nécessaires avant que votre téléchargement commence. Le fichier de téléchargement est un fichier .zip contenant des liaisons de code pour le langage que vous avez sélectionné.
9. Décompressez le fichier téléchargé et ajoutez-le à votre projet.

Le package téléchargé contient un fichier README qui explique comment configurer les dépendances du package dans différents frameworks.

Utilisez ces liaisons de code dans votre propre code pour développer rapidement des applications en utilisant cet événement EventBridge.

Didacticiel : Utilisation du transformateur d'entrée pour personnaliser les éléments qu'EventBridge transmet à la cible d'événement

Vous pouvez utiliser le transformateur d'entrée dans EventBridge pour personnaliser le texte d'un événement avant de l'envoyer à la cible d'une règle.

Pour ce faire, vous définissez des chemins JSON depuis l'événement et affectez leurs sorties à différentes variables. Vous pouvez ensuite utiliser ces variables dans le modèle d'entrée. Les caractères < et > ne peuvent pas être placés dans une séquence d'échappement. Pour de plus amples informations, veuillez consulter Transformation des EventBridge entrées Amazon.

> **Note**
>
> 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, vous allez créer une règle qui correspond à un événement avec `detail-type: "customerCreated"`. Le transformateur d'entrée mappe la variable `type` au chemin JSON $.detail-type depuis l'événement. EventBridge place ensuite la variable dans le modèle d'entrée "This event was <type>." Le résultat est le message Amazon SNS suivant.

```
"This event was of customerCreated type."
```

Étapes :

- Étape 1 : Créer une rubrique Amazon SNS
- Étape 2 : Créer un abonnement Amazon SNS
- Étape 3 : Créer une règle
- Étape 4 : Envoyer des événements de test
- Étape 5 : Confirmer la bonne exécution
- Étape 6 : Nettoyer vos ressources

Étape 1 : Créer une rubrique Amazon SNS

Créez la rubrique qui doit recevoir les événements en provenance d'EventBridge.

Pour créer une rubrique

1. Ouvrez la console Amazon SNS à partir de l'adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le panneau de navigation, sélectionnez Topics (Rubriques).
3. Choisissez Create topic (Créer une rubrique).
4. Pour Type, choisissez Standard.
5. Entrez **`eventbridge-IT-test`** comme nom de la rubrique.
6. Choisissez Create topic (Créer une rubrique).

Étape 2 : Créer un abonnement Amazon SNS

Créez un abonnement pour obtenir des e-mails contenant les informations transformées.

Pour créer un abonnement

1. Ouvrez la console Amazon SNS à partir de l'adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le panneau de navigation, sélectionnez Abonnements.
3. Sélectionnez Créer un abonnement.
4. Pour ARN de la rubrique, choisissez la rubrique que vous avez créée à l'étape 1. Pour ce didacticiel, choisissez eventbridge-IT-test.
5. Pour Protocole, choisissez E-mail.
6. Saisissez votre adresse e-mail dans Endpoint (Point de terminaison).
7. Choisissez Create subscription (Créer un abonnement).
8. Confirmez l'abonnement en choisissant Confirmer l'abonnement dans l'e-mail que vous recevez du service de notifications AWS.

Étape 3 : Créer une règle

Créez une règle de sorte qu'elle utilise le transformateur d'entrée pour personnaliser les informations d'état d'instance à destination d'une cible.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.

2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `ARTestRule`
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.
6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. Pour Modèle d'événement, entrez ce qui suit :

```
{
  "detail-type": [
    "customerCreated"
  ]
}
```

10. Choisissez Next (Suivant).
11. Pour Types de cibles, choisissez service AWS.
12. Pour Sélectionner une cible, choisissez Rubrique SNS dans la liste déroulante.
13. Pour Rubrique, sélectionnez la rubrique Amazon SNS que vous avez créée à l'étape 1. Pour ce didacticiel, choisissez eventbridge-IT-test.
14. Pour Réglages supplémentaires, procédez comme suit :

    a. Pour Configurer l'entrée cible, choisissez Transformateur d'entrée dans la liste déroulante.

    b. Choisissez Configurer le transformateur d'entrée.

    c. Pour Exemples d'événements, entrez ce qui suit :

```
{
  "detail-type": "customerCreated"
}
```

    d. Pour Transformateur d'entrée cible, procédez comme suit :

i. Pour Chemin d'entrée, entrez ce qui suit :

```
{"detail-type":"$.detail-type"}
```

ii. Pour Modèle d'entrée, entrez ce qui suit :

```
"This event was of <detail-type> type."
```

e. Choisissez Confirmer.

15. Choisissez Next (Suivant).
16. Choisissez Next (Suivant).
17. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 4 : Envoyer des événements de test

Maintenant que vous avez configuré la rubrique SNS et la règle, nous allons envoyer des événements de test pour vérifier que la règle fonctionne correctement.

Pour envoyer des événements de test (console)

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, sélectionnez Event Buses (Bus d'événements).
3. Dans la vignette Bus d'événements par défaut, choisissez Actions, Envoyer des événements.
4. Entrez une source d'événements. Par exemple, `TestEvent`.
5. Pour Type de détails, entrez `customerCreated`.
6. Pour Détails de l'événement, entrez `{}`.
7. Sélectionnez Send (Envoyer).

Étape 5 : Confirmer la bonne exécution

Si vous obtenez un e-mail du service de notifications AWS qui correspond à la sortie attendue, c'est que vous avez correctement effectué le didacticiel.

Étape 6 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la rubrique SNS

1. Ouvrez la page Rubriques de la console SNS.
2. Sélectionnez la rubrique que vous avez créée.
3. Choisissez Supprimer.
4. Saisissez **`delete me`**.
5. Choisissez Supprimer.

Pour supprimer l'abonnement SNS

1. Ouvrez la page Abonnements de la console SNS.
2. Sélectionnez l'abonnement que vous avez créé.
3. Choisissez Supprimer.
4. Choisissez Supprimer.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Sélectionnez Delete.

Didacticiels Amazon EventBridge pour une intégration avec d'autres services AWS

Amazon EventBridge fait appel à d'autres services AWS pour traiter les événements ou invoquer une ressource AWS en tant que cible d'une règle. Les didacticiels suivants vous montrent comment intégrer EventBridge avec d'autres services AWS.

Didacticiels:

- Didacticiel : Journalisation de l'état d'un groupe Auto Scaling à l'aide d'EventBridge
- Tutoriel : Consigner les appels AWS d'API à l'aide EventBridge
- Tutoriel : enregistrez l'état d'une instance Amazon EC2 à l'aide de EventBridge
- Didacticiel : Journalisation des opérations au niveau de l'objet Amazon S3 à l'aide d'EventBridge
- Tutoriel : Envoyer des événements vers un flux Amazon Kinesis à l'aide du EventBridge schéma aws.events
- Didacticiel : Planification d'instantanés automatisés Amazon EBS à l'aide d'EventBridge
- Didacticiel : Envoi d'une notification lors de la création d'un objet Amazon S3
- Didacticiel : Planification de fonctions AWS Lambda à l'aide d'EventBridge

Didacticiel : Journalisation de l'état d'un groupe Auto Scaling à l'aide d'EventBridge

Vous pouvez exécuter une fonction AWS Lambda qui journalise un événement chaque fois qu'un groupe Auto Scaling lance ou résilie une instance Amazon EC2 qui indique si un événement a abouti.

Pour en savoir plus sur les autres scénarios qui utilisent des événements Amazon EC2 Auto Scaling, consultez Utilisation d'EventBridge pour gérer les événements Auto Scaling dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Dans ce didacticiel, vous allez créer une fonction Lambda et une règle dans la console EventBridge qui appellera cette fonction lorsqu'un groupe Amazon EC2 Auto Scaling lancera ou résiliera une instance.

Étapes :

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des ressources suivantes :

- Un groupe Auto Scaling. Pour savoir comment en créer un, consultez Création d'un groupe Auto Scaling à l'aide d'une configuration de lancement dans le Guide de l'utilisateur Amazon EC2 Auto Scaling.

Étape 1 : Créer une fonction Lambda

Créez une fonction Lambda pour enregistrer les événements de montée et de diminution en charge de votre groupe Auto Scaling.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Sélectionnez Create function (Créer une fonction).
3. Choisissez Créer à partir de zéro.
4. Entrez un nom pour la fonction Lambda. Par exemple, nommez la fonction `LogAutoScalingEvent`.
5. Gardez les autres options comme valeurs par défaut et choisissez Créer une fonction.
6. Dans l'onglet Code de la page de fonction, double-cliquez sur index.js.
7. Remplacez le code existant par le code suivant.

```
'use strict';

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

8. Choisissez Deploy (Déployer).

Étape 2 : création d'une règle

Créez une règle pour exécuter la fonction Lambda que vous avez créée à l'étape 1. La règle s'exécute lorsque votre groupe Auto Scaling démarre ou arrête une instance.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `TestRule`
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.

6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).
8. Pour Event source (Origine de l'événement), choisissez AWSservices (Services).
9. Pour Event pattern (Modèle d'événement), procédez comme suit :

    a. Pour Source d'événement, sélectionnez Auto Scaling dans la liste déroulante.
    b. Pour Type d'événement, sélectionnez Lancement et résiliation d'une instance dans la liste déroulante.
    c. Choisissez Tout événement d'instance et Tout nom de groupe.
10. Choisissez Next (Suivant).
11. Pour Types de cibles, choisissez service AWS.
12. Pour Sélectionner une cible, choisissez Fonction Lambda dans la liste déroulante.
13. Pour Fonction, sélectionnez la fonction Lambda que vous avez créée dans la section Étape 1 : Créer une fonction Lambda. Dans cet exemple, sélectionnez `LogAutoScalingEvent`.
14. Choisissez Next (Suivant).
15. Choisissez Next (Suivant).
16. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

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. Attendez quelques minutes le temps que l'événement de montée en puissance se produise, puis vérifiez que votre fonction Lambda a été invoquée.

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 Auto Scaling Groups (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, entrez 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, sélectionnez Enregistrer.

2. Pour afficher la sortie de la fonction Lambda, procédez de la manière suivante :

   a. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
   b. Dans le panneau de navigation, sélectionnez Logs (Journaux).
   c. Sélectionnez le nom du groupe de journaux pour votre fonction Lambda (`/aws/lambda/function-name`).
   d. Sélectionnez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez lancée.

3. (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.

Étape 4 : Confirmer la bonne exécution

Si l'événement Lambda se trouve dans les journaux CloudWatch, cela signifie que vous avez correctement effectué ce didacticiel. Si l'événement ne figure pas dans vos journaux CloudWatch, essayez de résoudre le problème en vérifiant d'abord que la règle a bien été créée. Si celle-ci semble correcte, vérifiez que le code de votre fonction Lambda l'est également.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Choisissez Supprimer.

Pour supprimer la ou les fonctions Lambda

1. Ouvrez la page Functions (Fonctions) de la console Lambda.

2. Sélectionnez la ou les fonctions que vous avez créées.
3. Sélectionnez Actions, Delete (Supprimer).
4. Sélectionnez Delete.

Tutoriel : Consigner les appels AWS d'API à l'aide EventBridge

Vous pouvez utiliser EventBridge les règles Amazon pour réagir aux appels d'API effectués par un AWS service et enregistrés par AWS CloudTrail.

Dans ce didacticiel, vous allez créer une AWS CloudTrailpiste, une fonction Lambda et une règle dans la EventBridge console. La règle invoque la fonction Lambda lorsqu'une instance Amazon EC2 est arrêtée.

Étapes :

- Étape 1 : Création d'un AWS CloudTrail parcours
- Étape 2 : Créer une fonction AWS Lambda
- Étape 3 : Créer une règle
- Étape 4 : Tester la règle
- Étape 5 : Confirmer la bonne exécution
- Étape 6 : Nettoyer vos ressources

Étape 1 : Création d'un AWS CloudTrail parcours

Si vous disposez déjà d'un journal de suivi configuré, passez directement à l'étape 2.

Pour créer un journal de suivi

1. Ouvrez la CloudTrail console 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 Emplacement de stockage, choisissez Créer un compartiment S3.
5. Dans Alias AWS KMS , entrez un alias pour la clé KMS.
6. Choisissez Suivant.
7. Choisissez Suivant.
8. Choisissez Create trail (Créer un journal de suivi).

Étape 2 : Créer une fonction AWS Lambda

Créez une fonction Lambda pour enregistrer les événements d'appels d'API.

Pour créer une fonction Lambda

1. Ouvrez la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/.
2. Choisissez Créer une fonction.
3. Choisissez Créer à partir de zéro.
4. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction `LogEC2StopInstance`.
5. Gardez les autres options comme valeurs par défaut et choisissez Créer une fonction.
6. Dans l'onglet Code de la page de fonction, double-cliquez sur index.js.
7. Remplacez le code existant par le code suivant.

```
'use strict';

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

8. Choisissez Deploy (Déployer).

Étape 3 : Créer une règle

Créez une règle de sorte qu'elle exécute la fonction Lambda que vous avez créée à l'étape 2 chaque fois que vous arrêtez une instance Amazon EC2.

Pour créer une règle

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `TestRule`
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.

6. Pour Type de règle, choisissez Règle avec un modèle d’événement.
7. Choisissez Suivant.
8. Pour Source d'événement, choisissez Services AWS .
9. Pour Event pattern (Modèle d’événement), procédez comme suit :
    a. Pour Source d’événement, sélectionnez EC2 dans la liste déroulante.
    b. Pour le type d'événement, sélectionnez AWS API Call via CloudTrail dans la liste déroulante.
    c. Choisissez Opération(s) spécifique(s) et entrez `StopInstances`.
10. Choisissez Suivant.
11. Pour Types de cibles, choisissez service AWS .
12. Pour Sélectionner une cible, choisissez Fonction Lambda dans la liste déroulante.
13. Pour Fonction, sélectionnez la fonction Lambda que vous avez créée dans la section Étape 1 : Créer une fonction Lambda. Dans cet exemple, sélectionnez `LogEC2StopInstance`.
14. Choisissez Suivant.
15. Choisissez Suivant.
16. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 4 : Tester la règle

Vous pouvez tester votre règle en arrêtant une instance Amazon EC2 à l'aide de la console Amazon EC2. Attendez quelques minutes que l'instance s'arrête, puis vérifiez vos AWS Lambda métriques sur la CloudWatch console pour vérifier que votre fonction s'est exécutée.

Pour tester votre règle en arrêtant une instance

1. Ouvrez la console Amazon EC2 à l’adresse https://console.aws.amazon.com/ec2/.
2. Lancez une instance. Pour plus d'informations, consultez la section Lancer votre instance dans le guide de l'utilisateur Amazon EC2.
3. Arrêtez l’instance. Pour plus d'informations, consultez la section Arrêter et démarrer votre instance dans le guide de l'utilisateur Amazon EC2.
4. Pour afficher la sortie de la fonction Lambda, procédez de la manière suivante :
    a. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.

b. Dans le panneau de navigation, sélectionnez Logs (Journaux).

c. Sélectionnez le nom du groupe de journaux pour votre fonction Lambda (`/aws/lambda/function-name`).

d. Sélectionnez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez arrêtée.

5. (Facultatif) Lorsque vous avez terminé, mettez fin à l'instance arrêtée. Pour plus d'informations, consultez la section Résiliation de votre instance dans le guide de l'utilisateur Amazon EC2.

Étape 5 : Confirmer la bonne exécution

Si vous voyez l'événement Lambda dans les CloudWatch journaux, cela signifie que vous avez terminé ce didacticiel avec succès. Si l'événement ne figure pas dans vos CloudWatch journaux, commencez le dépannage en vérifiant que la règle a été créée avec succès et, si la règle semble correcte, vérifiez que le code de votre fonction Lambda est correct.

Étape 6 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. En supprimant AWS les ressources que vous n'utilisez plus, vous évitez des frais inutiles sur votre AWS compte.

Pour supprimer la ou les EventBridge règles

1. Ouvrez la page Règles de la EventBridge console.
2. Sélectionnez la ou les règles que vous avez créées.
3. Sélectionnez Delete.
4. Sélectionnez Delete.

Pour supprimer la ou les fonctions Lambda

1. Ouvrez la page Functions (Fonctions) de la console Lambda.
2. Sélectionnez la ou les fonctions que vous avez créées.
3. Sélectionnez Actions, Supprimer.
4. Sélectionnez Supprimer.

Pour supprimer le ou les CloudTrail parcours

1. Ouvrez la page Trails de la CloudTrail console.
2. Sélectionnez le ou les journaux de suivi que vous avez créés.
3. Sélectionnez Delete.
4. Sélectionnez Supprimer.

Tutoriel : enregistrez l'état d'une instance Amazon EC2 à l'aide de EventBridge

Vous pouvez créer une fonction AWS Lambda qui journalise un changement d'état pour une instance Amazon EC2. Vous pouvez donc créer une règle qui exécute la fonction Lambda à chaque changement d'état ou chaque fois qu'une transition vers un ou plusieurs états notables se produit. Dans ce didacticiel, vous consignez le lancement d'une nouvelle instance.

Étapes :

- Étape 1 : création d'une fonction AWS Lambda
- Étape 2 : création d'une règle
- Étape 3 : test de la règle
- Étape 4 : Confirmer la bonne exécution
- Étape 5 : Nettoyer vos ressources

Étape 1 : création d'une fonction AWS Lambda

Créez une fonction Lambda pour journaliser les évènements de changement d'état. Lorsque vous créerez la règle à l'étape 2, vous spécifierez cette fonction.

Pour créer une fonction Lambda

1. Ouvrez la AWS Lambda console à l'adresse https://console.aws.amazon.com/lambda/.
2. Choisissez Créer une fonction.
3. Choisissez Créer à partir de zéro.
4. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction `LogEC2InstanceStateChange`.
5. Gardez les autres options comme valeurs par défaut et choisissez Créer une fonction.
6. Dans l'onglet Code de la page de fonction, double-cliquez sur index.js.
7. Remplacez le code existant par le code suivant.

```
'use strict';

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

```
    callback(null, 'Finished');
};
```

8. Choisissez Deploy (Déployer).

Étape 2 : création d'une règle

Créez une règle pour exécuter la fonction Lambda que vous avez créée à l’étape 1. La règle s’exécute lorsque vous lancez une instance Amazon EC2.

Pour créer la EventBridge règle

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `TestRule`
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.
6. Pour Type de règle, choisissez Règle avec un modèle d’événement.
7. Choisissez Suivant.
8. Pour Source d'événement, choisissez Services AWS .
9. Pour Event pattern (Modèle d’événement), procédez comme suit :
    a. Pour Source d’événement, sélectionnez EC2 dans la liste déroulante.
    b. Pour Type d’événement, choisissez Notifications de changement d’état d’instance EC2.
    c. Choisissez État(s) spécifique(s), puis En cours d’exécution dans la liste déroulante.
    d. Sélectionnez Toute instance.
10. Choisissez Suivant.
11. Pour Types de cibles, choisissez service AWS .
12. Pour Sélectionner une cible, choisissez Fonction Lambda dans la liste déroulante.
13. Pour Fonction, sélectionnez la fonction Lambda que vous avez créée dans la section Étape 1 : Créer une fonction Lambda. Dans cet exemple, sélectionnez `LogEC2InstanceStateChange`.

14. Choisissez Suivant.
15. Choisissez Suivant.
16. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

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. Attendez quelques minutes que l'instance s'arrête, puis vérifiez vos AWS Lambda métriques sur la CloudWatch console pour vérifier que votre fonction s'est exécutée.

Pour tester votre règle en arrêtant une instance

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Lancez une instance. Pour plus d'informations, consultez la section Lancer votre instance dans le guide de l'utilisateur Amazon EC2.
3. Arrêtez l'instance. Pour plus d'informations, consultez la section Arrêter et démarrer votre instance dans le guide de l'utilisateur Amazon EC2.
4. Pour afficher la sortie de la fonction Lambda, procédez de la manière suivante :
   a. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
   b. Dans le panneau de navigation, sélectionnez Logs (Journaux).
   c. Sélectionnez le nom du groupe de journaux pour votre fonction Lambda (`/aws/lambda/function-name`).
   d. Sélectionnez le nom du flux de journaux pour afficher les données fournies par la fonction concernant l'instance que vous avez arrêtée.
5. (Facultatif) Lorsque vous avez terminé, mettez fin à l'instance arrêtée. Pour plus d'informations, consultez la section Résiliation de votre instance dans le guide de l'utilisateur Amazon EC2.

Étape 4 : Confirmer la bonne exécution

Si vous voyez l'événement Lambda dans les CloudWatch journaux, cela signifie que vous avez terminé ce didacticiel avec succès. Si l'événement ne figure pas dans vos CloudWatch journaux, commencez le dépannage en vérifiant que la règle a été créée avec succès et, si la règle semble correcte, vérifiez que le code de votre fonction Lambda est correct.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. En supprimant AWS les ressources que vous n'utilisez plus, vous évitez des frais inutiles sur votre AWS compte.

Pour supprimer la ou les EventBridge règles

1. Ouvrez la page Règles de la EventBridge console.
2. Sélectionnez la ou les règles que vous avez créées.
3. Sélectionnez Delete.
4. Sélectionnez Delete.

Pour supprimer la ou les fonctions Lambda

1. Ouvrez la page Functions (Fonctions) de la console Lambda.
2. Sélectionnez la ou les fonctions que vous avez créées.
3. Sélectionnez Actions, Supprimer.
4. Sélectionnez Supprimer.

Didacticiel : Journalisation des opérations au niveau de l'objet Amazon S3 à l'aide d'EventBridge

Vous pouvez journaliser les opérations d'API au niveau de l'objet dans vos compartiments Amazon S3. Pour permettre à Amazon EventBridge de mettre en correspondance ces événements, vous devez d'abord utiliser AWS CloudTrail pour créer et configurer un journal de suivi destiné à recevoir ces événements.

Dans ce didacticiel, vous allez créer successivement un journal de suivi CloudTrail, une fonction AWS Lambda, puis une règle dans la console EventBridge qui invoquera cette fonction en réponse à un événement de données S3.

Étapes :

- Étape 1 : Configuration de votre journal de suivi AWS CloudTrail
- Étape 2 : Créer une fonction AWS Lambda
- Étape 3 : Création d'une règle
- Étape 4 : Test de la règle
- Étape 5 : Confirmer la bonne exécution
- Étape 6 : Nettoyer vos ressources

Étape 1 : Configuration de votre journal de suivi AWS CloudTrail

Pour journaliser les événements de données d'un compartiment S3 dans AWS CloudTrail et EventBridge, commencez par créer un journal de suivi. Un journal de suivi capture les appels d'API et les événements associés de votre compte et transmet les fichiers journaux au compartiment S3 que vous spécifiez. Vous pouvez mettre à jour un journal de suivi existant ou en créer une nouveau.

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

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 Emplacement de stockage, choisissez Créer un compartiment S3.

5. Dans Alias AWS KMS, entrez un alias pour la clé KMS.
6. Choisissez Next (Suivant).
7. Pour Type d'événement, choisissez Événements de données
8. Pour Événements de données, effectuez l'une des opérations suivantes :

   - 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 journaliser les événements de données relatifs à certains objets Amazon S3, spécifiez un compartiment S3 et le préfixe d'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.
9. Pour chaque ressource, optez pour une journalisation des événements en Lecture, en Écriture ou les deux.
10. Choisissez Next (Suivant).
11. Choisissez Create trail (Créer un journal de suivi).

Étape 2 : Créer une fonction AWS Lambda

Créez une fonction Lambda pour enregistrer les événements de données de vos compartiments S3.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Sélectionnez Create function (Créer une fonction).
3. Choisissez Créer à partir de zéro.
4. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction `LogS3DataEvents`.
5. Gardez les autres options comme valeurs par défaut et choisissez Créer une fonction.
6. Dans l'onglet Code de la page de fonction, double-cliquez sur index.js.
7. Remplacez le code existant par le code suivant.

```
'use strict';

exports.handler = (event, context, callback) => {
```

```
    console.log('LogS3DataEvents');
    console.log('Received event:', JSON.stringify(event, null, 2));
    callback(null, 'Finished');
};
```

8. Choisissez Deploy (Déployer).

Étape 3 : Création d'une règle

Créez une règle pour exécuter la fonction Lambda que vous avez créée à l'étape 2. Cette règle s'exécutera en réponse à un événement de données Amazon S3.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `TestRule`
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle mette en correspondance les événements en provenance de votre compte, sélectionnez Par défaut. Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.
6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).
8. Pour Event source (Origine de l'événement), choisissez AWSservices (Services).
9. Pour Event pattern (Modèle d'événement), procédez comme suit :

    a. Sous Source d'événement, sélectionnez Simple Storage Service (S3) dans la liste déroulante.

    b. Pour Type d'événement, sélectionnez Appel d'API au niveau de l'objet via CloudTrail dans la liste déroulante.

    c. Choisissez Specific operation(s) (Opération(s) spécifique(s)), puis PutObject.

    d. Par défaut, la règle correspond aux événements de données pour tous les compartiments de la région. Pour faire correspondre des événements de données pour des compartiments

spécifiques, choisissez Specify bucket(s) by name (Spécifier les compartiments par nom), puis précisez un ou plusieurs compartiments.

10. Choisissez Next (Suivant).
11. Pour Types de cibles, choisissez service AWS.
12. Pour Sélectionner une cible, choisissez Fonction Lambda dans la liste déroulante.
13. Pour Fonction, sélectionnez la fonction Lambda `LogS3DataEvents` que vous avez créée à l'étape 1.
14. Choisissez Next (Suivant).
15. Choisissez Next (Suivant).
16. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

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 panneau de navigation, sélectionnez Logs (Journaux).
3. Sélectionnez le nom du groupe de journaux pour votre 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 examiner vos journaux CloudTrail dans le compartiment S3 que vous avez spécifié pour votre journal de suivi. Pour plus d'informations, consultez Obtention et consultation des fichiers journaux CloudTrail dans le AWS CloudTrail Guide de l'utilisateur.

Étape 5 : Confirmer la bonne exécution

Si l'événement Lambda se trouve dans les journaux CloudWatch, cela signifie que vous avez correctement effectué ce didacticiel. Si l'événement ne figure pas dans vos journaux CloudWatch, essayez de résoudre le problème en vérifiant d'abord que la règle a bien été créée. Si celle-ci semble correcte, vérifiez que le code de votre fonction Lambda l'est également.

Étape 6 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Choisissez Supprimer.

Pour supprimer la ou les fonctions Lambda

1. Ouvrez la page Functions (Fonctions) de la console Lambda.
2. Sélectionnez la ou les fonctions que vous avez créées.
3. Sélectionnez Actions, Delete (Supprimer).
4. Choisissez Supprimer.

Pour supprimer le ou les journaux de suivi CloudTrail

1. Ouvrez la page Trails (Journaux de suivi) de la console CloudTrail.
2. Sélectionnez le ou les journaux de suivi que vous avez créés.
3. Choisissez Supprimer.
4. Sélectionnez Delete.

Tutoriel : Envoyer des événements vers un flux Amazon Kinesis à l'aide du EventBridge schéma **aws.events**

Vous pouvez envoyer des événements d'appel d' AWS API EventBridge vers un flux Amazon Kinesis, créer des applications Kinesis Data Streams et traiter de grandes quantités de données. Dans ce didacticiel, vous allez créer un flux Kinesis, puis créer une règle dans la EventBridge console qui envoie des événements à ce flux lorsqu'une instance Amazon EC2 s'arrête.

Étapes :

Prérequis

Dans ce didacticiel, vous allez utiliser :

- Utilisez le AWS CLI pour travailler avec les flux Kinesis.

  Pour l'installer AWS CLI, consultez la section Installation, mise à jour et désinstallation de la AWS CLI version 2.

> **Note**
>
> Ce didacticiel utilise AWS les événements et le registre de `aws.events` schémas intégré. Vous pouvez également créer une EventBridge règle basée sur le schéma de vos événements personnalisés en les ajoutant manuellement à un registre de schémas personnalisé ou en utilisant la découverte de schémas.
> Pour plus d'informations sur les schémas, consultez ???. Pour plus d'informations sur la création d'une règle avec d'autres options de modèle d'événement, consultez ???.

Étape 1 : Créer un flux Amazon Kinesis

Pour créer un flux, utilisez la commande à l'invite de `create-stream` AWS CLI commande.

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

Lorsque le statut du flux est `ACTIVE`, le flux est prêt. Pour vérifier le statut du flux, utilisez la commande `describe-stream`.

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

Étape 2 : création d'une règle

Créez une règle de sorte que des événements soient envoyés à votre flux lorsque vous arrêtez une instance Amazon EC2.

Pour créer une règle

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `TestRule`
5. Pour Bus d'événements, sélectionnez Par défaut.
6. Pour Type de règle, choisissez Règle avec un modèle d'événement.
7. Choisissez Suivant.
8. Dans Source de l'événement, choisissez AWS des événements ou des événements EventBridge partenaires.
9. Pour Méthode de création, choisissez Utiliser le schéma.
10. Pour Event pattern (Modèle d'événement), procédez comme suit :

    a. Pour Type de schéma, choisissez Sélectionner un schéma dans le registre des schémas.

    b. Pour Registre des schémas, choisissez aws.events dans la liste déroulante.

    c. Pour Schema, choisissez aws.ec2 @EC2 dans la InstanceStateChangeNotification liste déroulante.

    EventBridge affiche le schéma d'événement sous Modèles.

EventBridge affiche un astérisque rouge à côté des propriétés requises pour l'événement, et non pour le modèle d'événement.

d. Dans Modèles, définissez les propriétés de filtre d'événements suivantes :

i. Sélectionnez + Modifier en regard de la propriété state.

Laissez Relation vide. Pour le champ Valeur, saisissez `running`. Choisissez Définir.

ii. Sélectionnez + Modifier en regard de la propriété source.

Laissez Relation vide. Pour le champ Valeur, saisissez `aws.ec2`. Choisissez Définir.

iii. Sélectionnez + Modifier en regard de la propriété detail-type.

Laissez Relation vide. Pour le champ Valeur, saisissez `EC2 Instance State-change Notification`. Choisissez Définir.

e. Pour afficher le modèle d'événement que vous avez construit, choisissez Générer un modèle d'événement au format JSON

EventBridge affiche le modèle d'événement au format JSON :

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

11. Choisissez Suivant.
12. Pour Types de cibles, choisissez service AWS .
13. Pour Sélectionner une cible, choisissez Flux Kinesis dans la liste déroulante.
14. Pour Flux, sélectionnez le flux Kinesis que vous avez créé dans la section Étape 1 : Créer un flux Amazon Kinesis. Dans cet exemple, sélectionnez `test`.
15. Pour Rôle d'exécution, choisissez Créer un rôle pour cette ressource spécifique.
16. Choisissez Suivant.
17. Choisissez Suivant.
18. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 3 : test de la règle

Pour tester la règle, arrêtez une instance Amazon EC2. Attendez quelques minutes que l'instance s'arrête, puis vérifiez vos CloudWatch métriques pour vérifier que votre fonction s'est exécutée.

Pour tester votre règle en arrêtant une instance

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Lancez une instance. Pour plus d'informations, consultez la section Lancer votre instance dans le guide de l'utilisateur Amazon EC2.
3. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
4. Dans le volet de navigation, choisissez Règles.

   Choisissez le nom de la règle que vous avez créée, puis choisissez Metrics for the rule (Métriques de la règle).
5. (Facultatif) Lorsque vous avez terminé, mettez fin à l'instance. Pour plus d'informations, consultez la section Résiliation de votre instance dans le guide de l'utilisateur Amazon EC2.

Étape 4 : Vérifier que l'événement a été envoyé

Vous pouvez utiliser le AWS CLI pour obtenir l'enregistrement du flux afin de vérifier que l'événement a été envoyé.

Pour obtenir l'enregistrement

1. Pour commencer à lire à partir de votre flux Kinesis, à l'invite de commande, utilisez la commande `get-shard-iterator`.

   ```
   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. Pour obtenir l'enregistrement, utilisez la commande `get-records` suivante. Utilisez l'itérateur de partition dans la sortie de 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`.
3. Dans Kinesis, les enregistrements sont encodés en Base64. Utilisez un décodeur Base64 pour décoder les données afin de pouvoir vérifier qu'il s'agit bien de l'événement qui été envoyé au flux au format JSON.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. En supprimant AWS les ressources que vous n'utilisez plus, vous évitez des frais inutiles sur votre AWS compte.

Pour supprimer la ou les EventBridge règles

1. Ouvrez la page Règles de la EventBridge console.
2. Sélectionnez la ou les règles que vous avez créées.
3. Sélectionnez Delete.
4. Sélectionnez Delete.

Pour supprimer le ou les flux Kinesis

1. Ouvrez la page Flux de données de la console Kinesis.
2. Sélectionnez le ou les flux que vous avez créés.
3. Sélectionnez Actions, Supprimer.
4. Entrez supprimer dans le champ et choisissez Supprimer.

Didacticiel : Planification d'instantanés automatisés Amazon EBS à l'aide d'EventBridge

Vous pouvez exécuter des règles EventBridge selon une planification. Dans ce didacticiel, vous allez créer un instantané d'un volume Amazon Elastic Block Store (Amazon EBS) existant selon une planification. Vous pouvez créer un instantané toutes les X minutes en choisissant une fréquence fixe ou créer l'instantané à une heure précise de la journée en utilisant expression cron.

> **Important**
>
> Pour créer des règles avec des cibles intégrées, vous devez utiliser la AWS Management Console.

Étapes :

- Étape 1 : Créer la règle
- Étape 2 : Tester la règle
- Étape 3 : Confirmer la bonne exécution
- Étape 4 : Nettoyer vos ressources

Étape 1 : Créer la 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 Création d'une règle Amazon EventBridge qui s'exécute selon un calendrier.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle.

   Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.

5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle corresponde aux événements provenant de votre compte, sélectionnez default event bus (Bus d'événement AWS par défaut). Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.
6. Pour Rule type (Type de règle), choisissez Schedule (Planifier).
7. Choisissez Next (Suivant).
8. Pour Modèle de planification, choisissez Un programme qui s'exécute à fréquence régulière, par exemple toutes les 10 minutes, entrez **5**, puis choisissez Minutes dans la liste déroulante.
9. Choisissez Next (Suivant).
10. Pour Types de cibles, choisissez service AWS.
11. Pour Sélectionner une cible, choisissez Création d'EBS Snapshot dans la liste déroulante.
12. Pour ID du volume, entrez l'ID du volume Amazon EBS.
13. Pour Rôle d'exécution, choisissez Créer un rôle pour cette ressource spécifique.
14. Choisissez Next (Suivant).
15. Choisissez Next (Suivant).
16. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 2 : Tester la règle

Vous pouvez vérifier que votre règle fonctionne en examinant votre premier instantané après l'avoir créé.

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.

Étape 3 : Confirmer la bonne exécution

La présence d'un instantané dans la liste indique que vous avez correctement effectué ce didacticiel. Si l'instantané ne se trouve pas dans la liste, essayez de résoudre le problème en vérifiant d'abord que la règle a bien été créée.

Étape 4 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Sélectionnez Delete.

Didacticiel : Envoi d'une notification lors de la création d'un objet Amazon S3

Vous pouvez envoyer des notifications par e-mail lorsque des objets Amazon Simple Storage Service (Amazon S3) sont créés avec Amazon EventBridge et Amazon SNS. Dans ce didacticiel, vous allez créer une rubrique et un abonnement SNS. Ensuite, vous créerez une règle dans la console EventBridge qui enverra des événements à cette rubrique à réception d'événements `Object Created` Amazon S3.

Étapes :

Prérequis

Pour pouvoir recevoir des événements Amazon S3 dans EventBridge, vous devez activer EventBridge dans la console Amazon S3. Ce didacticiel part du principe qu'EventBridge est activé. Pour plus d'informations, consultez Activation d'Amazon EventBridge dans la console S3.

Étape 1 : Créer une rubrique Amazon SNS

Créez la rubrique qui doit recevoir les événements en provenance d'EventBridge.

Pour créer une rubrique

1. Ouvrez la console Amazon SNS à partir de l'adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le panneau de navigation, sélectionnez Topics (Rubriques).
3. Choisissez Create topic (Créer une rubrique).
4. Pour Type, choisissez Standard.
5. Entrez **`eventbridge-test`** comme nom de la rubrique.
6. Choisissez Create topic (Créer une rubrique).

Étape 2 : Créer un abonnement Amazon SNS

Créez un abonnement pour obtenir des notifications par e-mail d'Amazon S3 lorsque la rubrique reçoit des événements.

Pour créer un abonnement

1. Ouvrez la console Amazon SNS à partir de l'adresse https://console.aws.amazon.com/sns/v3/home.
2. Dans le panneau de navigation, sélectionnez Abonnements.
3. Sélectionnez Créer un abonnement.
4. Pour ARN de la rubrique, choisissez la rubrique que vous avez créée à l'étape 1. Pour ce didacticiel, choisissez eventbridge-test.
5. Pour Protocole, choisissez E-mail.
6. Saisissez votre adresse e-mail dans Endpoint (Point de terminaison).
7. Choisissez Create subscription (Créer un abonnement).
8. Confirmez l'abonnement en choisissant Confirmer l'abonnement dans l'e-mail que vous recevez du service de notifications AWS.

Étape 3 : Créer une règle

Créez une règle de sorte que des événements soient envoyés à votre rubrique lorsqu'un objet Amazon S3 est créé.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle. Par exemple, nommez la règle `s3-test`
5. Pour Bus d'événements, sélectionnez Par défaut.
6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).

8. Pour Event source (Origine de l'événement), choisissez events or EventBridge partner events (Événements AWS ou événements partenaires EventBridge).
9. Pour Méthode de création, choisissez Utiliser le formulaire d'événement.
10. Pour Event pattern (Modèle d'événement), procédez comme suit :
    a. Pour Source d'événement, sélectionnez Services AWS dans la liste déroulante.
    b. Pour Service AWS, sélectionnez Simple Storage Service (S3) dans la liste déroulante.
    c. Pour Type d'événement, choisissez Notification d'événement Amazon S3 dans la liste déroulante.
    d. Choisissez Événement(s) spécifique(s), puis sélectionnez Objets créés dans la liste déroulante.
    e. Choisissez Tout compartiment.
11. Choisissez Next (Suivant).
12. Pour Types de cibles, choisissez service AWS.
13. Pour Sélectionner une cible, choisissez Rubrique SNS dans la liste déroulante.
14. Pour Rubrique, sélectionnez la rubrique Amazon SNS que vous avez créée dans la section Étape 1 : Créer une rubrique SNS. Dans cet exemple, sélectionnez `eventbridge-test`.
15. Choisissez Next (Suivant).
16. Choisissez Next (Suivant).
17. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 4 : Tester la règle

Pour tester votre règle, créez un objet Amazon S3 en chargeant un fichier sur un compartiment compatible EventBridge. Ensuite, attendez quelques minutes et vérifiez si vous avez reçu un e-mail du service de notifications AWS.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la rubrique SNS

1. Ouvrez la page Rubriques de la console SNS.

2. Sélectionnez la rubrique que vous avez créée.
3. Choisissez Supprimer.
4. Saisissez **`delete me`**.
5. Choisissez Supprimer.

Pour supprimer l'abonnement SNS

1. Ouvrez la page Abonnements de la console SNS.
2. Sélectionnez l'abonnement que vous avez créé.
3. Choisissez Supprimer.
4. Choisissez Supprimer.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Sélectionnez Delete.

Didacticiel : Planification de fonctions AWS Lambda à l'aide d'EventBridge

Vous pouvez configurer une règle de sorte qu'elle exécute une fonction AWS Lambda selon une planification. Ce didacticiel explique comment utiliser AWS Management Console ou l'AWS CLI pour créer la règle. Si vous souhaitez utiliser l'interface AWS CLI mais que vous ne l'avez pas installée, consultez Installation, mise à jour et désinstallation de l'interface AWS CLI version 2.

Pour les planifications, EventBridge n'offre pas de précision de deuxième niveau dans les expressions de planification. Le niveau de résolution maximal lors de l'utilisation d'une expression cron est d'une minute. Compte tenu de la nature distribuée d'EventBridge et des services cibles, un décalage de plusieurs secondes peut être observé entre le moment où la règle planifiée est déclenchée et le moment où le service cible exécute la ressource cible.

Étapes :

- Étape 1 : Créer une fonction Lambda
- Étape 2 : Création d'une règle
- Étape 3 : Vérifier la règle
- Étape 4 : Confirmer la bonne exécution
- Étape 5 : Nettoyer vos ressources

Étape 1 : Créer une fonction Lambda

Créez une fonction Lambda pour enregistrer les événements planifiés.

Pour créer une fonction Lambda

1. Ouvrez la console AWS Lambda à l'adresse https://console.aws.amazon.com/lambda/.
2. Sélectionnez Create function (Créer une fonction).
3. Choisissez Créer à partir de zéro.
4. Saisissez un nom et une description pour la fonction Lambda. Par exemple, nommez la fonction `LogScheduledEvent`.
5. Gardez les autres options comme valeurs par défaut et choisissez Créer une fonction.
6. Dans l'onglet Code de la page de fonction, double-cliquez sur index.js.
7. Remplacez le code existant par le code suivant.

```
'use strict';
```

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

8. Choisissez Deploy (Déployer).

Étape 2 : Création d'une règle

Créez une règle pour exécuter la fonction Lambda que vous avez créée à l'étape 1 selon une planification.

Vous pouvez utiliser la console ou l'interface AWS CLI pour créer la règle. Avant d'utiliser l'interface AWS CLI, vous devez d'abord accorder à la règle l'autorisation d'invoquer votre fonction Lambda. Vous pouvez ensuite créer la règle et ajouter la fonction Lambda comme cible.

Pour créer une règle (console)

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle.

   Une règle ne peut pas avoir le même nom qu'une autre règle de la même région et sur le même bus d'événement.
5. Pour Event bus (Bus d'événement), sélectionnez le bus d'événement que vous souhaitez associer à cette règle. Si vous souhaitez que cette règle corresponde aux événements provenant de votre compte, sélectionnez default event bus (Bus d'événement AWS par défaut). Lorsqu'un service AWS de votre compte émet un événement, il accède toujours au bus d'événement par défaut de votre compte.
6. Pour Rule type (Type de règle), choisissez Schedule (Planifier).
7. Choisissez Next (Suivant).
8. Pour Modèle de planification, choisissez Un programme qui s'exécute à fréquence régulière, par exemple toutes les 10 minutes, entrez **5**, puis choisissez Minutes dans la liste déroulante.
9. Choisissez Next (Suivant).

10. Pour Types de cibles, choisissez service AWS.
11. Pour Sélectionner une cible, choisissez Fonction Lambda dans la liste déroulante.
12. Pour Fonction, sélectionnez la fonction Lambda que vous avez créée dans la section Étape 1 : Créer une fonction Lambda. Dans cet exemple, sélectionnez `LogScheduledEvent`.
13. Choisissez Next (Suivant).
14. Choisissez Next (Suivant).
15. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Pour créer une règle (AWS CLI)

1. Pour créer une règle qui s'exécute selon une planification, utilisez la commande `put-rule`.

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

Lorsque cette règle s'exécute, elle crée un événement et l'envoie aux cibles. 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. Pour accorder au principal du service EventBridge (`events.amazonaws.com`) l'autorisation d'exécuter la règle, utilisez la commande `add-permission`.

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

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

4. Pour ajouter la fonction Lambda que vous avez créée à l'étape 1 à la règle, utilisez la commande `put-targets`.

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

Étape 3 : Vérifier la règle

Après avoir terminé l'étape 2, attendez au moins cinq minutes avant de vérifier que la fonction Lambda a bien été invoquée.

Affichage de la sortie de la fonction Lambda

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le panneau de navigation, sélectionnez Logs (Journaux).
3. Sélectionnez le nom du groupe de journaux pour votre 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.

Étape 4 : Confirmer la bonne exécution

Si l'événement Lambda se trouve dans les journaux CloudWatch, cela signifie que vous avez correctement effectué ce didacticiel. Si l'événement ne figure pas dans vos journaux CloudWatch, essayez de résoudre le problème en vérifiant d'abord que la règle a bien été créée. Si celle-ci semble correcte, vérifiez que le code de votre fonction Lambda l'est également.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Choisissez Supprimer.

Pour supprimer la ou les fonctions Lambda

1. Ouvrez la page Functions (Fonctions) de la console Lambda.
2. Sélectionnez la ou les fonctions que vous avez créées.
3. Sélectionnez Actions, Delete (Supprimer).
4. Sélectionnez Delete.

Didacticiels Amazon EventBridge pour une intégration avec des fournisseurs SaaS

EventBridge peut fonctionner directement avec des applications et services partenaires SaaS pour l'envoi et la réception d'événements. Les didacticiels suivants vous montrent comment intégrer EventBridge avec les partenaires SaaS.

Didacticiels:

- Didacticiel : Création d'une connexion à Datadog en tant que destination d'API
- Didacticiel : Création d’une connexion à Salesforce en tant que destination d’API
- Didacticiel : Création d'une connexion à Zendesk en tant que destination d'API

Didacticiel : Création d'une connexion à Datadog en tant que destination d'API

Vous pouvez utiliser EventBridge pour router des événements vers des services tiers, tels que Datadog.

Dans ce didacticiel, vous allez utiliser la console EventBridge pour créer successivement une connexion à Datadog, une destination d'API qui pointe vers Datadog, puis une règle visant à router les événements vers Datadog.

Étapes :

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des ressources suivantes :

- Un compte Datadog.
- Une clé d'API Datadog.
- Un compartiment Amazon Simple Storage Service (Amazon S3) pour lequel EventBridge est activé.

Étape 1 : Créer une connexion

Pour envoyer des événements à Datadog, vous devez d'abord établir une connexion à l'API Datadog.

Pour créer la connexion

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le panneau de navigation, choisissez Destinations d'API.
3. Choisissez l'onglet Connexions, puis sélectionnez Créer une connexion.

4. Nommez et décrivez la connexion. Par exemple, entrez **`Datadog`** pour le nom et **`Datadog API Connection`** pour la description.
5. Pour Type d'autorisation, choisissez Clé API.
6. Pour Nom de clé de l'API, entrez **`DD-API-KEY`**.
7. Pour Valeur, collez votre clé d'API secrète Datadog.
8. Choisissez Créer.

Étape 2 : Créer la destination d'API

Maintenant que vous avez créé la connexion, vous allez créer la destination d'API à utiliser comme cible de la règle.

Pour créer la destination d'API

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le panneau de navigation, choisissez Destinations d'API.
3. Choisissez Créer une destination d'API.
4. Nommez et décrivez la destination d'API. Par exemple, entrez **`DatadogAD`** pour le nom et **`Datadog API Destination`** pour la description.
5. Pour Point de terminaison de la destination d'API, entrez **`https://http-intake.logs.datadoghq.com/api/v2/logs`**.
6. Dans le champ HTTP Method, sélectionnez POST.
7. Pour Limite du taux d'appel, entrez **`300`**.
8. Pour Connexion, choisissez Utiliser une connexion existante et sélectionnez la connexion `Datadog` que vous avez créée à l'étape 1.
9. Choisissez Créer.

Étape 3 : Créer une règle

Vous allez maintenant créer une règle de sorte que des événements soient envoyés à Datadog lorsqu'un objet Amazon S3 est créé.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.

2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle. Par exemple, entrez **`DatadogRule`** pour le nom et **`Rule to send events to Datadog for S3 object creation`** pour la description.
5. Pour Event bus (Bus d'événement), choisissez default (défaut).
6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. Pour Modèle d'événement, entrez ce qui suit :

```
{
  "source": ["aws.s3"]
}
```

10. Choisissez Next (Suivant).
11. Pour Types de cibles, choisissez Destination d'API EventBridge.
12. Pour Destination d'API, choisissez Utiliser une destination d'API existante, puis choisissez la destination `DatadogAD` que vous avez créée à l'étape 2.
13. Pour Rôle d'exécution, choisissez Créer un rôle pour cette ressource spécifique.
14. Pour Réglages supplémentaires, procédez comme suit :

    a. Pour Configurer l'entrée cible, choisissez Transformateur d'entrée dans la liste déroulante.

    b. Choisissez Configurer le transformateur d'entrée.

    c. Pour Exemples d'événements, entrez ce qui suit :

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

    d. Pour Transformateur d'entrée cible, procédez comme suit :

        i. Pour Chemin d'entrée, entrez ce qui suit :

```
{"detail":"$.detail"}
```

ii. Pour Modèle d'entrée, entrez ce qui suit :

```
{"message": <detail>}
```

e. Choisissez Confirmer.

15. Choisissez Next (Suivant).
16. Choisissez Next (Suivant).
17. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 4 : Tester la règle

Pour tester votre règle, créez un objet Amazon S3 en chargeant un fichier sur un compartiment compatible EventBridge. L'objet créé sera journalisé dans la console Datadog Logs.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les connexions EventBridge

1. Ouvrez la page Destination d'API de la console EventBridge.
2. Choisissez l'onglet Connections (Connexions).
3. Sélectionnez la ou les connexions que vous avez créées.
4. Choisissez Supprimer.
5. Entrez le nom de la connexion et choisissez Supprimer.

Pour supprimer la ou les destinations d'API EventBridge

1. Ouvrez la page Destination d'API de la console EventBridge.
2. Sélectionnez la ou les destinations d'API que vous avez créées.
3. Choisissez Supprimer.
4. Entrez le nom de la destination d'API et choisissez Supprimer.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Sélectionnez Delete.

Didacticiel : Création d’une connexion à Salesforce en tant que destination d’API

Vous pouvez l'utiliser EventBridge pour acheminer des événements vers des services tiers, tels que Salesforce.

Dans ce didacticiel, vous allez utiliser la EventBridge console pour créer une connexionSalesforce, une destination d'API pointant vers Salesforce et une règle vers laquelle acheminer les événementsSalesforce.

Étapes :

- Prérequis
- Étape 1 : Créer une connexion
- Étape 2 : Créer la destination d’API
- Étape 3 : Créer une règle
- Étape 4 : Tester la règle
- Étape 5 : Nettoyer vos ressources

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des ressources suivantes :

- Un compte Salesforce.
- Une application connectée Salesforce.
- Un jeton de sécurité Salesforce.
- Un événement de plateforme personnalisé Salesforce.
- Un EventBridge bucket Amazon Simple Storage Service (Amazon S3) activé.

Étape 1 : Créer une connexion

Pour envoyer des événements à Salesforce, vous devez d’abord établir une connexion à l’API Salesforce.

Pour créer la connexion

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.

2. Dans le panneau de navigation, choisissez Destinations d'API.
3. Choisissez l'onglet Connexions, puis sélectionnez Créer une connexion.
4. Nommez et décrivez la connexion. Par exemple, entrez **`Salesforce`** pour le nom et **`Salesforce API Connection`** pour la description.
5. Pour Type de destination, choisissez Partenaires, et pour Destinations partenaires, sélectionnez Salesforce dans la liste déroulante.
6. Pour Point de terminaison d'autorisation, entrez l'une des adresses suivantes :

   - Si vous utilisez une organisation de production, entrez **`https://MyDomainName.my.salesforce.com./services/oauth2/token`**
   - Si vous utilisez un environnement de test (sandbox) sans domaines améliorés, entrez **`https://MyDomainName--SandboxName.my. salesforce.com/services /oauth2/token`**
   - Si vous utilisez un environnement de test (sandbox) avec domaines améliorés, entrez **`https://MyDomainName-- SandboxName.sandbox.my.salesforce.com/services/oauth2/token`**
7. Pour Méthode HTTP, choisissez POST dans la liste déroulante.
8. Pour ID client, entrez l'ID client de votre application connectée Salesforce.
9. Pour Clé secrète du client, entrez la clé secrète du client de votre application connectée Salesforce.
10. Pour les paramètres HTTP OAuth, entrez la paire clé/valeur suivante :

Clé	Valeur
grant_type	client_credentials

11. Choisissez Créer.

Étape 2 : Créer la destination d'API

Maintenant que vous avez créé la connexion, vous allez créer la destination d'API à utiliser comme cible de la règle.

Pour créer la destination d'API

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.

2. Dans le panneau de navigation, choisissez Destinations d'API.
3. Choisissez Créer une destination d'API.
4. Nommez et décrivez la destination d'API. Par exemple, entrez **`SalesforceAD`** pour le nom et **`Salesforce API Destination`** pour la description.
5. Pour Point de terminaison de la destination d'API, entrez **`https://`*`MyDomainName`*`.my.salesforce.com/services/data/v54.0/sobjects/`*`MyEvent__e`***, Myevent__e correspondant à l'événement de plateforme auquel vous souhaitez envoyer les informations.
6. Pour Méthode HTTP, choisissez POST dans la liste déroulante.
7. Pour Limite du taux d'appel, entrez **`300`**.
8. Pour Connexion, choisissez Utiliser une connexion existante et sélectionnez la connexion `Salesforce` que vous avez créée à l'étape 1.
9. Choisissez Créer.

Étape 3 : Créer une règle

Vous allez maintenant créer une règle de sorte que des événements soient envoyés à Salesforce lorsqu'un objet Amazon S3 est créé.

Pour créer une règle

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Règles.
3. Choisissez Créer une règle.
4. Saisissez un nom et une description pour la règle. Par exemple, entrez **`SalesforceRule`** pour le nom et **`Rule to send events to Salesforce for S3 object creation`** pour la description.
5. Pour Event bus (Bus d'événement), choisissez default (défaut).
6. Pour Type de règle, choisissez Règle avec un modèle d'événement.
7. Choisissez Suivant.
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. Pour Modèle d'événement, entrez ce qui suit :

```
{
```

```
  "source": ["aws.s3"]
}
```

10. Choisissez Suivant.
11. Pour les types de cibles, choisissez la destination de EventBridge l'API.
12. Pour Destination d’API, choisissez Utiliser une destination d’API existante, puis choisissez la destination `SalesforceAD` que vous avez créée à l’étape 2.
13. Pour Rôle d’exécution, choisissez Créer un rôle pour cette ressource spécifique.
14. Pour Réglages supplémentaires, procédez comme suit :

    a. Pour Configurer l’entrée cible, choisissez Transformateur d’entrée dans la liste déroulante.

    b. Choisissez Configurer le transformateur d’entrée.

    c. Pour Exemples d’événements, entrez ce qui suit :

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

    d. Pour Transformateur d’entrée cible, procédez comme suit :

        i. Pour Chemin d’entrée, entrez ce qui suit :

        ```
        {"detail":"$.detail"}
        ```

        ii. Pour Modèle d’entrée, entrez ce qui suit :

        ```
        {"message": <detail>}
        ```

    e. Choisissez Confirmer.
15. Choisissez Suivant.
16. Choisissez Suivant.
17. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 4 : Tester la règle

Pour tester votre règle, créez un objet Amazon S3 en téléchargeant un fichier dans un compartiment EventBridge activé. Les informations relatives à l’objet créé seront envoyées à l’événement de la plateforme Salesforce.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. En supprimant AWS les ressources que vous n'utilisez plus, vous évitez des frais inutiles sur votre AWS compte.

Pour supprimer les EventBridge connexions

1. Ouvrez la page de destination de l'API de la EventBridge console.
2. Choisissez l'onglet Connections (Connexions).
3. Sélectionnez la ou les connexions que vous avez créées.
4. Sélectionnez Delete (Supprimer).
5. Entrez le nom de la connexion et choisissez Supprimer.

Pour supprimer la ou les destinations de l' EventBridge API

1. Ouvrez la page de destination de l'API de la EventBridge console.
2. Sélectionnez la ou les destinations d’API que vous avez créées.
3. Sélectionnez Delete (Supprimer).
4. Entrez le nom de la destination d’API et choisissez Supprimer.

Pour supprimer la ou les EventBridge règles

1. Ouvrez la page Règles de la EventBridge console.
2. Sélectionnez la ou les règles que vous avez créées.
3. Sélectionnez Delete.
4. Sélectionnez Supprimer.

Didacticiel : Création d'une connexion à Zendesk en tant que destination d'API

Vous pouvez utiliser EventBridge pour router les événements vers des services tiers tels que Zendesk.

Dans ce didacticiel, vous allez utiliser la console EventBridge pour créer successivement une connexion à Zendesk, une destination d'API qui pointe vers Zendesk, puis une règle visant à router les événements vers Zendesk.

Étapes :

- Prérequis
- Étape 1 : Créer une connexion
- Étape 2 : Créer la destination d'API
- Étape 3 : Créer une règle
- Étape 4 : Tester la règle
- Étape 5 : Nettoyer vos ressources

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des ressources suivantes :

- Un compte Zendesk.
- Un compartiment Amazon Simple Storage Service (Amazon S3) pour lequel EventBridge est activé.

Étape 1 : Créer une connexion

Pour envoyer des événements à Zendesk, vous devez d'abord établir une connexion à l'API Zendesk.

Pour créer la connexion

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le panneau de navigation, choisissez Destinations d'API.
3. Choisissez l'onglet Connexions, puis sélectionnez Créer une connexion.

4. Nommez et décrivez la connexion. Par exemple, entrez **`Zendesk`** pour le nom et **`Connection to Zendesk API`** pour la description.
5. Pour Type d'autorisation, choisissez Basic (nom d'utilisateur/mot de passe).
6. Pour Nom d'utilisateur, entrez votre nom d'utilisateur Zendesk.
7. Pour Mot de passe, entrez votre mot de passe Zendesk.
8. Choisissez Créer.

Étape 2 : Créer la destination d'API

Maintenant que vous avez créé la connexion, vous allez créer la destination d'API à utiliser comme cible de la règle.

Pour créer la destination d'API

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le panneau de navigation, choisissez Destinations d'API.
3. Choisissez Créer une destination d'API.
4. Nommez et décrivez la destination d'API. Par exemple, entrez **`ZendeskAD`** pour le nom et **`Zendesk API destination`** pour la description.
5. Pour Point de terminaison de la destination d'API, entrez **`https://`*`your-subdomain`*`.zendesk.com/api/v2/tickets.json`**, *`your-subdomain`* étant le sous-domaine associé à votre compte Zendesk.
6. Dans le champ HTTP Method, sélectionnez POST.
7. Pour Limite du taux d'appel, entrez **`10`**.
8. Pour Connexion, choisissez Utiliser une connexion existante et sélectionnez la connexion `Zendesk` que vous avez créée à l'étape 1.
9. Choisissez Créer.

Étape 3 : Créer une règle

À présent, créez une règle de façon à envoyer des événements à Zendesk lorsqu'un objet Amazon S3 est créé.

Pour créer une règle

1. Ouvrez la console Amazon EventBridge à l'adresse https://console.aws.amazon.com/events/.
2. Dans le volet de navigation, choisissez Rules.
3. Choisissez Create rule.
4. Saisissez un nom et une description pour la règle. Par exemple, entrez **`ZendeskRule`** pour le nom et **`Rule to send events to Zendesk when S3 objects are created`** pour la description.
5. Pour Event bus (Bus d'événement), choisissez default (défaut).
6. Pour Rule type (Type de règle), choisissez Rule with an event pattern (Règle avec un modèle d'événement).
7. Choisissez Next (Suivant).
8. Pour Event source (Source de l'événement), choisissez Other (Autres).
9. Pour Modèle d'événement, entrez ce qui suit :

```
{
  "source": ["aws.s3"]
}
```

10. Choisissez Next (Suivant).
11. Pour Types de cibles, choisissez Destination d'API EventBridge.
12. Pour Destination d'API, choisissez Utiliser une destination d'API existante, puis choisissez la destination `ZendeskAD` que vous avez créée à l'étape 2.
13. Pour Rôle d'exécution, choisissez Créer un rôle pour cette ressource spécifique.
14. Pour Réglages supplémentaires, procédez comme suit :

    a. Pour Configurer l'entrée cible, choisissez Transformateur d'entrée dans la liste déroulante.

    b. Choisissez Configurer le transformateur d'entrée.

    c. Pour Exemples d'événements, entrez ce qui suit :

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

    d. Pour Transformateur d'entrée cible, procédez comme suit :

i. Pour Chemin d'entrée, entrez ce qui suit :

```
{"detail":"$.detail"}
```

ii. Pour Modèle d'entrée, entrez ce qui suit :

```
{"message": <detail>}
```

e. Choisissez Confirmer.

15. Choisissez Next (Suivant).
16. Choisissez Next (Suivant).
17. Consultez les détails de la règle et choisissez Create rule (Créer une règle).

Étape 4 : Tester la règle

Pour tester votre règle, créez un objet Amazon S3 en chargeant un fichier sur un compartiment compatible EventBridge. Lorsque l'événement correspond à la règle, EventBridge appelle l'API Zendesk Create Ticket. Le nouveau ticket s'affiche dans le tableau de bord Zendesk.

Étape 5 : Nettoyer vos ressources

Vous pouvez maintenant supprimer les ressources que vous avez créées pour ce didacticiel, sauf si vous souhaitez les conserver. La suppression des ressources AWS que vous n'utilisez plus vous évite des frais inutiles sur votre compte AWS.

Pour supprimer la ou les connexions EventBridge

1. Ouvrez la page Destination d'API de la console EventBridge.
2. Choisissez l'onglet Connections (Connexions).
3. Sélectionnez la ou les connexions que vous avez créées.
4. Choisissez Supprimer.
5. Entrez le nom de la connexion et choisissez Supprimer.

Pour supprimer la ou les destinations d'API EventBridge

1. Ouvrez la page Destination d'API de la console EventBridge.

2. Sélectionnez la ou les destinations d'API que vous avez créées.
3. Choisissez Supprimer.
4. Entrez le nom de la destination d'API et choisissez Supprimer.

Pour supprimer la ou les règles EventBridge

1. Ouvrez la page Règles de la console EventBridge.
2. Sélectionnez la ou les règles que vous avez créées.
3. Choisissez Supprimer.
4. Sélectionnez Delete.

Utilisation EventBridge avec un AWS SDK

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

Documentation SDK	Exemples de code
AWS SDK for C++	AWS SDK for C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK for Go	AWS SDK for Go exemples de code
AWS SDK for Java	AWS SDK for Java exemples de code
AWS SDK for JavaScript	AWS SDK for JavaScript exemples de code
Kit AWS SDK pour Kotlin	Kit AWS SDK pour Kotlin exemples de code
AWS SDK for .NET	AWS SDK for .NET exemples de code
AWS SDK for PHP	AWS SDK for PHP exemples de code
AWS Tools for PowerShell	Outils pour des exemples PowerShell de code
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) exemples de code
AWS SDK for Ruby	AWS SDK for Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Pour des exemples spécifiques à EventBridge, voirExemples de code pour EventBridge l'utilisation des AWS SDK.

Exemple de disponibilité

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

Exemples de code pour EventBridge l'utilisation des AWS SDK

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

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés et dans des exemples interservices.

Les Scénarios sont des exemples de code qui vous montrent comment accomplir une tâche spécifique en appelant plusieurs fonctions au sein d'un même service.

Les Exemples de services croisés sont des exemples d'applications fonctionnant sur plusieurs Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Mise en route

Bonjour EventBridge

Les exemples de code suivants montrent comment commencer à utiliser EventBridge.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
using Amazon.EventBridge;
using Amazon.EventBridge.Model;

namespace EventBridgeActions;

public static class HelloEventBridge
{
    static async Task Main(string[] args)
    {
        var eventBridgeClient = new AmazonEventBridgeClient();

        Console.WriteLine($"Hello Amazon EventBridge! Following are some of your
 EventBuses:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first five event buses.
        var response = await eventBridgeClient.ListEventBusesAsync(
            new ListEventBusesRequest()
            {
                Limit = 5
            });

        foreach (var eventBus in response.EventBuses)
        {
            Console.WriteLine($"\tEventBus: {eventBus.Name}");
            Console.WriteLine($"\tArn: {eventBus.Arn}");
            Console.WriteLine($"\tPolicy: {eventBus.Policy}");
            Console.WriteLine();
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListEventBusesà la section Référence des AWS SDK for .NET API.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 */
public class HelloEventBridge {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        EventBridgeClient eventBrClient = EventBridgeClient.builder()
                .region(region)
                .build();

        listBuses(eventBrClient);
        eventBrClient.close();
    }

    public static void listBuses(EventBridgeClient eventBrClient) {
        try {
            ListEventBusesRequest busesRequest = ListEventBusesRequest.builder()
                    .limit(10)
                    .build();

            ListEventBusesResponse response =
 eventBrClient.listEventBuses(busesRequest);
            List<EventBus> buses = response.eventBuses();
            for (EventBus bus : buses) {
```

```
                System.out.println("The name of the event bus is: " +
 bus.name());
                System.out.println("The ARN of the event bus is: " + bus.arn());
            }

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

- Pour plus de détails sur l'API, reportez-vous ListEventBusesà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
import aws.sdk.kotlin.services.eventbridge.EventBridgeClient
import aws.sdk.kotlin.services.eventbridge.model.ListEventBusesRequest
import aws.sdk.kotlin.services.eventbridge.model.ListEventBusesResponse

suspend fun main() {
    listBusesHello()
}

suspend fun listBusesHello() {
    val request = ListEventBusesRequest {
        limit = 10
    }

    EventBridgeClient { region = "us-west-2" }.use { eventBrClient ->
```

```
        val response: ListEventBusesResponse =
 eventBrClient.listEventBuses(request)
        response.eventBuses?.forEach { bus ->
            println("The name of the event bus is ${bus.name}")
            println("The ARN of the event bus is ${bus.arn}")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListEventBusesà la section AWS SDK pour la référence de l'API Kotlin.

Exemples de code

- Actions relatives à EventBridge l'utilisation des AWS SDK
  - Utilisation DeleteRule avec un AWS SDK ou une CLI
  - Utilisation DescribeRule avec un AWS SDK ou une CLI
  - Utilisation DisableRule avec un AWS SDK ou une CLI
  - Utilisation EnableRule avec un AWS SDK ou une CLI
  - Utilisation ListRuleNamesByTarget avec un AWS SDK ou une CLI
  - Utilisation ListRules avec un AWS SDK ou une CLI
  - Utilisation ListTargetsByRule avec un AWS SDK ou une CLI
  - Utilisation PutEvents avec un AWS SDK ou une CLI
  - Utilisation PutRule avec un AWS SDK ou une CLI
  - Utilisation PutTargets avec un AWS SDK ou une CLI
  - Utilisation RemoveTargets avec un AWS SDK ou une CLI
- Scénarios d' EventBridge utilisation des AWS SDK
  - Créez et déclenchez une règle dans Amazon à EventBridge l'aide d'un AWS SDK
  - Commencez avec EventBridge les règles et les cibles à l'aide d'un AWS SDK
- Exemples multiservices d' EventBridge utilisation des SDK AWS
  - Utilisent des événements planifiés pour appeler une fonction Lambda

Actions relatives à EventBridge l'utilisation des AWS SDK

Les exemples de code suivants montrent comment effectuer des EventBridge actions individuelles avec AWS les SDK. Ces extraits appellent l' EventBridge API et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

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

Exemples

- Utilisation DeleteRule avec un AWS SDK ou une CLI
- Utilisation DescribeRule avec un AWS SDK ou une CLI
- Utilisation DisableRule avec un AWS SDK ou une CLI
- Utilisation EnableRule avec un AWS SDK ou une CLI
- Utilisation ListRuleNamesByTarget avec un AWS SDK ou une CLI
- Utilisation ListRules avec un AWS SDK ou une CLI
- Utilisation ListTargetsByRule avec un AWS SDK ou une CLI
- Utilisation PutEvents avec un AWS SDK ou une CLI
- Utilisation PutRule avec un AWS SDK ou une CLI
- Utilisation PutTargets avec un AWS SDK ou une CLI
- Utilisation RemoveTargets avec un AWS SDK ou une CLI

Utilisation **`DeleteRule`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Supprimez une règle par son nom.

```
    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteRuleByName(string ruleName)
    {
        var response = await _amazonEventBridge.DeleteRuleAsync(
            new DeleteRuleRequest()
            {
                Name = ruleName
            });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous DeleteRuleà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour supprimer une règle d' CloudWatch événements

Cet exemple supprime la règle nommée InstanceStateChanges EC2 :

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- Pour plus de détails sur l'API, reportez-vous DeleteRuleà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void deleteRuleByName(EventBridgeClient eventBrClient, String
ruleName) {
       DeleteRuleRequest ruleRequest = DeleteRuleRequest.builder()
               .name(ruleName)
               .build();

       eventBrClient.deleteRule(ruleRequest);
       System.out.println("Successfully deleted the rule");
   }
```

- Pour plus de détails sur l'API, reportez-vous DeleteRuleà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun deleteRuleByName(ruleName: String?) {
    val ruleRequest = DeleteRuleRequest {
```

```
        name = ruleName
    }
    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.deleteRule(ruleRequest)
        println("Successfully deleted the rule")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DeleteRuleà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DescribeRule`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Obtenez l'état d'une règle à l'aide de sa description.

```
    /// <summary>
```

```
    /// Get the state for a rule by the rule name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="eventBusName">The optional name of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The state of the rule.</returns>
    public async Task<RuleState> GetRuleStateByRuleName(string ruleName, string?
eventBusName = null)
    {
        var ruleResponse = await _amazonEventBridge.DescribeRuleAsync(
            new DescribeRuleRequest()
            {
                Name = ruleName,
                EventBusName = eventBusName
            });
        return ruleResponse.State;
    }
```

- Pour plus de détails sur l'API, reportez-vous DescribeRuleà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour afficher les informations relatives à une règle d' CloudWatch événements

Cet exemple affiche des informations sur la règle nommée DailyLambdaFunction :

```
aws events describe-rule --name "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous DescribeRuleà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void checkRule(EventBridgeClient eventBrClient, String
eventRuleName) {
       try {
           DescribeRuleRequest ruleRequest = DescribeRuleRequest.builder()
                   .name(eventRuleName)
                   .build();

           DescribeRuleResponse response =
eventBrClient.describeRule(ruleRequest);
           System.out.println("The state of the rule is " +
response.stateAsString());

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

- Pour plus de détails sur l'API, reportez-vous DescribeRuleà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun checkRule(eventRuleName: String?) {
    val ruleRequest = DescribeRuleRequest {
        name = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.describeRule(ruleRequest)
        println("The state of the rule is $response")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DescribeRuleà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`DisableRule`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Désactivez une règle par son nom.

```
    /// <summary>
    /// Disable a particular rule on an event bus.
    /// </summary
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableRuleByName(string ruleName)
    {
        var ruleResponse = await _amazonEventBridge.DisableRuleAsync(
            new DisableRuleRequest()
            {
                Name = ruleName
            });
        return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous DisableRuleà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour désactiver une règle d' CloudWatch événements

Cet exemple désactive la règle nommée DailyLambdaFunction. La règle n’est pas supprimée :

```
aws events disable-rule --name "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous DisableRuleà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Désactivez une règle à l'aide de son nom.

```
   public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
       try {
           if (!isEnabled) {
               System.out.println("Disabling the rule: " + eventRuleName);
               DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();

               eventBrClient.disableRule(ruleRequest);
           } else {
               System.out.println("Enabling the rule: " + eventRuleName);
               EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();
               eventBrClient.enableRule(ruleRequest);
           }

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

- Pour plus de détails sur l'API, reportez-vous DisableRuleà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
```

```
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous DisableRuleà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`EnableRule`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Activez une règle par son nom.

```
/// <summary>
/// Enable a particular rule on an event bus.
/// </summary>
/// <param name="ruleName">The name of the rule.</param>
/// <returns>True if successful.</returns>
public async Task<bool> EnableRuleByName(string ruleName)
{
    var ruleResponse = await _amazonEventBridge.EnableRuleAsync(
        new EnableRuleRequest()
        {
            Name = ruleName
        });
    return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
}
```

- Pour plus de détails sur l'API, reportez-vous EnableRuleà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour activer une règle d' CloudWatch événements

Cet exemple active la règle nommée DailyLambdaFunction, qui avait été précédemment désactivée :

```
aws events enable-rule --name "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous EnableRuleà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Activez une règle à l'aide de son nom.

```
   public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
       try {
           if (!isEnabled) {
               System.out.println("Disabling the rule: " + eventRuleName);
               DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();

               eventBrClient.disableRule(ruleRequest);
           } else {
               System.out.println("Enabling the rule: " + eventRuleName);
               EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                       .name(eventRuleName)
                       .build();
               eventBrClient.enableRule(ruleRequest);
           }

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

- Pour plus de détails sur l'API, reportez-vous EnableRuleà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous EnableRuleà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListRuleNamesByTarget`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Répertoriez tous les noms de règle à l'aide de la cible.

```
    /// <summary>
    /// List names of all rules matching a target.
    /// </summary>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <returns>The list of rule names.</returns>
    public async Task<List<string>> ListAllRuleNamesByTarget(string targetArn)
    {
        var results = new List<string>();
        var request = new ListRuleNamesByTargetRequest()
        {
            TargetArn = targetArn
        };
        ListRuleNamesByTargetResponse response;
        do
        {
            response = await
_amazonEventBridge.ListRuleNamesByTargetAsync(request);
            results.AddRange(response.RuleNames);
            request.NextToken = response.NextToken;
```

```
        } while (response.NextToken is not null);

        return results;
    }
```

- Pour plus de détails sur l'API, reportez-vous ListRuleNamesByTargetà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour afficher toutes les règles ayant une cible spécifiée

Cet exemple affiche toutes les règles dont la cible est la fonction Lambda nommée MyFunctionName « » :

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-
east-1:123456789012:function:MyFunctionName"
```

- Pour plus de détails sur l'API, reportez-vous ListRuleNamesByTargetà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Répertoriez tous les noms de règle à l’aide de la cible.

```
   public static void listTargetRules(EventBridgeClient eventBrClient, String
topicArn) {
```

```
        ListRuleNamesByTargetRequest ruleNamesByTargetRequest =
 ListRuleNamesByTargetRequest.builder()
                .targetArn(topicArn)
                .build();

        ListRuleNamesByTargetResponse response =
 eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest);
        List<String> rules = response.ruleNames();
        for (String rule : rules) {
            System.out.println("The rule name is " + rule);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous ListRuleNamesByTargetà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun listTargetRules(topicArnVal: String?) {
    val ruleNamesByTargetRequest = ListRuleNamesByTargetRequest {
        targetArn = topicArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response =
 eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest)
        response.ruleNames?.forEach { rule ->
            println("The rule name is $rule")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListRuleNamesByTargetà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListRules`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Répertoriez toutes les règles pour un bus d'événements.

```
   /// <summary>
   /// List the rules on an event bus.
   /// </summary>
   /// <param name="eventBusArn">The optional ARN of the event bus. If empty,
uses the default event bus.</param>
   /// <returns>The list of rules.</returns>
   public async Task<List<Rule>> ListAllRulesForEventBus(string? eventBusArn =
null)
   {
       var results = new List<Rule>();
```

```
        var request = new ListRulesRequest()
        {
            EventBusName = eventBusArn
        };
        // Get all of the pages of rules.
        ListRulesResponse response;
        do
        {
            response = await _amazonEventBridge.ListRulesAsync(request);
            results.AddRange(response.Rules);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }
```

- Pour plus de détails sur l'API, reportez-vous ListRulesà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour afficher une liste de toutes les règles relatives aux CloudWatch événements

Cet exemple affiche toutes les règles relatives aux CloudWatch événements de la région :

```
aws events list-rules
```

Pour afficher une liste de règles relatives aux CloudWatch événements commençant par une certaine chaîne.

Cet exemple affiche toutes les règles relatives aux CloudWatch événements de la région dont le nom commence par « Quotidien » :

```
aws events list-rules --name-prefix "Daily"
```

- Pour plus de détails sur l'API, reportez-vous ListRulesà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Activez une règle à l'aide de son nom.

```
    public static void listRules(EventBridgeClient eventBrClient) {
        try {
            ListRulesRequest rulesRequest = ListRulesRequest.builder()
                    .eventBusName("default")
                    .limit(10)
                    .build();

            ListRulesResponse response = eventBrClient.listRules(rulesRequest);
            List<Rule> rules = response.rules();
            for (Rule rule : rules) {
                System.out.println("The rule name is : " + rule.name());
                System.out.println("The rule description is : " +
rule.description());
                System.out.println("The rule state is : " +
rule.stateAsString());
            }

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

- Pour plus de détails sur l'API, reportez-vous ListRulesà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun listRules() {
    val rulesRequest = ListRulesRequest {
        eventBusName = "default"
        limit = 10
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listRules(rulesRequest)
        response.rules?.forEach { rule ->
            println("The rule name is ${rule.name}")
            println("The rule ARN is ${rule.arn}")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListRulesà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`ListTargetsByRule`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Répertoriez toutes les cibles d'une règle à l'aide de son nom.

```
/// <summary>
/// List all of the targets matching a rule by name.
/// </summary>
/// <param name="ruleName">The name of the rule.</param>
/// <returns>The list of targets.</returns>
public async Task<List<Target>> ListAllTargetsOnRule(string ruleName)
{
    var results = new List<Target>();
    var request = new ListTargetsByRuleRequest()
    {
        Rule = ruleName
    };
    ListTargetsByRuleResponse response;
    do
    {
        response = await _amazonEventBridge.ListTargetsByRuleAsync(request);
        results.AddRange(response.Targets);
        request.NextToken = response.NextToken;

    } while (response.NextToken is not null);

    return results;
}
```

- Pour plus de détails sur l'API, reportez-vous ListTargetsByRuleà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour afficher toutes les cibles d'une règle d' CloudWatch événements

Cet exemple affiche toutes les cibles de la règle nommée DailyLambdaFunction :

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- Pour plus de détails sur l'API, reportez-vous ListTargetsByRuleà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Répertoriez toutes les cibles d’une règle à l’aide de son nom.

```
   public static void listTargets(EventBridgeClient eventBrClient, String
ruleName) {
       ListTargetsByRuleRequest ruleRequest = ListTargetsByRuleRequest.builder()
               .rule(ruleName)
               .build();

       ListTargetsByRuleResponse res =
eventBrClient.listTargetsByRule(ruleRequest);
       List<Target> targetsList = res.targets();
       for (Target target: targetsList) {
           System.out.println("Target ARN: "+target.arn());
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous ListTargetsByRuleà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun listTargets(ruleName: String?) {
    val ruleRequest = ListTargetsByRuleRequest {
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(ruleRequest)
        response.targets?.forEach { target ->
            println("Target ARN: ${target.arn}")
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListTargetsByRuleà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutEvents`** avec un AWS SDK ou une CLI

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

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

- Création et déclenchement d'une règle
- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Envoyez un événement qui correspond à un modèle personnalisé pour une règle.

```
    /// <summary>
    /// Add an event to the event bus that includes an email, message, and time.
    /// </summary>
    /// <param name="email">The email to use in the event detail of the custom
event.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> PutCustomEmailEvent(string email)
    {
        var eventDetail = new
        {
            UserEmail = email,
            Message = "This event was generated by example code.",
            UtcTime = DateTime.UtcNow.ToString("g")
        };
        var response = await _amazonEventBridge.PutEventsAsync(
            new PutEventsRequest()
            {
                Entries = new List<PutEventsRequestEntry>()
                {
                    new PutEventsRequestEntry()
                    {
                        Source = "ExampleSource",
```

```
                    Detail = JsonSerializer.Serialize(eventDetail),
                    DetailType = "ExampleType"
                }
            }
        });

    return response.FailedEntryCount == 0;
}
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Joignez les fichiers requis.

```
#include <aws/core/Aws.h>
#include <aws/events/EventBridgeClient.h>
#include <aws/events/model/PutEventsRequest.h>
#include <aws/events/model/PutEventsResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Envoyez l'événement.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;

        Aws::CloudWatchEvents::Model::PutEventsRequestEntry event_entry;
        event_entry.SetDetail(MakeDetails(event_key, event_value));
        event_entry.SetDetailType("sampleSubmitted");
        event_entry.AddResources(resource_arn);
```

```
        event_entry.SetSource("aws-sdk-cpp-cloudwatch-example");

        Aws::CloudWatchEvents::Model::PutEventsRequest request;
        request.AddEntries(event_entry);

        auto outcome = cwe.PutEvents(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to post CloudWatch event: " <<
                outcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout << "Successfully posted CloudWatch event" << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour envoyer un événement personnalisé à CloudWatch Events

Cet exemple envoie un événement personnalisé à CloudWatch Events. L'événement est contenu dans le fichier putevents.json :

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

Voici le contenu du fichier putevents.json :

```
[
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value1\", \"key2\": \"value2\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
```

```
  },
  {
    "Source": "com.mycompany.myapp",
    "Detail": "{ \"key1\": \"value3\", \"key2\": \"value4\" }",
    "Resources": [
      "resource1",
      "resource2"
    ],
    "DetailType": "myDetailType"
   }
]
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
   public static void triggerCustomRule(EventBridgeClient eventBrClient, String
email) {
       String json = "{" +
               "\"UserEmail\": \"" + email + "\"," +
               "\"Message\": \"This event was generated by example code.\"," +
               "\"UtcTime\": \"Now.\"" +
               "}";

       PutEventsRequestEntry entry = PutEventsRequestEntry.builder()
               .source("ExampleSource")
               .detail(json)
               .detailType("ExampleType")
               .build();

       PutEventsRequest eventsRequest = PutEventsRequest.builder()
               .entries(entry)
```

```
                .build();

        eventBrClient.putEvents(eventsRequest);
    }
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

```
import {
  EventBridgeClient,
  PutEventsCommand,
} from "@aws-sdk/client-eventbridge";

export const putEvents = async (
  source = "eventbridge.integration.test",
  detailType = "greeting",
  resources = [],
) => {
  const client = new EventBridgeClient({});

  const response = await client.send(
    new PutEventsCommand({
      Entries: [
        {
          Detail: JSON.stringify({ greeting: "Hello there." }),
          DetailType: detailType,
          Resources: resources,
          Source: source,
        },
```

```
      ],
    }),
  );

  console.log("PutEvents response:");
  console.log(response);
  // PutEvents response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: '3d0df73d-dcea-4a23-ae0d-f5556a3ac109',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   Entries: [ { EventId: '51620841-5af4-6402-d9bc-b77734991eb5' } ],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS SDK for JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

// Create CloudWatchEvents service object
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });
```

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

ebevents.putEvents(params, function (err, data) {
  if (err) {
    console.log("Error", err);
  } else {
    console.log("Success", data.Entries);
  }
});
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun triggerCustomRule(email: String) {
    val json = "{" +
        "\"UserEmail\": \"" + email + "\"," +
        "\"Message\": \"This event was generated by example code.\"" +
        "\"UtcTime\": \"Now.\"" +
        "}"

    val entry = PutEventsRequestEntry {
```

```
        source = "ExampleSource"
        detail = json
        detailType = "ExampleType"
    }

    val eventsRequest = PutEventsRequest {
        this.entries = listOf(entry)
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putEvents(eventsRequest)
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutEventsà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutRule`** avec un AWS SDK ou une CLI

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

Les exemples d’actions sont des extraits de code de programmes de plus grande envergure et doivent être exécutés en contexte. Vous pouvez voir cette action en contexte dans les exemples de code suivants :

- Création et déclenchement d’une règle
- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une règle qui se déclenche lorsqu'un objet est ajouté à un compartiment Amazon Simple Storage Service.

```
    /// <summary>
    /// Create a new event rule that triggers when an Amazon S3 object is created
 in a bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role.</param>
    /// <param name="ruleName">The name to give the rule.</param>
    /// <param name="bucketName">The name of the bucket to trigger the event.</
param>
    /// <returns>The ARN of the new rule.</returns>
    public async Task<string> PutS3UploadRule(string roleArn, string ruleName,
 string bucketName)
    {
        string eventPattern = "{" +
                              "\"source\": [\"aws.s3\"]," +
                                  "\"detail-type\": [\"Object Created\"]," +
                                  "\"detail\": {" +
                                      "\"bucket\": {" +
                                          "\"name\": [\"" + bucketName + "\"]"
 +
                                      "}" +
                                  "}" +
                              "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Example S3 upload rule for EventBridge",
                RoleArn = roleArn,
```

```
                EventPattern = eventPattern
            });

        return response.RuleArn;
    }
```

Créez une règle qui utilise un modèle personnalisé.

```
    /// <summary>
    /// Update a rule to use a custom defined event pattern.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <returns>The ARN of the updated rule.</returns>
    public async Task<string> UpdateCustomEventPattern(string ruleName)
    {
        string customEventsPattern = "{" +
                                     "\"source\": [\"ExampleSource\"]," +
                                     "\"detail-type\": [\"ExampleType\"]" +
                                     "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Custom test rule",
                EventPattern = customEventsPattern
            });

        return response.RuleArn;
    }
```

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Joignez les fichiers requis.

```
#include <aws/core/Aws.h>
#include <aws/events/EventBridgeClient.h>
#include <aws/events/model/PutRuleRequest.h>
#include <aws/events/model/PutRuleResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Créez la règle .

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;
        Aws::CloudWatchEvents::Model::PutRuleRequest request;
        request.SetName(rule_name);
        request.SetRoleArn(role_arn);
        request.SetScheduleExpression("rate(5 minutes)");
        request.SetState(Aws::CloudWatchEvents::Model::RuleState::ENABLED);

        auto outcome = cwe.PutRule(request);
        if (!outcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch events rule " <<
                rule_name << ": " << outcome.GetError().GetMessage() <<
                std::endl;
        }
        else
        {
            std::cout << "Successfully created CloudWatch events rule " <<
                rule_name << " with resulting Arn " <<
                outcome.GetResult().GetRuleArn() << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour créer des règles relatives aux CloudWatch événements

Cet exemple crée une règle qui se déclenche chaque jour à 9 h 00 (UTC). Si vous utilisez put-targets pour ajouter une fonction Lambda comme cible de cette règle, vous pouvez exécuter la fonction Lambda tous les jours à l'heure spécifiée :

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9
 * * ? *)"
```

Cet exemple crée une règle qui se déclenche lorsqu'une instance EC2 de la région change d'état :

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

Cet exemple crée une règle qui se déclenche lorsqu'une instance EC2 de la région est arrêtée ou supprimée :

```
aws events put-rule --name "EC2InstanceStateChangeStopOrTerminate" --event-
pattern "{\"source\":[\"aws.ec2\"],\"detail-type\":[\"EC2 Instance State-change
 Notification\"],\"detail\":{\"state\":[\"stopped\",\"terminated\"]}}" --role-arn
 "arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une règle planifiée.

```
   public static void createEBRule(EventBridgeClient eventBrClient, String
ruleName, String cronExpression) {
       try {
           PutRuleRequest ruleRequest = PutRuleRequest.builder()
                   .name(ruleName)
                   .eventBusName("default")
                   .scheduleExpression(cronExpression)
                   .state("ENABLED")
                   .description("A test rule that runs on a schedule created by
the Java API")
                   .build();

           PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
           System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

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

Créez une règle qui se déclenche lorsqu'un objet est ajouté à un compartiment Amazon Simple Storage Service.

```
   // Create a new event rule that triggers when an Amazon S3 object is created
in
   // a bucket.
```

```
   public static void addEventRule(EventBridgeClient eventBrClient, String
roleArn, String bucketName,
           String eventRuleName) {
       String pattern = "{\n" +
               "  \"source\": [\"aws.s3\"],\n" +
               "  \"detail-type\": [\"Object Created\"],\n" +
               "  \"detail\": {\n" +
               "    \"bucket\": {\n" +
               "      \"name\": [\"" + bucketName + "\"]\n" +
               "    }\n" +
               "  }\n" +
               "}";

       try {
           PutRuleRequest ruleRequest = PutRuleRequest.builder()
                   .description("Created by using the AWS SDK for Java v2")
                   .name(eventRuleName)
                   .eventPattern(pattern)
                   .roleArn(roleArn)
                   .build();

           PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
           System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

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

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

```
import { EventBridgeClient, PutRuleCommand } from "@aws-sdk/client-eventbridge";

export const putRule = async (
  ruleName = "some-rule",
  source = "some-source",
) => {
  const client = new EventBridgeClient({});

  const response = await client.send(
    new PutRuleCommand({
      Name: ruleName,
      EventPattern: JSON.stringify({ source: [source] }),
      State: "ENABLED",
      EventBusName: "default",
    }),
  );

  console.log("PutRule response:");
  console.log(response);
  // PutRule response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'd7292ced-1544-421b-842f-596326bc7072',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   RuleArn: 'arn:aws:events:us-east-1:xxxxxxxxxxxx:rule/
EventBridgeTestRule-1696280037720'
```

```
  // }
  return response;
};
```

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section Référence des AWS SDK for JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

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

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

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

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Créez une règle planifiée.

```
suspend fun createScRule(ruleName: String?, cronExpression: String?) {
    val ruleRequest = PutRuleRequest {
        name = ruleName
        eventBusName = "default"
        scheduleExpression = cronExpression
        state = RuleState.Enabled
        description = "A test rule that runs on a schedule created by the Kotlin
 API"
    }

    EventBridgeClient { region = "us-west-2" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}
```

Créez une règle qui se déclenche lorsqu'un objet est ajouté à un compartiment Amazon Simple Storage Service.

```
// Create a new event rule that triggers when an Amazon S3 object is created in a
 bucket.
suspend fun addEventRule(roleArnVal: String?, bucketName: String, eventRuleName:
 String?) {
    val pattern = """{
        "source": ["aws.s3"],
        "detail-type": ["Object Created"],
        "detail": {
        "bucket": {
            "name": ["$bucketName"]
```

```
            }
        }
    }"""

    val ruleRequest = PutRuleRequest {
        description = "Created by using the AWS SDK for Kotlin"
        name = eventRuleName
        eventPattern = pattern
        roleArn = roleArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutRuleà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`PutTargets`** avec un AWS SDK ou une CLI

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

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

- Démarrer avec les règles et les cibles

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Ajoutez une rubrique Amazon SNS en tant que cible pour une règle.

```
    /// <summary>
    /// Add an Amazon SNS target topic to a rule.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <param name="targetArn">The ARN of the Amazon SNS target.</param>
    /// <param name="eventBusArn">The optional event bus name, uses default if
empty.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> AddSnsTargetToRule(string ruleName, string
targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        // Create the list of targets and add a new target.
        var targets = new List<Target>
        {
            new Target()
            {
                Arn = targetArn,
                Id = targetID
            }
        };

        // Add the targets to the rule.
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
```

```
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }

        return targetID;
    }
```

Ajoutez un transformateur d'entrée à une cible pour une règle.

```
    /// <summary>
    /// Update an Amazon S3 object created rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateS3UploadRuleTargetWithTransform(string
ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputPathsMap = new Dictionary<string, string>()
                    {
                        {"bucket", "$.detail.bucket.name"},
                        {"time", "$.time"}
                    },
```

```
                    InputTemplate = "\"Notification: an object was uploaded to
bucket <bucket> at <time>.\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }
        return targetID;
    }
```

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section Référence des AWS SDK for .NET API.

C++

SDK pour C++

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Joignez les fichiers requis.

```
#include <aws/core/Aws.h>
```

```
#include <aws/events/EventBridgeClient.h>
#include <aws/events/model/PutTargetsRequest.h>
#include <aws/events/model/PutTargetsResult.h>
#include <aws/core/utils/Outcome.h>
#include <iostream>
```

Ajoutez la cible.

```
        Aws::CloudWatchEvents::EventBridgeClient cwe;

        Aws::CloudWatchEvents::Model::Target target;
        target.SetArn(lambda_arn);
        target.SetId(target_id);

        Aws::CloudWatchEvents::Model::PutTargetsRequest request;
        request.SetRule(rule_name);
        request.AddTargets(target);

        auto putTargetsOutcome = cwe.PutTargets(request);
        if (!putTargetsOutcome.IsSuccess())
        {
            std::cout << "Failed to create CloudWatch events target for rule "
                << rule_name << ": " <<
                putTargetsOutcome.GetError().GetMessage() << std::endl;
        }
        else
        {
            std::cout <<
                "Successfully created CloudWatch events target for rule "
                << rule_name << std::endl;
        }
```

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section Référence des AWS SDK for C++ API.

CLI

AWS CLI

Pour ajouter des cibles aux règles relatives aux CloudWatch événements

Cet exemple ajoute une fonction Lambda comme cible d'une règle :

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

Cet exemple définit un flux Amazon Kinesis comme cible, afin que les événements concernés par cette règle soient relayés vers le flux :

```
aws events put-targets --rule EC2InstanceStateChanges --targets
 "Id"="1","Arn"="arn:aws:kinesis:us-east-1:123456789012:stream/
MyStream","RoleArn"="arn:aws:iam::123456789012:role/MyRoleForThisRule"
```

Cet exemple définit deux flux Amazon Kinesis comme cibles pour une règle :

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="Target1","Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream1","RoleArn"="arn:aws:iam::379642911888:role/ MyRoleToAccessLambda"
  "Id"="Target2"," Arn"="arn:aws:kinesis:us-east-1:379642911888:stream/
MyStream2","RoleArn"="arn:aws:iam::379642911888:role/MyRoleToAccessLambda"
```

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Ajoutez une rubrique Amazon SNS en tant que cible pour une règle.

```
    // Add a rule which triggers an SNS target when a file is uploaded to an S3
    // bucket.
    public static void addSnsEventRule(EventBridgeClient eventBrClient, String
 ruleName, String topicArn,
```

```
            String topicName, String eventRuleName, String bucketName) {
        String targetID = java.util.UUID.randomUUID().toString();
        Target myTarget = Target.builder()
                .id(targetID)
                .arn(topicArn)
                .build();

        List<Target> targets = new ArrayList<>();
        targets.add(myTarget);
        PutTargetsRequest request = PutTargetsRequest.builder()
                .eventBusName(null)
                .targets(targets)
                .rule(ruleName)
                .build();

        eventBrClient.putTargets(request);
        System.out.println("Added event rule " + eventRuleName + " with Amazon
SNS target " + topicName + " for bucket "
                + bucketName + ".");
    }
```

Ajoutez un transformateur d'entrée à une cible pour une règle.

```
    public static void updateCustomRuleTargetWithTransform(EventBridgeClient
eventBrClient, String topicArn,
            String ruleName) {
        String targetId = java.util.UUID.randomUUID().toString();
        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: sample event was received.\"")
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
                    .targets(target)
                    .eventBusName(null)
```

```
                    .build();

            eventBrClient.putTargets(targetsRequest);
        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }
```

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

```
import {
  EventBridgeClient,
  PutTargetsCommand,
} from "@aws-sdk/client-eventbridge";

export const putTarget = async (
  existingRuleName = "some-rule",
  targetArn = "arn:aws:lambda:us-east-1:000000000000:function:test-func",
  uniqueId = Date.now().toString(),
) => {
  const client = new EventBridgeClient({});
  const response = await client.send(
    new PutTargetsCommand({
      Rule: existingRuleName,
      Targets: [
        {
          Arn: targetArn,
```

```
          Id: uniqueId,
        },
      ],
    }),
  );

  console.log("PutTargets response:");
  console.log(response);
  // PutTargets response:
  // {
  //   '$metadata': {
  //     httpStatusCode: 200,
  //     requestId: 'f5b23b9a-2c17-45c1-ad5c-f926c3692e3d',
  //     extendedRequestId: undefined,
  //     cfId: undefined,
  //     attempts: 1,
  //     totalRetryDelay: 0
  //   },
  //   FailedEntries: [],
  //   FailedEntryCount: 0
  // }

  return response;
};
```

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section Référence des AWS SDK for JavaScript API.

SDK pour JavaScript (v2)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

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

// Create CloudWatchEvents service object
```

```
var ebevents = new AWS.EventBridge({ apiVersion: "2015-10-07" });

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

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

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section Référence des AWS SDK for JavaScript API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
// Add a rule that triggers an SNS target when a file is uploaded to an S3
 bucket.
suspend fun addSnsEventRule(ruleName: String?, topicArn: String?, topicName:
 String, eventRuleName: String, bucketName: String) {
    val targetID = UUID.randomUUID().toString()
    val myTarget = Target {
        id = targetID
```

```
        arn = topicArn
    }

    val targetsOb = mutableListOf<Target>()
    targetsOb.add(myTarget)

    val request = PutTargetsRequest {
        eventBusName = null
        targets = targetsOb
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(request)
        println("Added event rule $eventRuleName with Amazon SNS target
 $topicName for bucket $bucketName.")
    }
}
```

Ajoutez un transformateur d'entrée à une cible pour une règle.

```
suspend fun updateCustomRuleTargetWithTransform(topicArn: String?, ruleName:
 String?) {
    val targetId = UUID.randomUUID().toString()

    val inputTransformerOb = InputTransformer {
        inputTemplate = "\"Notification: sample event was received.\""
    }

    val target = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransformerOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(target)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
```

```
        eventBrClient.putTargets(targetsRequest)
    }
}
```

- Pour plus de détails sur l'API, reportez-vous PutTargetsà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **`RemoveTargets`** avec un AWS SDK ou une CLI

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

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Retirez toutes les cibles d’une règle à l’aide de son nom.

```
    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> RemoveAllTargetsFromRule(string ruleName)
    {
        var targetIds = new List<string>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse targetsResponse;
```

```
        do
        {
            targetsResponse = await
_amazonEventBridge.ListTargetsByRuleAsync(request);
            targetIds.AddRange(targetsResponse.Targets.Select(t => t.Id));
            request.NextToken = targetsResponse.NextToken;

        } while (targetsResponse.NextToken is not null);

        var removeResponse = await _amazonEventBridge.RemoveTargetsAsync(
            new RemoveTargetsRequest()
            {
                Rule = ruleName,
                Ids = targetIds
            });

        if (removeResponse.FailedEntryCount > 0)
        {
            removeResponse.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to remove target {e.TargetId}: {e.ErrorMessage},
code {e.ErrorCode}");
            });
        }

        return removeResponse.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Pour plus de détails sur l'API, reportez-vous RemoveTargetsà la section Référence des AWS SDK for .NET API.

CLI

AWS CLI

Pour supprimer une cible pour un événement

Cet exemple supprime le flux Amazon Kinesis nommé MyStream 1 de la cible de la règle. DailyLambdaFunction Lors DailyLambdaFunction de sa création, ce flux a été défini comme cible avec l'ID Target1 :

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- Pour plus de détails sur l'API, reportez-vous RemoveTargetsà la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Retirez toutes les cibles d’une règle à l’aide de son nom.

```
   public static void deleteTargetsFromRule(EventBridgeClient eventBrClient,
String eventRuleName) {
       // First, get all targets that will be deleted.
       ListTargetsByRuleRequest request = ListTargetsByRuleRequest.builder()
               .rule(eventRuleName)
               .build();

       ListTargetsByRuleResponse response =
eventBrClient.listTargetsByRule(request);
       List<Target> allTargets = response.targets();

       // Get all targets and delete them.
       for (Target myTarget : allTargets) {
           RemoveTargetsRequest removeTargetsRequest =
RemoveTargetsRequest.builder()
                   .rule(eventRuleName)
                   .ids(myTarget.id())
                   .build();

           eventBrClient.removeTargets(removeTargetsRequest);
           System.out.println("Successfully removed the target");
       }
   }
```

- Pour plus de détails sur l'API, reportez-vous RemoveTargetsà la section Référence des AWS SDK for Java 2.x API.

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
suspend fun deleteTargetsFromRule(eventRuleName: String?) {
    // First, get all targets that will be deleted.
    val request = ListTargetsByRuleRequest {
        rule = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(request)
        val allTargets = response.targets

        // Get all targets and delete them.
        if (allTargets != null) {
            for (myTarget in allTargets) {
                val removeTargetsRequest = RemoveTargetsRequest {
                    rule = eventRuleName
                    ids = listOf(myTarget.id.toString())
                }
                eventBrClient.removeTargets(removeTargetsRequest)
                println("Successfully removed the target")
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous RemoveTargetsà la section AWS SDK pour la référence de l'API Kotlin.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Scénarios d' EventBridge utilisation des AWS SDK

Les exemples de code suivants vous montrent comment implémenter des scénarios courants EventBridge avec AWS les SDK. Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions EventBridge. Chaque scénario inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter le code.

Exemples

- Créez et déclenchez une règle dans Amazon à EventBridge l'aide d'un AWS SDK
- Commencez avec EventBridge les règles et les cibles à l'aide d'un AWS SDK

Créez et déclenchez une règle dans Amazon à EventBridge l'aide d'un AWS SDK

L'exemple de code suivant montre comment créer et déclencher une règle dans Amazon EventBridge.

Ruby

Kit SDK pour Ruby

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Appelez les fonctions dans le bon ordre.

```
require "aws-sdk-sns"
require "aws-sdk-iam"
require "aws-sdk-cloudwatchevents"
require "aws-sdk-ec2"
require "aws-sdk-cloudwatch"
```

```
require "aws-sdk-cloudwatchlogs"
require "securerandom"
```

Vérifie si la rubrique Amazon Simple Notification Service (Amazon SNS) spécifiée existe parmi celles fournies pour cette fonction.

```
# Checks whether the specified Amazon SNS
# topic exists among those provided to this function.
# This is a helper function that is called by the topic_exists? function.
#
# @param topics [Array] An array of Aws::SNS::Types::Topic objects.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
#   sns_client = Aws::SNS::Client.new(region: 'us-east-1')
#   response = sns_client.list_topics
#   if topic_found?(
#     response.topics,
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
#     puts 'Topic found.'
#   end

def topic_found?(topics, topic_arn)
  topics.each do |topic|
    return true if topic.topic_arn == topic_arn
  end
  return false
end
```

Vérifie si la rubrique spécifiée existe parmi celles disponibles pour l'appelant dans Amazon SNS.

```
# Checks whether the specified topic exists among those available to the
# caller in Amazon SNS.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_arn [String] The ARN of the topic to find.
# @return [Boolean] true if the topic ARN was found; otherwise, false.
# @example
```

```
#   exit 1 unless topic_exists?(
#     Aws::SNS::Client.new(region: 'us-east-1'),
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def topic_exists?(sns_client, topic_arn)
  puts "Searching for topic with ARN '#{topic_arn}'..."
  response = sns_client.list_topics
  if response.topics.count.positive?
    if topic_found?(response.topics, topic_arn)
      puts "Topic found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.topics.count.positive?
        if topic_found?(response.topics, topic_arn)
          puts "Topic found."
          return true
        end
      end
    end
  end
  puts "Topic not found."
  return false
rescue StandardError => e
  puts "Topic not found: #{e.message}"
  return false
end
```

Créez une rubrique dans Amazon SNS, puis abonnez-y une adresse e-mail pour recevoir des notifications relatives à cette rubrique.

```
# Creates a topic in Amazon SNS
# and then subscribes an email address to receive notifications to that topic.
#
# @param sns_client [Aws::SNS::Client] An initialized Amazon SNS client.
# @param topic_name [String] The name of the topic to create.
# @param email_address [String] The email address of the recipient to notify.
# @return [String] The ARN of the topic that was created.
# @example
#   puts create_topic(
#     Aws::SNS::Client.new(region: 'us-east-1'),
```

```
#     'aws-doc-sdk-examples-topic',
#     'mary@example.com'
#   )
def create_topic(sns_client, topic_name, email_address)
  puts "Creating the topic named '#{topic_name}'..."
  topic_response = sns_client.create_topic(name: topic_name)
  puts "Topic created with ARN '#{topic_response.topic_arn}'."
  subscription_response = sns_client.subscribe(
    topic_arn: topic_response.topic_arn,
    protocol: "email",
    endpoint: email_address,
    return_subscription_arn: true
  )
  puts "Subscription created with ARN " \
    "'#{subscription_response.subscription_arn}'. Have the owner of the " \
    "email address '#{email_address}' check their inbox in a few minutes " \
    "and confirm the subscription to start receiving notification emails."
  return topic_response.topic_arn
rescue StandardError => e
  puts "Error creating or subscribing to topic: #{e.message}"
  return "Error"
end
```

Vérifiez si le rôle AWS Identity and Access Management (IAM) spécifié existe parmi ceux fournis à cette fonction.

```
# Checks whether the specified AWS Identity and Access Management (IAM)
# role exists among those provided to this function.
# This is a helper function that is called by the role_exists? function.
#
# @param roles [Array] An array of Aws::IAM::Role objects.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   iam_client = Aws::IAM::Client.new(region: 'us-east-1')
#   response = iam_client.list_roles
#   if role_found?(
#     response.roles,
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
#     puts 'Role found.'
#   end
```

```
def role_found?(roles, role_arn)
  roles.each do |role|
    return true if role.arn == role_arn
  end
  return false
end
```

Vérifiez si le rôle spécifié existe parmi ceux disponibles pour l'appelant dans IAM.

```
# Checks whether the specified role exists among those available to the
# caller in AWS Identity and Access Management (IAM).
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_arn [String] The ARN of the role to find.
# @return [Boolean] true if the role ARN was found; otherwise, false.
# @example
#   exit 1 unless role_exists?(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change'
#   )
def role_exists?(iam_client, role_arn)
  puts "Searching for role with ARN '#{role_arn}'..."
  response = iam_client.list_roles
  if response.roles.count.positive?
    if role_found?(response.roles, role_arn)
      puts "Role found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.roles.count.positive?
        if role_found?(response.roles, role_arn)
          puts "Role found."
          return true
        end
      end
    end
  end
  puts "Role not found."
  return false
rescue StandardError => e
  puts "Role not found: #{e.message}"
```

```
  return false
end
```

Créez un rôle dans IAM.

```
# Creates a role in AWS Identity and Access Management (IAM).
# This role is used by a rule in Amazon EventBridge to allow
# that rule to operate within the caller's account.
# This role is designed to be used specifically by this code example.
#
# @param iam_client [Aws::IAM::Client] An initialized IAM client.
# @param role_name [String] The name of the role to create.
# @return [String] The ARN of the role that was created.
# @example
#   puts create_role(
#     Aws::IAM::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def create_role(iam_client, role_name)
  puts "Creating the role named '#{role_name}'..."
  response = iam_client.create_role(
    assume_role_policy_document: {
      'Version': "2012-10-17",
      'Statement': [
        {
          'Sid': "",
          'Effect': "Allow",
          'Principal': {
            'Service': "events.amazonaws.com"
          },
          'Action': "sts:AssumeRole"
        }
      ]
    }.to_json,
    path: "/",
    role_name: role_name
  )
  puts "Role created with ARN '#{response.role.arn}'."
  puts "Adding access policy to role..."
  iam_client.put_role_policy(
    policy_document: {
      'Version': "2012-10-17",
```

```
      'Statement': [
        {
          'Sid': "CloudWatchEventsFullAccess",
          'Effect': "Allow",
          'Resource': "*",
          'Action': "events:*"
        },
        {
          'Sid': "IAMPassRoleForCloudWatchEvents",
          'Effect': "Allow",
          'Resource': "arn:aws:iam::*:role/AWS_Events_Invoke_Targets",
          'Action': "iam:PassRole"
        }
      ]
    }.to_json,
    policy_name: "CloudWatchEventsPolicy",
    role_name: role_name
  )
  puts "Access policy added to role."
  return response.role.arn
rescue StandardError => e
  puts "Error creating role or adding policy to it: #{e.message}"
  puts "If the role was created, you must add the access policy " \
    "to the role yourself, or delete the role yourself and try again."
  return "Error"
end
```

Vérifie si la EventBridge règle spécifiée existe parmi celles fournies à cette fonction.

```
# Checks whether the specified Amazon EventBridge rule exists among
# those provided to this function.
# This is a helper function that is called by the rule_exists? function.
#
# @param rules [Array] An array of Aws::CloudWatchEvents::Types::Rule objects.
# @param rule_arn [String] The name of the rule to find.
# @return [Boolean] true if the name of the rule was found; otherwise, false.
# @example
#   cloudwatchevents_client = Aws::CloudWatch::Client.new(region: 'us-east-1')
#   response = cloudwatchevents_client.list_rules
#   if rule_found?(response.rules, 'aws-doc-sdk-examples-ec2-state-change')
#     puts 'Rule found.'
#   end
```

```
def rule_found?(rules, rule_name)
  rules.each do |rule|
    return true if rule.name == rule_name
  end
  return false
end
```

Vérifie si la règle spécifiée existe parmi celles disponibles pour l'appelant. EventBridge

```
# Checks whether the specified rule exists among those available to the
# caller in Amazon EventBridge.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to find.
# @return [Boolean] true if the rule name was found; otherwise, false.
# @example
#   exit 1 unless rule_exists?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1')
#     'aws-doc-sdk-examples-ec2-state-change'
#   )
def rule_exists?(cloudwatchevents_client, rule_name)
  puts "Searching for rule with name '#{rule_name}'..."
  response = cloudwatchevents_client.list_rules
  if response.rules.count.positive?
    if rule_found?(response.rules, rule_name)
      puts "Rule found."
      return true
    end
    while response.next_page? do
      response = response.next_page
      if response.rules.count.positive?
        if rule_found?(response.rules, rule_name)
          puts "Rule found."
          return true
        end
      end
    end
  end
  puts "Rule not found."
  return false
rescue StandardError => e
```

```
  puts "Rule not found: #{e.message}"
  return false
end
```

Créez une règle dans EventBridge.

```
# Creates a rule in Amazon EventBridge.
# This rule is triggered whenever an available instance in
# Amazon EC2 changes to the specified state.
# This rule is designed to be used specifically by this code example.
#
# Prerequisites:
#
# - A role in AWS Identity and Access Management (IAM) that is designed
#   to be used specifically by this code example.
# - A topic in Amazon SNS.
#
# @param cloudwatchevents_client [Aws::CloudWatchEvents::Client]
#   An initialized Amazon EventBridge client.
# @param rule_name [String] The name of the rule to create.
# @param rule_description [String] Some description for this rule.
# @param instance_state [String] The state that available instances in
#   Amazon EC2 must change to, to
#   trigger this rule.
# @param role_arn [String] The Amazon Resource Name (ARN) of the IAM role.
# @param target_id [String] Some identifying string for the rule's target.
# @param topic_arn [String] The ARN of the Amazon SNS topic.
# @return [Boolean] true if the rule was created; otherwise, false.
# @example
#   exit 1 unless rule_created?(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     'Triggers when any available EC2 instance starts.',
#     'running',
#     'arn:aws:iam::111111111111:role/aws-doc-sdk-examples-ec2-state-change',
#     'sns-topic',
#     'arn:aws:sns:us-east-1:111111111111:aws-doc-sdk-examples-topic'
#   )
def rule_created?(
  cloudwatchevents_client,
  rule_name,
  rule_description,
```

```
  instance_state,
  role_arn,
  target_id,
  topic_arn
)
  puts "Creating rule with name '#{rule_name}'..."
  put_rule_response = cloudwatchevents_client.put_rule(
    name: rule_name,
    description: rule_description,
    event_pattern: {
      'source': [
        "aws.ec2"
      ],
      'detail-type': [
        "EC2 Instance State-change Notification"
      ],
      'detail': {
        'state': [
          instance_state
        ]
      }
    }.to_json,
    state: "ENABLED",
    role_arn: role_arn
  )
  puts "Rule created with ARN '#{put_rule_response.rule_arn}'."

  put_targets_response = cloudwatchevents_client.put_targets(
    rule: rule_name,
    targets: [
      {
        id: target_id,
        arn: topic_arn
      }
    ]
  )
  if put_targets_response.key?(:failed_entry_count) &&
      put_targets_response.failed_entry_count > 0
    puts "Error(s) adding target to rule:"
    put_targets_response.failed_entries.each do |failure|
      puts failure.error_message
    end
    return false
  else
```

```
    return true
  end
rescue StandardError => e
  puts "Error creating rule or adding target to rule: #{e.message}"
  puts "If the rule was created, you must add the target " \
    "to the rule yourself, or delete the rule yourself and try again."
  return false
end
```

Vérifiez si le groupe de journaux spécifié existe parmi ceux mis à la disposition de l'appelant dans Amazon CloudWatch Logs.

```
# Checks to see whether the specified log group exists among those available
# to the caller in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to find.
# @return [Boolean] true if the log group name was found; otherwise, false.
# @example
#   exit 1 unless log_group_exists?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_exists?(cloudwatchlogs_client, log_group_name)
  puts "Searching for log group with name '#{log_group_name}'..."
  response = cloudwatchlogs_client.describe_log_groups(
    log_group_name_prefix: log_group_name
  )
  if response.log_groups.count.positive?
    response.log_groups.each do |log_group|
      if log_group.log_group_name == log_group_name
        puts "Log group found."
        return true
      end
    end
  end
  puts "Log group not found."
  return false
rescue StandardError => e
  puts "Log group not found: #{e.message}"
  return false
```

```
end
```

Créez un groupe de CloudWatch journaux dans Logs.

```
# Creates a log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group to create.
# @return [Boolean] true if the log group name was created; otherwise, false.
# @example
#   exit 1 unless log_group_created?(
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def log_group_created?(cloudwatchlogs_client, log_group_name)
  puts "Attempting to create log group with the name '#{log_group_name}'..."
  cloudwatchlogs_client.create_log_group(log_group_name: log_group_name)
  puts "Log group created."
  return true
rescue StandardError => e
  puts "Error creating log group: #{e.message}"
  return false
end
```

Écrivez un événement dans un flux de journal dans CloudWatch Logs.

```
# Writes an event to a log stream in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
# - A log stream within the log group.
#
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @param log_stream_name [String] The name of the log stream within
#   the log group.
# @param message [String] The message to write to the log stream.
# @param sequence_token [String] If available, the sequence token from the
```

```
#   message that was written immediately before this message. This sequence
#   token is returned by Amazon CloudWatch Logs whenever you programmatically
#   write a message to the log stream.
# @return [String] The sequence token that is returned by
#   Amazon CloudWatch Logs after successfully writing the message to the
#   log stream.
# @example
#   puts log_event(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#     '2020/11/19/53f985be-199f-408e-9a45-fc242df41fEX',
#     "Instance 'i-033c48ef067af3dEX' restarted.",
#     '495426724868310740095796045676567882148068632824696073EX'
#   )
def log_event(
  cloudwatchlogs_client,
  log_group_name,
  log_stream_name,
  message,
  sequence_token
)
  puts "Attempting to log '#{message}' to log stream '#{log_stream_name}'..."
  event = {
    log_group_name: log_group_name,
    log_stream_name: log_stream_name,
    log_events: [
      {
        timestamp: (Time.now.utc.to_f.round(3) * 1_000).to_i,
        message: message
      }
    ]
  }
  unless sequence_token.empty?
    event[:sequence_token] = sequence_token
  end

  response = cloudwatchlogs_client.put_log_events(event)
  puts "Message logged."
  return response.next_sequence_token
rescue StandardError => e
  puts "Message not logged: #{e.message}"
end
```

Redémarrez une instance Amazon Elastic Compute Cloud (Amazon EC2) et ajoutez des informations sur l'activité associée à un flux de journal dans Logs. CloudWatch

```
# Restarts an Amazon EC2 instance
# and adds information about the related activity to a log stream
# in Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - The Amazon EC2 instance to restart.
# - The log group in Amazon CloudWatch Logs to add related activity
#   information to.
#
# @param ec2_client [Aws::EC2::Client] An initialized Amazon EC2 client.
# @param cloudwatchlogs_client [Aws::CloudWatchLogs::Client]
#   An initialized Amazon CloudWatch Logs client.
# @param instance_id [String] The ID of the instance.
# @param log_group_name [String] The name of the log group.
# @return [Boolean] true if the instance was restarted and the information
#   was written to the log stream; otherwise, false.
# @example
#   exit 1 unless instance_restarted?(
#     Aws::EC2::Client.new(region: 'us-east-1'),
#     Aws::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'i-033c48ef067af3dEX',
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  log_stream_name = "#{Time.now.year}/#{Time.now.month}/#{Time.now.day}/" \
    "#{SecureRandom.uuid}"
  cloudwatchlogs_client.create_log_stream(
    log_group_name: log_group_name,
    log_stream_name: log_stream_name
  )
  sequence_token = ""

  puts "Attempting to stop the instance with the ID '#{instance_id}'. " \
```

```
    "This might take a few minutes..."
  ec2_client.stop_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_stopped, instance_ids: [instance_id])
  puts "Instance stopped."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' stopped.",
    sequence_token
  )

  puts "Attempting to restart the instance. This might take a few minutes..."
  ec2_client.start_instances(instance_ids: [instance_id])
  ec2_client.wait_until(:instance_running, instance_ids: [instance_id])
  puts "Instance restarted."
  sequence_token = log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Instance '#{instance_id}' restarted.",
    sequence_token
  )

  return true
rescue StandardError => e
  puts "Error creating log stream or stopping or restarting the instance: " \
    "#{e.message}"
  log_event(
    cloudwatchlogs_client,
    log_group_name,
    log_stream_name,
    "Error stopping or starting instance '#{instance_id}': #{e.message}",
    sequence_token
  )
  return false
end
```

Afficher les informations relatives à l'activité d'une règle dans EventBridge.

```
# Displays information about activity for a rule in Amazon EventBridge.
#
```

```
# Prerequisites:
#
# - A rule in Amazon EventBridge.
#
# @param cloudwatch_client [Amazon::CloudWatch::Client] An initialized
#   Amazon CloudWatch client.
# @param rule_name [String] The name of the rule.
# @param start_time [Time] The timestamp that determines the first datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param end_time [Time] The timestamp that determines the last datapoint
#   to return. Can also be expressed as DateTime, Date, Integer, or String.
# @param period [Integer] The interval, in seconds, to check for activity.
# @example
#   display_rule_activity(
#     Aws::CloudWatch::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-ec2-state-change',
#     Time.now - 600, # Start checking from 10 minutes ago.
#     Time.now, # Check up until now.
#     60 # Check every minute during those 10 minutes.
#   )
def display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)
  puts "Attempting to display rule activity..."
  response = cloudwatch_client.get_metric_statistics(
    namespace: "AWS/Events",
    metric_name: "Invocations",
    dimensions: [
      {
        name: "RuleName",
        value: rule_name
      }
    ],
    start_time: start_time,
    end_time: end_time,
    period: period,
    statistics: ["Sum"],
    unit: "Count"
  )
```

```
  if response.key?(:datapoints) && response.datapoints.count.positive?
    puts "The event rule '#{rule_name}' was triggered:"
    response.datapoints.each do |datapoint|
      puts "  #{datapoint.sum} time(s) at #{datapoint.timestamp}"
    end
  else
    puts "The event rule '#{rule_name}' was not triggered during the " \
      "specified time period."
  end
rescue StandardError => e
  puts "Error getting information about event rule activity: #{e.message}"
end
```

Afficher les informations de journal pour tous les flux de journaux d'un groupe de CloudWatch journaux de journaux.

```
# Displays log information for all of the log streams in a log group in
# Amazon CloudWatch Logs.
#
# Prerequisites:
#
# - A log group in Amazon CloudWatch Logs.
#
# @param cloudwatchlogs_client [Amazon::CloudWatchLogs::Client] An initialized
#   Amazon CloudWatch Logs client.
# @param log_group_name [String] The name of the log group.
# @example
#   display_log_data(
#     Amazon::CloudWatchLogs::Client.new(region: 'us-east-1'),
#     'aws-doc-sdk-examples-cloudwatch-log'
#   )
def display_log_data(cloudwatchlogs_client, log_group_name)
  puts "Attempting to display log stream data for the log group " \
    "named '#{log_group_name}'..."
  describe_log_streams_response = cloudwatchlogs_client.describe_log_streams(
    log_group_name: log_group_name,
    order_by: "LastEventTime",
    descending: true
  )
  if describe_log_streams_response.key?(:log_streams) &&
      describe_log_streams_response.log_streams.count.positive?
    describe_log_streams_response.log_streams.each do |log_stream|
```

```
      get_log_events_response = cloudwatchlogs_client.get_log_events(
        log_group_name: log_group_name,
        log_stream_name: log_stream.log_stream_name
      )
      puts "\nLog messages for '#{log_stream.log_stream_name}':"
      puts "-" * (log_stream.log_stream_name.length + 20)
      if get_log_events_response.key?(:events) &&
          get_log_events_response.events.count.positive?
        get_log_events_response.events.each do |event|
          puts event.message
        end
      else
        puts "No log messages for this log stream."
      end
    end
  end
rescue StandardError => e
  puts "Error getting information about the log streams or their messages: " \
    "#{e.message}"
end
```

Afficher un rappel à l'appelant pour qu'il nettoie manuellement toutes les AWS ressources associées dont il n'a plus besoin.

```
# Displays a reminder to the caller to manually clean up any associated
# AWS resources that they no longer need.
#
# @param topic_name [String] The name of the Amazon SNS topic.
# @param role_name [String] The name of the IAM role.
# @param rule_name [String] The name of the Amazon EventBridge rule.
# @param log_group_name [String] The name of the Amazon CloudWatch Logs log
 group.
# @param instance_id [String] The ID of the Amazon EC2 instance.
# @example
#   manual_cleanup_notice(
#     'aws-doc-sdk-examples-topic',
#     'aws-doc-sdk-examples-cloudwatch-events-rule-role',
#     'aws-doc-sdk-examples-ec2-state-change',
#     'aws-doc-sdk-examples-cloudwatch-log',
#     'i-033c48ef067af3dEX'
#   )
```

```
def manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
)
  puts "-" * 10
  puts "Some of the following AWS resources might still exist in your account."
  puts "If you no longer want to use this code example, then to clean up"
  puts "your AWS account and avoid unexpected costs, you might want to"
  puts "manually delete any of the following resources if they exist:"
  puts "- The Amazon SNS topic named '#{topic_name}'."
  puts "- The IAM role named '#{role_name}'."
  puts "- The Amazon EventBridge rule named '#{rule_name}'."
  puts "- The Amazon CloudWatch Logs log group named '#{log_group_name}'."
  puts "- The Amazon EC2 instance with the ID '#{instance_id}'."
end

# Example usage:
def run_me
  # Properties for the Amazon SNS topic.
  topic_name = "aws-doc-sdk-examples-topic"
  email_address = "mary@example.com"
  # Properties for the IAM role.
  role_name = "aws-doc-sdk-examples-cloudwatch-events-rule-role"
  # Properties for the Amazon EventBridge rule.
  rule_name = "aws-doc-sdk-examples-ec2-state-change"
  rule_description = "Triggers when any available EC2 instance starts."
  instance_state = "running"
  target_id = "sns-topic"
  # Properties for the Amazon EC2 instance.
  instance_id = "i-033c48ef067af3dEX"
  # Properties for displaying the event rule's activity.
  start_time = Time.now - 600 # Go back over the past 10 minutes
                              # (10 minutes * 60 seconds = 600 seconds).
  end_time = Time.now
  period = 60 # Look back every 60 seconds over the past 10 minutes.
  # Properties for the Amazon CloudWatch Logs log group.
  log_group_name = "aws-doc-sdk-examples-cloudwatch-log"
  # AWS service clients for this code example.
  region = "us-east-1"
  sts_client = Aws::STS::Client.new(region: region)
  sns_client = Aws::SNS::Client.new(region: region)
  iam_client = Aws::IAM::Client.new(region: region)
  cloudwatchevents_client = Aws::CloudWatchEvents::Client.new(region: region)
  ec2_client = Aws::EC2::Client.new(region: region)
  cloudwatch_client = Aws::CloudWatch::Client.new(region: region)
```

```
cloudwatchlogs_client = Aws::CloudWatchLogs::Client.new(region: region)

# Get the caller's account ID for use in forming
# Amazon Resource Names (ARNs) that this code relies on later.
account_id = sts_client.get_caller_identity.account

# If the Amazon SNS topic doesn't exist, create it.
topic_arn = "arn:aws:sns:#{region}:#{account_id}:#{topic_name}"
unless topic_exists?(sns_client, topic_arn)
  topic_arn = create_topic(sns_client, topic_name, email_address)
  if topic_arn == "Error"
    puts "Could not create the Amazon SNS topic correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
    exit 1
  end
end

# If the IAM role doesn't exist, create it.
role_arn = "arn:aws:iam::#{account_id}:role/#{role_name}"
unless role_exists?(iam_client, role_arn)
  role_arn = create_role(iam_client, role_name)
  if role_arn == "Error"
    puts "Could not create the IAM role correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon EventBridge rule doesn't exist, create it.
unless rule_exists?(cloudwatchevents_client, rule_name)
  unless rule_created?(
    cloudwatchevents_client,
    rule_name,
    rule_description,
    instance_state,
    role_arn,
    target_id,
    topic_arn
  )
    puts "Could not create the Amazon EventBridge rule correctly. " \
      "Program stopped."
```

```
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# If the Amazon CloudWatch Logs log group doesn't exist, create it.
unless log_group_exists?(cloudwatchlogs_client, log_group_name)
  unless log_group_created?(cloudwatchlogs_client, log_group_name)
    puts "Could not create the Amazon CloudWatch Logs log group " \
    "correctly. Program stopped."
    manual_cleanup_notice(
      topic_name, role_name, rule_name, log_group_name, instance_id
    )
  end
end

# Restart the Amazon EC2 instance, which triggers the rule.
unless instance_restarted?(
  ec2_client,
  cloudwatchlogs_client,
  instance_id,
  log_group_name
)
  puts "Could not restart the instance to trigger the rule. " \
    "Continuing anyway to show information about the rule and logs..."
end

# Display how many times the rule was triggered over the past 10 minutes.
display_rule_activity(
  cloudwatch_client,
  rule_name,
  start_time,
  end_time,
  period
)

# Display related log data in Amazon CloudWatch Logs.
display_log_data(cloudwatchlogs_client, log_group_name)

# Reminder the caller to clean up any AWS resources that are used
# by this code example and are no longer needed.
manual_cleanup_notice(
  topic_name, role_name, rule_name, log_group_name, instance_id
```

```
  )
end

run_me if $PROGRAM_NAME == __FILE__
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for Ruby .
  - PutEvents
  - PutRule

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Commencez avec EventBridge les règles et les cibles à l'aide d'un AWS SDK

Les exemples de code suivants montrent comment :

- Créez une règle et ajoutez-y une cible.
- Activez et désactivez les règles.
- Répertoriez et mettez à jour les règles et les cibles.
- Envoyez des événements, puis nettoyez les ressources.

.NET

AWS SDK for .NET

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Exécutez un scénario interactif à une invite de commande.

```
public class EventBridgeScenario
```

```
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks with Amazon EventBridge:
    - Create a rule.
    - Add a target to a rule.
    - Enable and disable rules.
    - List rules and targets.
    - Update rules and targets.
    - Send events.
    - Delete the rule.
    */

    private static ILogger logger = null!;
    private static EventBridgeWrapper _eventBridgeWrapper = null!;
    private static IConfiguration _configuration = null!;

    private static IAmazonIdentityManagementService? _iamClient = null!;
    private static IAmazonSimpleNotificationService? _snsClient = null!;
    private static IAmazonS3 _s3Client = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for Amazon EventBridge.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
            services.AddAWSService<IAmazonEventBridge>()
            .AddAWSService<IAmazonIdentityManagementService>()
            .AddAWSService<IAmazonS3>()
            .AddAWSService<IAmazonSimpleNotificationService>()
            .AddTransient<EventBridgeWrapper>()
            )
            .Build();

        _configuration = new ConfigurationBuilder()
            .SetBasePath(Directory.GetCurrentDirectory())
```

```
            .AddJsonFile("settings.json") // Load settings from .json file.
            .AddJsonFile("settings.local.json",
                true) // Optionally, load local settings.
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<EventBridgeScenario>();

        ServicesSetup(host);

        string topicArn = "";
        string roleArn = "";

        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Welcome to the Amazon EventBridge example scenario.");
        Console.WriteLine(new string('-', 80));

        try
        {
            roleArn = await CreateRole();

            await CreateBucketWithEventBridgeEvents();

            await AddEventRule(roleArn);

            await ListEventRules();

            topicArn = await CreateSnsTopic();

            var email = await SubscribeToSnsTopic(topicArn);

            await AddSnsTarget(topicArn);

            await ListTargets();

            await ListRulesForTarget(topicArn);

            await UploadS3File(_s3Client);

            await ChangeRuleState(false);

            await GetRuleState();

            await UpdateSnsEventRule(topicArn);
```

```
            await ChangeRuleState(true);

            await UploadS3File(_s3Client);

            await UpdateToCustomRule(topicArn);

            await TriggerCustomRule(email);

            await CleanupResources(topicArn);
        }
        catch (Exception ex)
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
            await CleanupResources(topicArn);
        }
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("The Amazon EventBridge example scenario is
complete.");
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Populate the services for use within the console application.
    /// </summary>
    /// <param name="host">The services host.</param>
    private static void ServicesSetup(IHost host)
    {
        _eventBridgeWrapper =
host.Services.GetRequiredService<EventBridgeWrapper>();
        _snsClient =
host.Services.GetRequiredService<IAmazonSimpleNotificationService>();
        _s3Client = host.Services.GetRequiredService<IAmazonS3>();
        _iamClient =
host.Services.GetRequiredService<IAmazonIdentityManagementService>();
    }

    /// <summary>
    /// Create a role to be used by EventBridge.
    /// </summary>
    /// <returns>The role Amazon Resource Name (ARN).</returns>
    public static async Task<string> CreateRole()
    {
        Console.WriteLine(new string('-', 80));
```

```
        Console.WriteLine("Creating a role to use with EventBridge and attaching
 managed policy AmazonEventBridgeFullAccess.");
        Console.WriteLine(new string('-', 80));

        var roleName = _configuration["roleName"];

        var assumeRolePolicy = "{" +
                               "\"Version\": \"2012-10-17\"," +
                               "\"Statement\": [{" +
                               "\"Effect\": \"Allow\"," +
                               "\"Principal\": {" +
                               $"\"Service\": \"events.amazonaws.com\"" +
                               "}," +
                               "\"Action\": \"sts:AssumeRole\"" +
                               "}]" +
                               "}";

        var roleResult = await _iamClient!.CreateRoleAsync(
            new CreateRoleRequest()
            {
                AssumeRolePolicyDocument = assumeRolePolicy,
                Path = "/",
                RoleName = roleName
            });

        await _iamClient.AttachRolePolicyAsync(
            new AttachRolePolicyRequest()
            {
                PolicyArn = "arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess",
                RoleName = roleName
            });
        // Allow time for the role to be ready.
        Thread.Sleep(10000);
        return roleResult.Role.Arn;
    }

    /// <summary>
    /// Create an Amazon Simple Storage Service (Amazon S3) bucket with
 EventBridge events enabled.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task CreateBucketWithEventBridgeEvents()
    {
```

```
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Creating an S3 bucket with EventBridge events
enabled.");

        var testBucketName = _configuration["testBucketName"];

        var bucketExists = await
Amazon.S3.Util.AmazonS3Util.DoesS3BucketExistV2Async(_s3Client,
            testBucketName);

        if (!bucketExists)
        {
            await _s3Client.PutBucketAsync(new PutBucketRequest()
            {
                BucketName = testBucketName,
                UseClientRegion = true
            });
        }

        await _s3Client.PutBucketNotificationAsync(new
PutBucketNotificationRequest()
        {
            BucketName = testBucketName,
            EventBridgeConfiguration = new EventBridgeConfiguration()
        });

        Console.WriteLine($"\tAdded bucket {testBucketName} with EventBridge
events enabled.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create and upload a file to an S3 bucket to trigger an event.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task UploadS3File(IAmazonS3 s3Client)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Uploading a file to the test bucket. This will trigger
a subscription email.");

        var testBucketName = _configuration["testBucketName"];
```

```
        var fileName = $"example_upload_{DateTime.UtcNow.Ticks}.txt";

        // Create the file if it does not already exist.
        if (!File.Exists(fileName))
        {
            await using StreamWriter sw = File.CreateText(fileName);
            await sw.WriteLineAsync(
                "This is a sample file for testing uploads.");
        }

        await s3Client.PutObjectAsync(new PutObjectRequest()
        {
            FilePath = fileName,
            BucketName = testBucketName
        });

        Console.WriteLine($"\tPress Enter to continue.");
        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Create an Amazon Simple Notification Service (Amazon SNS) topic to use as
an EventBridge target.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task<string> CreateSnsTopic()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine(
            "Creating an Amazon Simple Notification Service (Amazon SNS) topic
for email subscriptions.");

        var topicName = _configuration["topicName"];

        string topicPolicy = "{" +
                             "\"Version\": \"2012-10-17\"," +
                             "\"Statement\": [{" +
                             "\"Sid\": \"EventBridgePublishTopic\"," +
                             "\"Effect\": \"Allow\"," +
                             "\"Principal\": {" +
                             $"\"Service\": \"events.amazonaws.com\"" +
                             "}," +
```

```
                            "\"Resource\": \"*\"," +
                            "\"Action\": \"sns:Publish\"" +
                            "}]" +
                            "}";

        var topicAttributes = new Dictionary<string, string>()
        {
            { "Policy", topicPolicy }
        };

        var topicResponse = await _snsClient!.CreateTopicAsync(new
CreateTopicRequest()
        {
            Name = topicName,
            Attributes = topicAttributes

        });

        Console.WriteLine($"\tAdded topic {topicName} for email subscriptions.");

        Console.WriteLine(new string('-', 80));

        return topicResponse.TopicArn;
    }

    /// <summary>
    /// Subscribe a user email to an SNS topic.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>The user's email.</returns>
    private static async Task<string> SubscribeToSnsTopic(string topicArn)
    {
        Console.WriteLine(new string('-', 80));


        string email = "";
        while (string.IsNullOrEmpty(email))
        {
            Console.WriteLine("Enter your email to subscribe to the Amazon SNS
topic:");
            email = Console.ReadLine()!;
        }

        var subscriptions = new List<string>();
```

```
        var paginatedSubscriptions =
_snsClient!.Paginators.ListSubscriptionsByTopic(
            new ListSubscriptionsByTopicRequest()
            {
                TopicArn = topicArn
            });

        // Get the entire list using the paginator.
        await foreach (var subscription in paginatedSubscriptions.Subscriptions)
        {
            subscriptions.Add(subscription.Endpoint);
        }

        if (subscriptions.Contains(email))
        {
            Console.WriteLine($"\tYour email is already subscribed.");
            Console.WriteLine(new string('-', 80));
            return email;
        }

        await _snsClient.SubscribeAsync(new SubscribeRequest()
        {
            TopicArn = topicArn,
            Protocol = "email",
            Endpoint = email
        });

        Console.WriteLine($"Use the link in the email you received to confirm
your subscription, then press Enter to continue.");

        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
        return email;
    }

    /// <summary>
    /// Add a rule which triggers when a file is uploaded to an S3 bucket.
    /// </summary>
    /// <param name="roleArn">The ARN of the role used by EventBridge.</param>
    /// <returns>Async task.</returns>
    private static async Task AddEventRule(string roleArn)
    {
        Console.WriteLine(new string('-', 80));
```

```
        Console.WriteLine("Creating an EventBridge event that sends an email when
an Amazon S3 object is created.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];

        await _eventBridgeWrapper.PutS3UploadRule(roleArn, eventRuleName,
testBucketName);
        Console.WriteLine($"\tAdded event rule {eventRuleName} for bucket
{testBucketName}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Add an SNS target to the rule.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>Async task.</returns>
    private static async Task AddSnsTarget(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Adding a target to the rule to that sends an email
when the rule is triggered.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];
        var topicName = _configuration["topicName"];
        await _eventBridgeWrapper.AddSnsTargetToRule(eventRuleName, topicArn);
        Console.WriteLine($"\tAdded event rule {eventRuleName} with Amazon SNS
target {topicName} for bucket {testBucketName}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List the event rules on the default event bus.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListEventRules()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Current event rules:");
```

```
        var rules = await _eventBridgeWrapper.ListAllRulesForEventBus();
        rules.ForEach(r => Console.WriteLine($"\tRule: {r.Name} Description:
{r.Description} State: {r.State}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Update the event target to use a transform.
    /// </summary>
    /// <param name="topicArn">The SNS topic ARN target to update.</param>
    /// <returns>Async task.</returns>
    private static async Task UpdateSnsEventRule(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Let's update the event target with a transform.");

        var eventRuleName = _configuration["eventRuleName"];
        var testBucketName = _configuration["testBucketName"];

        await
_eventBridgeWrapper.UpdateS3UploadRuleTargetWithTransform(eventRuleName,
topicArn);
        Console.WriteLine($"\tUpdated event rule {eventRuleName} with Amazon SNS
target {topicArn} for bucket {testBucketName}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Update the rule to use a custom event pattern.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task UpdateToCustomRule(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Updating the event pattern to be triggered by a custom
event instead.");

        var eventRuleName = _configuration["eventRuleName"];

        await _eventBridgeWrapper.UpdateCustomEventPattern(eventRuleName);
```

```
        Console.WriteLine($"\tUpdated event rule {eventRuleName} to custom
pattern.");
        await
_eventBridgeWrapper.UpdateCustomRuleTargetWithTransform(eventRuleName,
            topicArn);

        Console.WriteLine($"\tUpdated event target {topicArn}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Send rule events for a custom rule using the user's email address.
    /// </summary>
    /// <param name="email">The email address to include.</param>
    /// <returns>Async task.</returns>
    private static async Task TriggerCustomRule(string email)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("Sending an event to trigger the rule. This will
trigger a subscription email.");

        await _eventBridgeWrapper.PutCustomEmailEvent(email);

        Console.WriteLine($"\tEvents have been sent. Press Enter to continue.");
        Console.ReadLine();

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List all of the targets for a rule.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task ListTargets()
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("List all of the targets for a particular rule.");

        var eventRuleName = _configuration["eventRuleName"];
        var targets = await
_eventBridgeWrapper.ListAllTargetsOnRule(eventRuleName);
        targets.ForEach(t => Console.WriteLine($"\tTarget: {t.Arn} Id: {t.Id}
Input: {t.Input}"));
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// List all of the rules for a particular target.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic.</param>
    /// <returns>Async task.</returns>
    private static async Task ListRulesForTarget(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine("List all of the rules for a particular target.");

        var rules = await _eventBridgeWrapper.ListAllRuleNamesByTarget(topicArn);
        rules.ForEach(r => Console.WriteLine($"\tRule: {r}"));

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Enable or disable a particular rule.
    /// </summary>
    /// <param name="isEnabled">True to enable the rule, otherwise false.</param>
    /// <returns>Async task.</returns>
    private static async Task ChangeRuleState(bool isEnabled)
    {
        Console.WriteLine(new string('-', 80));
        var eventRuleName = _configuration["eventRuleName"];

        if (!isEnabled)
        {
            Console.WriteLine($"Disabling the rule: {eventRuleName}");
            await _eventBridgeWrapper.DisableRuleByName(eventRuleName);
        }
        else
        {
            Console.WriteLine($"Enabling the rule: {eventRuleName}");
            await _eventBridgeWrapper.EnableRuleByName(eventRuleName);
        }

        Console.WriteLine(new string('-', 80));
    }
```

```
    /// <summary>
    /// Get the current state of the rule.
    /// </summary>
    /// <returns>Async task.</returns>
    private static async Task GetRuleState()
    {
        Console.WriteLine(new string('-', 80));
        var eventRuleName = _configuration["eventRuleName"];

        var state = await
_eventBridgeWrapper.GetRuleStateByRuleName(eventRuleName);
        Console.WriteLine($"Rule {eventRuleName} is in current state {state}.");

        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Clean up the resources from the scenario.
    /// </summary>
    /// <param name="topicArn">The ARN of the SNS topic to clean up.</param>
    /// <returns>Async task.</returns>
    private static async Task CleanupResources(string topicArn)
    {
        Console.WriteLine(new string('-', 80));
        Console.WriteLine($"Clean up resources.");

        var eventRuleName = _configuration["eventRuleName"];
        if (GetYesNoResponse($"\tDelete all targets and event rule
{eventRuleName}? (y/n)"))
        {
            Console.WriteLine($"\tRemoving all targets from the event rule.");
            await _eventBridgeWrapper.RemoveAllTargetsFromRule(eventRuleName);

            Console.WriteLine($"\tDeleting event rule.");
            await _eventBridgeWrapper.DeleteRuleByName(eventRuleName);
        }

        var topicName = _configuration["topicName"];
        if (GetYesNoResponse($"\tDelete Amazon SNS subscription topic
{topicName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting topic.");
            await _snsClient!.DeleteTopicAsync(new DeleteTopicRequest()
            {
```

```
                TopicArn = topicArn
            });
        }

        var bucketName = _configuration["testBucketName"];
        if (GetYesNoResponse($"\tDelete Amazon S3 bucket {bucketName}? (y/n)"))
        {
            Console.WriteLine($"\tDeleting bucket.");
            // Delete all objects in the bucket.
            var deleteList = await _s3Client.ListObjectsV2Async(new
 ListObjectsV2Request()
            {
                BucketName = bucketName
            });
            await _s3Client.DeleteObjectsAsync(new DeleteObjectsRequest()
            {
                BucketName = bucketName,
                Objects = deleteList.S3Objects
                    .Select(o => new KeyVersion { Key = o.Key }).ToList()
            });
            // Now delete the bucket.
            await _s3Client.DeleteBucketAsync(new DeleteBucketRequest()
            {
                BucketName = bucketName
            });
        }

        var roleName = _configuration["roleName"];
        if (GetYesNoResponse($"\tDelete role {roleName}? (y/n)"))
        {
            Console.WriteLine($"\tDetaching policy and deleting role.");

            await _iamClient!.DetachRolePolicyAsync(new DetachRolePolicyRequest()
            {
                RoleName = roleName,
                PolicyArn = "arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess",
            });

            await _iamClient!.DeleteRoleAsync(new DeleteRoleRequest()
            {
                RoleName = roleName
            });
        }
```

```
        Console.WriteLine(new string('-', 80));
    }

    /// <summary>
    /// Helper method to get a yes or no response from the user.
    /// </summary>
    /// <param name="question">The question string to print on the console.</
param>
    /// <returns>True if the user responds with a yes.</returns>
    private static bool GetYesNoResponse(string question)
    {
        Console.WriteLine(question);
        var ynResponse = Console.ReadLine();
        var response = ynResponse != null &&
                       ynResponse.Equals("y",
                           StringComparison.InvariantCultureIgnoreCase);
        return response;
    }
}
```

Créez une classe qui englobe les EventBridge opérations.

```
/// <summary>
/// Wrapper for Amazon EventBridge operations.
/// </summary>
public class EventBridgeWrapper
{
    private readonly IAmazonEventBridge _amazonEventBridge;
    private readonly ILogger<EventBridgeWrapper> _logger;

    /// <summary>
    /// Constructor for the EventBridge wrapper.
    /// </summary>
    /// <param name="amazonEventBridge">The injected EventBridge client.</param>
    /// <param name="logger">The injected logger for the wrapper.</param>
    public EventBridgeWrapper(IAmazonEventBridge amazonEventBridge,
 ILogger<EventBridgeWrapper> logger)

    {
        _amazonEventBridge = amazonEventBridge;
```

```
        _logger = logger;
    }

    /// <summary>
    /// Get the state for a rule by the rule name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="eventBusName">The optional name of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The state of the rule.</returns>
    public async Task<RuleState> GetRuleStateByRuleName(string ruleName, string?
eventBusName = null)
    {
        var ruleResponse = await _amazonEventBridge.DescribeRuleAsync(
            new DescribeRuleRequest()
            {
                Name = ruleName,
                EventBusName = eventBusName
            });
        return ruleResponse.State;
    }

    /// <summary>
    /// Enable a particular rule on an event bus.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> EnableRuleByName(string ruleName)
    {
        var ruleResponse = await _amazonEventBridge.EnableRuleAsync(
            new EnableRuleRequest()
            {
                Name = ruleName
            });
        return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Disable a particular rule on an event bus.
    /// </summary
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DisableRuleByName(string ruleName)
    {
```

```
        var ruleResponse = await _amazonEventBridge.DisableRuleAsync(
            new DisableRuleRequest()
            {
                Name = ruleName
            });
        return ruleResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// List the rules on an event bus.
    /// </summary>
    /// <param name="eventBusArn">The optional ARN of the event bus. If empty,
uses the default event bus.</param>
    /// <returns>The list of rules.</returns>
    public async Task<List<Rule>> ListAllRulesForEventBus(string? eventBusArn =
null)
    {
        var results = new List<Rule>();
        var request = new ListRulesRequest()
        {
            EventBusName = eventBusArn
        };
        // Get all of the pages of rules.
        ListRulesResponse response;
        do
        {
            response = await _amazonEventBridge.ListRulesAsync(request);
            results.AddRange(response.Rules);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// List all of the targets matching a rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <returns>The list of targets.</returns>
    public async Task<List<Target>> ListAllTargetsOnRule(string ruleName)
    {
        var results = new List<Target>();
        var request = new ListTargetsByRuleRequest()
```

```
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse response;
        do
        {
            response = await _amazonEventBridge.ListTargetsByRuleAsync(request);
            results.AddRange(response.Targets);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// List names of all rules matching a target.
    /// </summary>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <returns>The list of rule names.</returns>
    public async Task<List<string>> ListAllRuleNamesByTarget(string targetArn)
    {
        var results = new List<string>();
        var request = new ListRuleNamesByTargetRequest()
        {
            TargetArn = targetArn
        };
        ListRuleNamesByTargetResponse response;
        do
        {
            response = await
_amazonEventBridge.ListRuleNamesByTargetAsync(request);
            results.AddRange(response.RuleNames);
            request.NextToken = response.NextToken;

        } while (response.NextToken is not null);

        return results;
    }

    /// <summary>
    /// Create a new event rule that triggers when an Amazon S3 object is created
in a bucket.
    /// </summary>
```

```
    /// <param name="roleArn">The ARN of the role.</param>
    /// <param name="ruleName">The name to give the rule.</param>
    /// <param name="bucketName">The name of the bucket to trigger the event.</
param>
    /// <returns>The ARN of the new rule.</returns>
    public async Task<string> PutS3UploadRule(string roleArn, string ruleName,
 string bucketName)
    {
        string eventPattern = "{" +
                              "\"source\": [\"aws.s3\"]," +
                                  "\"detail-type\": [\"Object Created\"]," +
                                  "\"detail\": {" +
                                      "\"bucket\": {" +
                                          "\"name\": [\"" + bucketName + "\"]"
 +
                                      "}" +
                                  "}" +
                              "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Example S3 upload rule for EventBridge",
                RoleArn = roleArn,
                EventPattern = eventPattern
            });

        return response.RuleArn;
    }

    /// <summary>
    /// Update an Amazon S3 object created rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
 default event bus.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateS3UploadRuleTargetWithTransform(string
 ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();
```

```
        var targets = new List<Target>
        {
            new Target()
            {
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputPathsMap = new Dictionary<string, string>()
                    {
                        {"bucket", "$.detail.bucket.name"},
                        {"time", "$.time"}
                    },
                    InputTemplate = "\"Notification: an object was uploaded to
bucket <bucket> at <time>.\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
        }
        return targetID;
    }

    /// <summary>
    /// Update a custom rule with a transform on the target.
    /// </summary>
    /// <param name="ruleName">The name of the rule.</param>
    /// <param name="targetArn">The ARN of the target.</param>
    /// <param name="eventBusArn">Optional event bus ARN. If empty, uses the
default event bus.</param>
```

```
    /// <returns>The ID of the target.</returns>
    public async Task<string> UpdateCustomRuleTargetWithTransform(string
 ruleName, string targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        var targets = new List<Target>
        {
            new Target()
            {
                Id = targetID,
                Arn = targetArn,
                InputTransformer = new InputTransformer()
                {
                    InputTemplate = "\"Notification: sample event was received.
\""
                }
            }
        };
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });
        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
 {e.ErrorCode}");
            });
        }
        return targetID;
    }

    /// <summary>
    /// Add an event to the event bus that includes an email, message, and time.
    /// </summary>
    /// <param name="email">The email to use in the event detail of the custom
 event.</param>
    /// <returns>True if successful.</returns>
```

```
    public async Task<bool> PutCustomEmailEvent(string email)
    {
        var eventDetail = new
        {
            UserEmail = email,
            Message = "This event was generated by example code.",
            UtcTime = DateTime.UtcNow.ToString("g")
        };
        var response = await _amazonEventBridge.PutEventsAsync(
            new PutEventsRequest()
            {
                Entries = new List<PutEventsRequestEntry>()
                {
                    new PutEventsRequestEntry()
                    {
                        Source = "ExampleSource",
                        Detail = JsonSerializer.Serialize(eventDetail),
                        DetailType = "ExampleType"
                    }
                }
            });

        return response.FailedEntryCount == 0;
    }

    /// <summary>
    /// Update a rule to use a custom defined event pattern.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <returns>The ARN of the updated rule.</returns>
    public async Task<string> UpdateCustomEventPattern(string ruleName)
    {
        string customEventsPattern = "{" +
                                     "\"source\": [\"ExampleSource\"]," +
                                     "\"detail-type\": [\"ExampleType\"]" +
                                     "}";

        var response = await _amazonEventBridge.PutRuleAsync(
            new PutRuleRequest()
            {
                Name = ruleName,
                Description = "Custom test rule",
                EventPattern = customEventsPattern
            });
```

```
        return response.RuleArn;
    }

    /// <summary>
    /// Add an Amazon SNS target topic to a rule.
    /// </summary>
    /// <param name="ruleName">The name of the rule to update.</param>
    /// <param name="targetArn">The ARN of the Amazon SNS target.</param>
    /// <param name="eventBusArn">The optional event bus name, uses default if
empty.</param>
    /// <returns>The ID of the target.</returns>
    public async Task<string> AddSnsTargetToRule(string ruleName, string
targetArn, string? eventBusArn = null)
    {
        var targetID = Guid.NewGuid().ToString();

        // Create the list of targets and add a new target.
        var targets = new List<Target>
        {
            new Target()
            {
                Arn = targetArn,
                Id = targetID
            }
        };

        // Add the targets to the rule.
        var response = await _amazonEventBridge.PutTargetsAsync(
            new PutTargetsRequest()
            {
                EventBusName = eventBusArn,
                Rule = ruleName,
                Targets = targets,
            });

        if (response.FailedEntryCount > 0)
        {
            response.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to add target {e.TargetId}: {e.ErrorMessage}, code
{e.ErrorCode}");
            });
```

```
        }

        return targetID;
    }

    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> RemoveAllTargetsFromRule(string ruleName)
    {
        var targetIds = new List<string>();
        var request = new ListTargetsByRuleRequest()
        {
            Rule = ruleName
        };
        ListTargetsByRuleResponse targetsResponse;
        do
        {
            targetsResponse = await
_amazonEventBridge.ListTargetsByRuleAsync(request);
            targetIds.AddRange(targetsResponse.Targets.Select(t => t.Id));
            request.NextToken = targetsResponse.NextToken;

        } while (targetsResponse.NextToken is not null);

        var removeResponse = await _amazonEventBridge.RemoveTargetsAsync(
            new RemoveTargetsRequest()
            {
                Rule = ruleName,
                Ids = targetIds
            });

        if (removeResponse.FailedEntryCount > 0)
        {
            removeResponse.FailedEntries.ForEach(e =>
            {
                _logger.LogError(
                    $"Failed to remove target {e.TargetId}: {e.ErrorMessage},
code {e.ErrorCode}");
            });
        }
```

```
        return removeResponse.HttpStatusCode == HttpStatusCode.OK;
    }

    /// <summary>
    /// Delete an event rule by name.
    /// </summary>
    /// <param name="ruleName">The name of the event rule.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteRuleByName(string ruleName)
    {
        var response = await _amazonEventBridge.DeleteRuleAsync(
            new DeleteRuleRequest()
            {
                Name = ruleName
            });

        return response.HttpStatusCode == HttpStatusCode.OK;
    }
}
```

- Pour plus d'informations sur l'API consultez les rubriques suivantes dans la référence de l'API AWS SDK for .NET .
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Java

SDK pour Java 2.x

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This Java code example performs the following tasks:
 *
 * This Java V2 example performs the following tasks with Amazon EventBridge:
 *
 * 1. Creates an AWS Identity and Access Management (IAM) role to use with
 * Amazon EventBridge.
 * 2. Amazon Simple Storage Service (Amazon S3) bucket with EventBridge events
 * enabled.
 * 3. Creates a rule that triggers when an object is uploaded to Amazon S3.
 * 4. Lists rules on the event bus.
 * 5. Creates a new Amazon Simple Notification Service (Amazon SNS) topic and
 * lets the user subscribe to it.
 * 6. Adds a target to the rule that sends an email to the specified topic.
 * 7. Creates an EventBridge event that sends an email when an Amazon S3 object
 * is created.
 * 8. Lists Targets.
 * 9. Lists the rules for the same target.
 * 10. Triggers the rule by uploading a file to the Amazon S3 bucket.
 * 11. Disables a specific rule.
 * 12. Checks and print the state of the rule.
 * 13. Adds a transform to the rule to change the text of the email.
 * 14. Enables a specific rule.
 * 15. Triggers the updated rule by uploading a file to the Amazon S3 bucket.
```

```
 * 16. Updates the rule to be a custom rule pattern.
 * 17. Sending an event to trigger the rule.
 * 18. Cleans up resources.
 *
 */
public class EventbridgeMVP {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws InterruptedException,
 IOException {
        final String usage = """

                Usage:
                    <roleName> <bucketName> <topicName> <eventRuleName>

                Where:
                    roleName - The name of the role to create.
                    bucketName - The Amazon Simple Storage Service (Amazon S3)
 bucket name to create.
                    topicName - The name of the Amazon Simple Notification
 Service (Amazon SNS) topic to create.
                    eventRuleName - The Amazon EventBridge rule name to create.
                """;

        if (args.length != 5) {
            System.out.println(usage);
            System.exit(1);
        }

        String polJSON = "{" +
                "\"Version\": \"2012-10-17\"," +
                "\"Statement\": [{" +
                "\"Effect\": \"Allow\"," +
                "\"Principal\": {" +
                "\"Service\": \"events.amazonaws.com\"" +
                "}," +
                "\"Action\": \"sts:AssumeRole\"" +
                "}]" +
                "}";

        Scanner sc = new Scanner(System.in);
        String roleName = args[0];
        String bucketName = args[1];
```

```
        String topicName = args[2];
        String eventRuleName = args[3];

        Region region = Region.US_EAST_1;
        EventBridgeClient eventBrClient = EventBridgeClient.builder()
                .region(region)
                .build();

        S3Client s3Client = S3Client.builder()
                .region(region)
                .build();

        Region regionGl = Region.AWS_GLOBAL;
        IamClient iam = IamClient.builder()
                .region(regionGl)
                .build();

        SnsClient snsClient = SnsClient.builder()
                .region(region)
                .build();

        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon EventBridge example
scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out
                .println("1. Create an AWS Identity and Access Management (IAM)
role to use with Amazon EventBridge.");
        String roleArn = createIAMRole(iam, roleName, polJSON);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create an S3 bucket with EventBridge events
enabled.");
        if (checkBucket(s3Client, bucketName)) {
            System.out.println("Bucket " + bucketName + " already exists. Ending
this scenario.");
            System.exit(1);
        }

        createBucket(s3Client, bucketName);
        Thread.sleep(3000);
```

```
        setBucketNotification(s3Client, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Create a rule that triggers when an object is
uploaded to Amazon S3.");
        Thread.sleep(10000);
        addEventRule(eventBrClient, roleArn, bucketName, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. List rules on the event bus.");
        listRules(eventBrClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Create a new SNS topic for testing and let the
user subscribe to the topic.");
        String topicArn = createSnsTopic(snsClient, topicName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Add a target to the rule that sends an email to
the specified topic.");
        System.out.println("Enter your email to subscribe to the Amazon SNS
topic:");
        String email = sc.nextLine();
        subEmail(snsClient, topicArn, email);
        System.out.println(
                "Use the link in the email you received to confirm your
subscription. Then, press Enter to continue.");
        sc.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create an EventBridge event that sends an email
when an Amazon S3 object is created.");
        addSnsEventRule(eventBrClient, eventRuleName, topicArn, topicName,
eventRuleName, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 8. List Targets.");
        listTargets(eventBrClient, eventRuleName);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 9. List the rules for the same target.");
        listTargetRules(eventBrClient, topicArn);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 10. Trigger the rule by uploading a file to the S3
bucket.");
        System.out.println("Press Enter to continue.");
        sc.nextLine();
        uploadTextFiletoS3(s3Client, bucketName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Disable a specific rule.");
        changeRuleState(eventBrClient, eventRuleName, false);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Check and print the state of the rule.");
        checkRule(eventBrClient, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Add a transform to the rule to change the text of
the email.");
        updateSnsEventRule(eventBrClient, topicArn, eventRuleName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Enable a specific rule.");
        changeRuleState(eventBrClient, eventRuleName, true);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(" 15. Trigger the updated rule by uploading a file to
the S3 bucket.");
        System.out.println("Press Enter to continue.");
        sc.nextLine();
        uploadTextFiletoS3(s3Client, bucketName);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println(" 16. Update the rule to be a custom rule pattern.");
        updateToCustomRule(eventBrClient, eventRuleName);
        System.out.println("Updated event rule " + eventRuleName + " to use a
custom pattern.");
        updateCustomRuleTargetWithTransform(eventBrClient, topicArn,
eventRuleName);
        System.out.println("Updated event target " + topicArn + ".");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("17. Sending an event to trigger the rule. This will
trigger a subscription email.");
        triggerCustomRule(eventBrClient, email);
        System.out.println("Events have been sent. Press Enter to continue.");
        sc.nextLine();
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("18. Clean up resources.");
        System.out.println("Do you want to clean up resources (y/n)");
        String ans = sc.nextLine();
        if (ans.compareTo("y") == 0) {
            cleanupResources(eventBrClient, snsClient, s3Client, iam, topicArn,
eventRuleName, bucketName, roleName);
        } else {
            System.out.println("The resources will not be cleaned up. ");
        }
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Amazon EventBridge example scenario has
successfully completed.");
        System.out.println(DASHES);
    }

    public static void cleanupResources(EventBridgeClient eventBrClient,
SnsClient snsClient, S3Client s3Client,
            IamClient iam, String topicArn, String eventRuleName, String
bucketName, String roleName) {
        System.out.println("Removing all targets from the event rule.");
        deleteTargetsFromRule(eventBrClient, eventRuleName);
        deleteRuleByName(eventBrClient, eventRuleName);
        deleteSNSTopic(snsClient, topicArn);
```

```
        deleteS3Bucket(s3Client, bucketName);
        deleteRole(iam, roleName);
    }

    public static void deleteRole(IamClient iam, String roleName) {
        String policyArn = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess";
        DetachRolePolicyRequest policyRequest = DetachRolePolicyRequest.builder()
                .policyArn(policyArn)
                .roleName(roleName)
                .build();

        iam.detachRolePolicy(policyRequest);
        System.out.println("Successfully detached policy " + policyArn + " from
role " + roleName);

        // Delete the role.
        DeleteRoleRequest roleRequest = DeleteRoleRequest.builder()
                .roleName(roleName)
                .build();

        iam.deleteRole(roleRequest);
        System.out.println("*** Successfully deleted " + roleName);
    }

    public static void deleteS3Bucket(S3Client s3Client, String bucketName) {
        // Remove all the objects from the S3 bucket.
        ListObjectsRequest listObjects = ListObjectsRequest.builder()
                .bucket(bucketName)
                .build();

        ListObjectsResponse res = s3Client.listObjects(listObjects);
        List<S3Object> objects = res.contents();
        ArrayList<ObjectIdentifier> toDelete = new ArrayList<>();

        for (S3Object myValue : objects) {
            toDelete.add(ObjectIdentifier.builder()
                    .key(myValue.key())
                    .build());
        }

        DeleteObjectsRequest dor = DeleteObjectsRequest.builder()
                .bucket(bucketName)
                .delete(Delete.builder()
                        .objects(toDelete).build())
```

```
                .build();

        s3Client.deleteObjects(dor);

        // Delete the S3 bucket.
        DeleteBucketRequest deleteBucketRequest = DeleteBucketRequest.builder()
                .bucket(bucketName)
                .build();

        s3Client.deleteBucket(deleteBucketRequest);
        System.out.println("You have deleted the bucket and the objects");
    }

    // Delete the SNS topic.
    public static void deleteSNSTopic(SnsClient snsClient, String topicArn) {
        try {
            DeleteTopicRequest request = DeleteTopicRequest.builder()
                    .topicArn(topicArn)
                    .build();

            DeleteTopicResponse result = snsClient.deleteTopic(request);
            System.out.println("\n\nStatus was " +
result.sdkHttpResponse().statusCode());

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

    public static void deleteRuleByName(EventBridgeClient eventBrClient, String
ruleName) {
        DeleteRuleRequest ruleRequest = DeleteRuleRequest.builder()
                .name(ruleName)
                .build();

        eventBrClient.deleteRule(ruleRequest);
        System.out.println("Successfully deleted the rule");
    }

    public static void deleteTargetsFromRule(EventBridgeClient eventBrClient,
String eventRuleName) {
        // First, get all targets that will be deleted.
        ListTargetsByRuleRequest request = ListTargetsByRuleRequest.builder()
```

```
                .rule(eventRuleName)
                .build();

        ListTargetsByRuleResponse response =
eventBrClient.listTargetsByRule(request);
        List<Target> allTargets = response.targets();

        // Get all targets and delete them.
        for (Target myTarget : allTargets) {
            RemoveTargetsRequest removeTargetsRequest =
RemoveTargetsRequest.builder()
                    .rule(eventRuleName)
                    .ids(myTarget.id())
                    .build();

            eventBrClient.removeTargets(removeTargetsRequest);
            System.out.println("Successfully removed the target");
        }
    }

    public static void triggerCustomRule(EventBridgeClient eventBrClient, String
email) {
        String json = "{" +
                "\"UserEmail\": \"" + email + "\"," +
                "\"Message\": \"This event was generated by example code.\"," +
                "\"UtcTime\": \"Now.\"" +
                "}";

        PutEventsRequestEntry entry = PutEventsRequestEntry.builder()
                .source("ExampleSource")
                .detail(json)
                .detailType("ExampleType")
                .build();

        PutEventsRequest eventsRequest = PutEventsRequest.builder()
                .entries(entry)
                .build();

        eventBrClient.putEvents(eventsRequest);
    }

    public static void updateCustomRuleTargetWithTransform(EventBridgeClient
eventBrClient, String topicArn,
            String ruleName) {
```

```
        String targetId = java.util.UUID.randomUUID().toString();
        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: sample event was received.\"")
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);
        } catch (EventBridgeException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void updateToCustomRule(EventBridgeClient eventBrClient, String
ruleName) {
        String customEventsPattern = "{" +
                "\"source\": [\"ExampleSource\"]," +
                "\"detail-type\": [\"ExampleType\"]" +
                "}";

        PutRuleRequest request = PutRuleRequest.builder()
                .name(ruleName)
                .description("Custom test rule")
                .eventPattern(customEventsPattern)
                .build();

        eventBrClient.putRule(request);
    }

    // Update an Amazon S3 object created rule with a transform on the target.
    public static void updateSnsEventRule(EventBridgeClient eventBrClient, String
topicArn, String ruleName) {
```

```
        String targetId = java.util.UUID.randomUUID().toString();
        Map<String, String> myMap = new HashMap<>();
        myMap.put("bucket", "$.detail.bucket.name");
        myMap.put("time", "$.time");

        InputTransformer inputTransformer = InputTransformer.builder()
                .inputTemplate("\"Notification: an object was uploaded to bucket
<bucket> at <time>.\"")
                .inputPathsMap(myMap)
                .build();

        Target target = Target.builder()
                .id(targetId)
                .arn(topicArn)
                .inputTransformer(inputTransformer)
                .build();

        try {
            PutTargetsRequest targetsRequest = PutTargetsRequest.builder()
                    .rule(ruleName)
                    .targets(target)
                    .eventBusName(null)
                    .build();

            eventBrClient.putTargets(targetsRequest);

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

    public static void checkRule(EventBridgeClient eventBrClient, String
eventRuleName) {
        try {
            DescribeRuleRequest ruleRequest = DescribeRuleRequest.builder()
                    .name(eventRuleName)
                    .build();

            DescribeRuleResponse response =
eventBrClient.describeRule(ruleRequest);
            System.out.println("The state of the rule is " +
response.stateAsString());
```

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

    public static void changeRuleState(EventBridgeClient eventBrClient, String
eventRuleName, Boolean isEnabled) {
        try {
            if (!isEnabled) {
                System.out.println("Disabling the rule: " + eventRuleName);
                DisableRuleRequest ruleRequest = DisableRuleRequest.builder()
                        .name(eventRuleName)
                        .build();

                eventBrClient.disableRule(ruleRequest);
            } else {
                System.out.println("Enabling the rule: " + eventRuleName);
                EnableRuleRequest ruleRequest = EnableRuleRequest.builder()
                        .name(eventRuleName)
                        .build();
                eventBrClient.enableRule(ruleRequest);
            }

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

    // Create and upload a file to an S3 bucket to trigger an event.
    public static void uploadTextFiletoS3(S3Client s3Client, String bucketName)
throws IOException {
        // Create a unique file name.
        String fileSuffix = new SimpleDateFormat("yyyyMMddHHmmss").format(new
Date());
        String fileName = "TextFile" + fileSuffix + ".txt";

        File myFile = new File(fileName);
        FileWriter fw = new FileWriter(myFile.getAbsoluteFile());
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write("This is a sample file for testing uploads.");
        bw.close();
```

```
        try {
            PutObjectRequest putOb = PutObjectRequest.builder()
                    .bucket(bucketName)
                    .key(fileName)
                    .build();

            s3Client.putObject(putOb, RequestBody.fromFile(myFile));

        } catch (S3Exception e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    public static void listTargetRules(EventBridgeClient eventBrClient, String
topicArn) {
        ListRuleNamesByTargetRequest ruleNamesByTargetRequest =
ListRuleNamesByTargetRequest.builder()
                .targetArn(topicArn)
                .build();

        ListRuleNamesByTargetResponse response =
eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest);
        List<String> rules = response.ruleNames();
        for (String rule : rules) {
            System.out.println("The rule name is " + rule);
        }
    }

    public static void listTargets(EventBridgeClient eventBrClient, String
ruleName) {
        ListTargetsByRuleRequest ruleRequest = ListTargetsByRuleRequest.builder()
                .rule(ruleName)
                .build();

        ListTargetsByRuleResponse res =
eventBrClient.listTargetsByRule(ruleRequest);
        List<Target> targetsList = res.targets();
        for (Target target: targetsList) {
            System.out.println("Target ARN: "+target.arn());
        }
    }

    // Add a rule which triggers an SNS target when a file is uploaded to an S3
```

```
   // bucket.
   public static void addSnsEventRule(EventBridgeClient eventBrClient, String
ruleName, String topicArn,
           String topicName, String eventRuleName, String bucketName) {
       String targetID = java.util.UUID.randomUUID().toString();
       Target myTarget = Target.builder()
               .id(targetID)
               .arn(topicArn)
               .build();

       List<Target> targets = new ArrayList<>();
       targets.add(myTarget);
       PutTargetsRequest request = PutTargetsRequest.builder()
               .eventBusName(null)
               .targets(targets)
               .rule(ruleName)
               .build();

       eventBrClient.putTargets(request);
       System.out.println("Added event rule " + eventRuleName + " with Amazon
SNS target " + topicName + " for bucket "
               + bucketName + ".");
   }

   public static void subEmail(SnsClient snsClient, String topicArn, String
email) {
       try {
           SubscribeRequest request = SubscribeRequest.builder()
                   .protocol("email")
                   .endpoint(email)
                   .returnSubscriptionArn(true)
                   .topicArn(topicArn)
                   .build();

           SubscribeResponse result = snsClient.subscribe(request);
           System.out.println("Subscription ARN: " + result.subscriptionArn() +
"\n\n Status is "
                   + result.sdkHttpResponse().statusCode());

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

```
   public static void listRules(EventBridgeClient eventBrClient) {
       try {
           ListRulesRequest rulesRequest = ListRulesRequest.builder()
                   .eventBusName("default")
                   .limit(10)
                   .build();

           ListRulesResponse response = eventBrClient.listRules(rulesRequest);
           List<Rule> rules = response.rules();
           for (Rule rule : rules) {
               System.out.println("The rule name is : " + rule.name());
               System.out.println("The rule description is : " +
rule.description());
               System.out.println("The rule state is : " +
rule.stateAsString());
           }

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

   public static String createSnsTopic(SnsClient snsClient, String topicName) {
       String topicPolicy = "{" +
               "\"Version\": \"2012-10-17\"," +
               "\"Statement\": [{" +
               "\"Sid\": \"EventBridgePublishTopic\"," +
               "\"Effect\": \"Allow\"," +
               "\"Principal\": {" +
               "\"Service\": \"events.amazonaws.com\"" +
               "}," +
               "\"Resource\": \"*\"," +
               "\"Action\": \"sns:Publish\"" +
               "}]" +
               "}";

       Map<String, String> topicAttributes = new HashMap<>();
       topicAttributes.put("Policy", topicPolicy);
       CreateTopicRequest topicRequest = CreateTopicRequest.builder()
               .name(topicName)
               .attributes(topicAttributes)
               .build();
```

```
        CreateTopicResponse response = snsClient.createTopic(topicRequest);
        System.out.println("Added topic " + topicName + " for email
subscriptions.");
        return response.topicArn();
    }

    // Create a new event rule that triggers when an Amazon S3 object is created
in
    // a bucket.
    public static void addEventRule(EventBridgeClient eventBrClient, String
roleArn, String bucketName,
            String eventRuleName) {
        String pattern = "{\n" +
                "  \"source\": [\"aws.s3\"],\n" +
                "  \"detail-type\": [\"Object Created\"],\n" +
                "  \"detail\": {\n" +
                "    \"bucket\": {\n" +
                "      \"name\": [\"" + bucketName + "\"]\n" +
                "    }\n" +
                "  }\n" +
                "}";

        try {
            PutRuleRequest ruleRequest = PutRuleRequest.builder()
                    .description("Created by using the AWS SDK for Java v2")
                    .name(eventRuleName)
                    .eventPattern(pattern)
                    .roleArn(roleArn)
                    .build();

            PutRuleResponse ruleResponse = eventBrClient.putRule(ruleRequest);
            System.out.println("The ARN of the new rule is " +
ruleResponse.ruleArn());

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

    // Determine if the S3 bucket exists.
    public static Boolean checkBucket(S3Client s3Client, String bucketName) {
        try {
```

```
            HeadBucketRequest headBucketRequest = HeadBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            s3Client.headBucket(headBucketRequest);
            return true;
        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
        }
        return false;
    }

    // Set the S3 bucket notification configuration.
    public static void setBucketNotification(S3Client s3Client, String
bucketName) {
        try {
            EventBridgeConfiguration eventBridgeConfiguration =
EventBridgeConfiguration.builder()
                    .build();

            NotificationConfiguration configuration =
NotificationConfiguration.builder()
                    .eventBridgeConfiguration(eventBridgeConfiguration)
                    .build();

            PutBucketNotificationConfigurationRequest configurationRequest =
PutBucketNotificationConfigurationRequest
                    .builder()
                    .bucket(bucketName)
                    .notificationConfiguration(configuration)
                    .skipDestinationValidation(true)
                    .build();

            s3Client.putBucketNotificationConfiguration(configurationRequest);
            System.out.println("Added bucket " + bucketName + " with EventBridge
events enabled.");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static void createBucket(S3Client s3Client, String bucketName) {
```

```
        try {
            S3Waiter s3Waiter = s3Client.waiter();
            CreateBucketRequest bucketRequest = CreateBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            s3Client.createBucket(bucketRequest);
            HeadBucketRequest bucketRequestWait = HeadBucketRequest.builder()
                    .bucket(bucketName)
                    .build();

            // Wait until the bucket is created and print out the response.
            WaiterResponse<HeadBucketResponse> waiterResponse =
 s3Waiter.waitUntilBucketExists(bucketRequestWait);
            waiterResponse.matched().response().ifPresent(System.out::println);
            System.out.println(bucketName + " is ready");

        } catch (S3Exception e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
    }

    public static String createIAMRole(IamClient iam, String rolename, String
 polJSON) {
        try {
            CreateRoleRequest request = CreateRoleRequest.builder()
                    .roleName(rolename)
                    .assumeRolePolicyDocument(polJSON)
                    .description("Created using the AWS SDK for Java")
                    .build();

            CreateRoleResponse response = iam.createRole(request);
            AttachRolePolicyRequest rolePolicyRequest =
 AttachRolePolicyRequest.builder()
                    .roleName(rolename)
                    .policyArn("arn:aws:iam::aws:policy/
AmazonEventBridgeFullAccess")
                    .build();

            iam.attachRolePolicy(rolePolicyRequest);
            return response.role().arn();

        } catch (IamException e) {
```

```
            System.err.println(e.awsErrorDetails().errorMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans la référence de l'API AWS SDK for Java 2.x .
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Kotlin

SDK pour Kotlin

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

```
/*
 Before running this Kotlin code example, set up your development environment,
 including your credentials.

 For more information, see the following documentation topic:
 https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
```

```
 This Kotlin example performs the following tasks with Amazon EventBridge:

 1. Creates an AWS Identity and Access Management (IAM) role to use with Amazon
 EventBridge.
 2. Creates an Amazon Simple Storage Service (Amazon S3) bucket with EventBridge
 events enabled.
 3. Creates a rule that triggers when an object is uploaded to Amazon S3.
 4. Lists rules on the event bus.
 5. Creates a new Amazon Simple Notification Service (Amazon SNS) topic and lets
 the user subscribe to it.
 6. Adds a target to the rule that sends an email to the specified topic.
 7. Creates an EventBridge event that sends an email when an Amazon S3 object is
 created.
 8. Lists targets.
 9. Lists the rules for the same target.
 10. Triggers the rule by uploading a file to the S3 bucket.
 11. Disables a specific rule.
 12. Checks and prints the state of the rule.
 13. Adds a transform to the rule to change the text of the email.
 14. Enables a specific rule.
 15. Triggers the updated rule by uploading a file to the S3 bucket.
 16. Updates the rule to a custom rule pattern.
 17. Sends an event to trigger the rule.
 18. Cleans up resources.
*/
val DASHES: String = String(CharArray(80)).replace("\u0000", "-")
suspend fun main(args: Array<String>) {
    val usage = """
    Usage:
        <roleName> <bucketName> <topicName> <eventRuleName>

    Where:
        roleName - The name of the role to create.
        bucketName - The Amazon Simple Storage Service (Amazon S3) bucket name to
 create.
        topicName - The name of the Amazon Simple Notification Service (Amazon
 SNS) topic to create.
        eventRuleName - The Amazon EventBridge rule name to create.
    """
    val polJSON = "{" +
        "\"Version\": \"2012-10-17\"," +
        "\"Statement\": [{" +
        "\"Effect\": \"Allow\"," +
```

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

    if (args.size != 4) {
        println(usage)
        exitProcess(1)
    }

    val sc = Scanner(System.`in`)
    val roleName = args[0]
    val bucketName = args[1]
    val topicName = args[2]
    val eventRuleName = args[3]

    println(DASHES)
    println("Welcome to the Amazon EventBridge example scenario.")
    println(DASHES)

    println(DASHES)
    println("1. Create an AWS Identity and Access Management (IAM) role to use
with Amazon EventBridge.")
    val roleArn = createIAMRole(roleName, polJSON)
    println(DASHES)

    println(DASHES)
    println("2. Create an S3 bucket with EventBridge events enabled.")
    if (checkBucket(bucketName)) {
        println("$bucketName already exists. Ending this scenario.")
        exitProcess(1)
    }

    createBucket(bucketName)
    delay(3000)
    setBucketNotification(bucketName)
    println(DASHES)

    println(DASHES)
    println("3. Create a rule that triggers when an object is uploaded to Amazon
S3.")
    delay(10000)
```

```
    addEventRule(roleArn, bucketName, eventRuleName)
    println(DASHES)

    println(DASHES)
    println("4. List rules on the event bus.")
    listRules()
    println(DASHES)

    println(DASHES)
    println("5. Create a new SNS topic for testing and let the user subscribe to
the topic.")
    val topicArn = createSnsTopic(topicName)
    println(DASHES)

    println(DASHES)
    println("6. Add a target to the rule that sends an email to the specified
topic.")
    println("Enter your email to subscribe to the Amazon SNS topic:")
    val email = sc.nextLine()
    subEmail(topicArn, email)
    println("Use the link in the email you received to confirm your subscription.
Then press Enter to continue.")
    sc.nextLine()
    println(DASHES)

    println(DASHES)
    println("7. Create an EventBridge event that sends an email when an Amazon S3
object is created.")
    addSnsEventRule(eventRuleName, topicArn, topicName, eventRuleName,
bucketName)
    println(DASHES)

    println(DASHES)
    println("8. List targets.")
    listTargets(eventRuleName)
    println(DASHES)

    println(DASHES)
    println(" 9. List the rules for the same target.")
    listTargetRules(topicArn)
    println(DASHES)

    println(DASHES)
    println("10. Trigger the rule by uploading a file to the S3 bucket.")
```

```
    println("Press Enter to continue.")
    sc.nextLine()
    uploadTextFiletoS3(bucketName)
    println(DASHES)

    println(DASHES)
    println("11. Disable a specific rule.")
    changeRuleState(eventRuleName, false)
    println(DASHES)

    println(DASHES)
    println("12. Check and print the state of the rule.")
    checkRule(eventRuleName)
    println(DASHES)

    println(DASHES)
    println("13. Add a transform to the rule to change the text of the email.")
    updateSnsEventRule(topicArn, eventRuleName)
    println(DASHES)

    println(DASHES)
    println("14. Enable a specific rule.")
    changeRuleState(eventRuleName, true)
    println(DASHES)

    println(DASHES)
    println("15. Trigger the updated rule by uploading a file to the S3 bucket.")
    println("Press Enter to continue.")
    sc.nextLine()
    uploadTextFiletoS3(bucketName)
    println(DASHES)

    println(DASHES)
    println("16. Update the rule to a custom rule pattern.")
    updateToCustomRule(eventRuleName)
    println("Updated event rule $eventRuleName to use a custom pattern.")
    updateCustomRuleTargetWithTransform(topicArn, eventRuleName)
    println("Updated event target $topicArn.")
    println(DASHES)

    println(DASHES)
    println("17. Send an event to trigger the rule. This will trigger a
 subscription email.")
    triggerCustomRule(email)
```

```
    println("Events have been sent. Press Enter to continue.")
    sc.nextLine()
    println(DASHES)

    println(DASHES)
    println("18. Clean up resources.")
    println("Do you want to clean up resources (y/n)")
    val ans = sc.nextLine()
    if (ans.compareTo("y") == 0) {
        cleanupResources(topicArn, eventRuleName, bucketName, roleName)
    } else {
        println("The resources will not be cleaned up. ")
    }
    println(DASHES)

    println(DASHES)
    println("The Amazon EventBridge example scenario has successfully
 completed.")
    println(DASHES)
}

suspend fun cleanupResources(topicArn: String?, eventRuleName: String?,
 bucketName: String?, roleName: String?) {
    println("Removing all targets from the event rule.")
    deleteTargetsFromRule(eventRuleName)
    deleteRuleByName(eventRuleName)
    deleteSNSTopic(topicArn)
    deleteS3Bucket(bucketName)
    deleteRole(roleName)
}

suspend fun deleteRole(roleNameVal: String?) {
    val policyArnVal = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess"
    val policyRequest = DetachRolePolicyRequest {
        policyArn = policyArnVal
        roleName = roleNameVal
    }
    IamClient { region = "us-east-1" }.use { iam ->
        iam.detachRolePolicy(policyRequest)
        println("Successfully detached policy $policyArnVal from role
 $roleNameVal")

        // Delete the role.
        val roleRequest = DeleteRoleRequest {
```

```
            roleName = roleNameVal
        }

        iam.deleteRole(roleRequest)
        println("*** Successfully deleted $roleNameVal")
    }
}

suspend fun deleteS3Bucket(bucketName: String?) {
    // Remove all the objects from the S3 bucket.
    val listObjects = ListObjectsRequest {
        bucket = bucketName
    }
    S3Client { region = "us-east-1" }.use { s3Client ->
        val res = s3Client.listObjects(listObjects)
        val myObjects = res.contents
        val toDelete = mutableListOf<ObjectIdentifier>()

        if (myObjects != null) {
            for (myValue in myObjects) {
                toDelete.add(
                    ObjectIdentifier {
                        key = myValue.key
                    }
                )
            }
        }

        val delOb = Delete {
            objects = toDelete
        }

        val dor = DeleteObjectsRequest {
            bucket = bucketName
            delete = delOb
        }
        s3Client.deleteObjects(dor)

        // Delete the S3 bucket.
        val deleteBucketRequest = DeleteBucketRequest {
            bucket = bucketName
        }
        s3Client.deleteBucket(deleteBucketRequest)
        println("You have deleted the bucket and the objects")
```

```
    }
}

// Delete the SNS topic.
suspend fun deleteSNSTopic(topicArnVal: String?) {
    val request = DeleteTopicRequest {
        topicArn = topicArnVal
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        snsClient.deleteTopic(request)
        println(" $topicArnVal was deleted.")
    }
}

suspend fun deleteRuleByName(ruleName: String?) {
    val ruleRequest = DeleteRuleRequest {
        name = ruleName
    }
    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.deleteRule(ruleRequest)
        println("Successfully deleted the rule")
    }
}

suspend fun deleteTargetsFromRule(eventRuleName: String?) {
    // First, get all targets that will be deleted.
    val request = ListTargetsByRuleRequest {
        rule = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(request)
        val allTargets = response.targets

        // Get all targets and delete them.
        if (allTargets != null) {
            for (myTarget in allTargets) {
                val removeTargetsRequest = RemoveTargetsRequest {
                    rule = eventRuleName
                    ids = listOf(myTarget.id.toString())
                }
                eventBrClient.removeTargets(removeTargetsRequest)
                println("Successfully removed the target")
```

```
            }
        }
    }
}

suspend fun triggerCustomRule(email: String) {
    val json = "{" +
        "\"UserEmail\": \"" + email + "\"," +
        "\"Message\": \"This event was generated by example code.\"" +
        "\"UtcTime\": \"Now.\"" +
        "}"

    val entry = PutEventsRequestEntry {
        source = "ExampleSource"
        detail = json
        detailType = "ExampleType"
    }

    val eventsRequest = PutEventsRequest {
        this.entries = listOf(entry)
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putEvents(eventsRequest)
    }
}

suspend fun updateCustomRuleTargetWithTransform(topicArn: String?, ruleName:
 String?) {
    val targetId = UUID.randomUUID().toString()

    val inputTransformerOb = InputTransformer {
        inputTemplate = "\"Notification: sample event was received.\""
    }

    val target = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransformerOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
        targets = listOf(target)
```

```
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}

suspend fun updateToCustomRule(ruleName: String?) {
    val customEventsPattern = "{" +
        "\"source\": [\"ExampleSource\"]," +
        "\"detail-type\": [\"ExampleType\"]" +
        "}"
    val request = PutRuleRequest {
        name = ruleName
        description = "Custom test rule"
        eventPattern = customEventsPattern
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putRule(request)
    }
}

// Update an Amazon S3 object created rule with a transform on the target.
suspend fun updateSnsEventRule(topicArn: String?, ruleName: String?) {
    val targetId = UUID.randomUUID().toString()
    val myMap = mutableMapOf<String, String>()
    myMap["bucket"] = "$.detail.bucket.name"
    myMap["time"] = "$.time"

    val inputTransOb = InputTransformer {
        inputTemplate = "\"Notification: an object was uploaded to bucket
 <bucket> at <time>.\""
        inputPathsMap = myMap
    }
    val targetOb = Target {
        id = targetId
        arn = topicArn
        inputTransformer = inputTransOb
    }

    val targetsRequest = PutTargetsRequest {
        rule = ruleName
```

```
        targets = listOf(targetOb)
        eventBusName = null
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(targetsRequest)
    }
}

suspend fun checkRule(eventRuleName: String?) {
    val ruleRequest = DescribeRuleRequest {
        name = eventRuleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.describeRule(ruleRequest)
        println("The state of the rule is $response")
    }
}

suspend fun changeRuleState(eventRuleName: String, isEnabled: Boolean?) {
    if (!isEnabled!!) {
        println("Disabling the rule: $eventRuleName")
        val ruleRequest = DisableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.disableRule(ruleRequest)
        }
    } else {
        println("Enabling the rule: $eventRuleName")
        val ruleRequest = EnableRuleRequest {
            name = eventRuleName
        }
        EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
            eventBrClient.enableRule(ruleRequest)
        }
    }
}

// Create and upload a file to an S3 bucket to trigger an event.
@Throws(IOException::class)
suspend fun uploadTextFiletoS3(bucketName: String?) {
    val fileSuffix = SimpleDateFormat("yyyyMMddHHmmss").format(Date())
```

```
    val fileName = "TextFile$fileSuffix.txt"
    val myFile = File(fileName)
    val fw = FileWriter(myFile.absoluteFile)
    val bw = BufferedWriter(fw)
    bw.write("This is a sample file for testing uploads.")
    bw.close()

    val putOb = PutObjectRequest {
        bucket = bucketName
        key = fileName
        body = myFile.asByteStream()
    }

    S3Client { region = "us-east-1" }.use { s3Client ->
        s3Client.putObject(putOb)
    }
}

suspend fun listTargetRules(topicArnVal: String?) {
    val ruleNamesByTargetRequest = ListRuleNamesByTargetRequest {
        targetArn = topicArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response =
 eventBrClient.listRuleNamesByTarget(ruleNamesByTargetRequest)
        response.ruleNames?.forEach { rule ->
            println("The rule name is $rule")
        }
    }
}

suspend fun listTargets(ruleName: String?) {
    val ruleRequest = ListTargetsByRuleRequest {
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listTargetsByRule(ruleRequest)
        response.targets?.forEach { target ->
            println("Target ARN: ${target.arn}")
        }
    }
}
```

```
// Add a rule that triggers an SNS target when a file is uploaded to an S3
 bucket.
suspend fun addSnsEventRule(ruleName: String?, topicArn: String?, topicName:
 String, eventRuleName: String, bucketName: String) {
    val targetID = UUID.randomUUID().toString()
    val myTarget = Target {
        id = targetID
        arn = topicArn
    }

    val targetsOb = mutableListOf<Target>()
    targetsOb.add(myTarget)

    val request = PutTargetsRequest {
        eventBusName = null
        targets = targetsOb
        rule = ruleName
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        eventBrClient.putTargets(request)
        println("Added event rule $eventRuleName with Amazon SNS target
 $topicName for bucket $bucketName.")
    }
}

suspend fun subEmail(topicArnVal: String?, email: String?) {
    val request = SubscribeRequest {
        protocol = "email"
        endpoint = email
        returnSubscriptionArn = true
        topicArn = topicArnVal
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        val result = snsClient.subscribe(request)
        println(" Subscription ARN: ${result.subscriptionArn}")
    }
}

suspend fun createSnsTopic(topicName: String): String? {
    val topicPolicy = "{" +
        "\"Version\": \"2012-10-17\"," +
```

```
        "\"Statement\": [{" +
        "\"Sid\": \"EventBridgePublishTopic\"," +
        "\"Effect\": \"Allow\"," +
        "\"Principal\": {" +
        "\"Service\": \"events.amazonaws.com\"" +
        "}," +
        "\"Resource\": \"*\"," +
        "\"Action\": \"sns:Publish\"" +
        "}]" +
        "}"

    val topicAttributes = mutableMapOf<String, String>()
    topicAttributes["Policy"] = topicPolicy

    val topicRequest = CreateTopicRequest {
        name = topicName
        attributes = topicAttributes
    }

    SnsClient { region = "us-east-1" }.use { snsClient ->
        val response = snsClient.createTopic(topicRequest)
        println("Added topic $topicName for email subscriptions.")
        return response.topicArn
    }
}

suspend fun listRules() {
    val rulesRequest = ListRulesRequest {
        eventBusName = "default"
        limit = 10
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val response = eventBrClient.listRules(rulesRequest)
        response.rules?.forEach { rule ->
            println("The rule name is ${rule.name}")
            println("The rule ARN is ${rule.arn}")
        }
    }
}

// Create a new event rule that triggers when an Amazon S3 object is created in a
 bucket.
```

```
suspend fun addEventRule(roleArnVal: String?, bucketName: String, eventRuleName:
 String?) {
    val pattern = """{
        "source": ["aws.s3"],
        "detail-type": ["Object Created"],
        "detail": {
        "bucket": {
            "name": ["$bucketName"]
            }
        }
    }"""

    val ruleRequest = PutRuleRequest {
        description = "Created by using the AWS SDK for Kotlin"
        name = eventRuleName
        eventPattern = pattern
        roleArn = roleArnVal
    }

    EventBridgeClient { region = "us-east-1" }.use { eventBrClient ->
        val ruleResponse = eventBrClient.putRule(ruleRequest)
        println("The ARN of the new rule is ${ruleResponse.ruleArn}")
    }
}

// Set the Amazon S3 bucket notification configuration.
suspend fun setBucketNotification(bucketName: String) {
    val eventBridgeConfig = EventBridgeConfiguration {
    }

    val configuration = NotificationConfiguration {
        eventBridgeConfiguration = eventBridgeConfig
    }

    val configurationRequest = PutBucketNotificationConfigurationRequest {
        bucket = bucketName
        notificationConfiguration = configuration
        skipDestinationValidation = true
    }

    S3Client { region = "us-east-1" }.use { s3Client ->
        s3Client.putBucketNotificationConfiguration(configurationRequest)
        println("Added bucket $bucketName with EventBridge events enabled.")
    }
```

```
}

// Create an S3 bucket using a waiter.
suspend fun createBucket(bucketName: String) {
    val request = CreateBucketRequest {
        bucket = bucketName
    }

    S3Client { region = "us-east-1" }.use { s3 ->
        s3.createBucket(request)
        s3.waitUntilBucketExists {
            bucket = bucketName
        }
        println("$bucketName is ready")
    }
}

suspend fun checkBucket(bucketName: String?): Boolean {
    try {
        // Determine if the S3 bucket exists.
        val headBucketRequest = HeadBucketRequest {
            bucket = bucketName
        }

        S3Client { region = "us-east-1" }.use { s3Client ->
            s3Client.headBucket(headBucketRequest)
            return true
        }
    } catch (e: S3Exception) {
        System.err.println(e.message)
    }
    return false
}

suspend fun createIAMRole(rolenameVal: String?, polJSON: String?): String? {
    val request = CreateRoleRequest {
        roleName = rolenameVal
        assumeRolePolicyDocument = polJSON
        description = "Created using the AWS SDK for Kotlin"
    }

    val rolePolicyRequest = AttachRolePolicyRequest {
        roleName = rolenameVal
        policyArn = "arn:aws:iam::aws:policy/AmazonEventBridgeFullAccess"
```

```
    }

    IamClient { region = "us-east-1" }.use { iam ->
        val response = iam.createRole(request)
        iam.attachRolePolicy(rolePolicyRequest)
        return response.role?.arn
    }
}
```

- Pour plus d'informations sur l'API, consultez les rubriques suivantes dans AWS SDK for Kotlin API reference.
  - DeleteRule
  - DescribeRule
  - DisableRule
  - EnableRule
  - ListRuleNamesByTarget
  - ListRules
  - ListTargetsByRule
  - PutEvents
  - PutRule
  - PutTargets

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Exemples multiservices d' EventBridge utilisation des SDK AWS

Les exemples d'applications suivants utilisent AWS des SDK à combiner EventBridge avec d'autres Services AWS. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions sur la façon de configurer et d'exécuter l'application.

Exemples

- Utilisent des événements planifiés pour appeler une fonction Lambda

Utilisent des événements planifiés pour appeler une fonction Lambda

Les exemples de code suivants montrent comment créer une AWS Lambda fonction invoquée par un événement EventBridge planifié par Amazon.

Java

SDK pour Java 2.x

Montre comment créer un événement EventBridge planifié Amazon qui invoque une AWS Lambda fonction. Configurez EventBridge pour utiliser une expression cron afin de planifier le moment où la fonction Lambda est invoquée. Dans cet exemple, vous créez une fonction Lambda à l'aide de l'API d'exécution Lambda. Cet exemple fait appel à différents AWS services pour réaliser un cas d'utilisation spécifique. Cet exemple montre comment créer une application qui envoie un message texte mobile à vos employés pour les féliciter à leur date d'anniversaire.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

JavaScript

SDK pour JavaScript (v3)

Montre comment créer un événement EventBridge planifié Amazon qui invoque une AWS Lambda fonction. Configurez EventBridge pour utiliser une expression cron afin de planifier le moment où la fonction Lambda est invoquée. Dans cet exemple, vous créez une fonction Lambda à l'aide de l'API d'exécution JavaScript Lambda. Cet exemple fait appel à différents AWS services pour réaliser un cas d'utilisation spécifique. Cet exemple montre comment créer une application qui envoie un message texte mobile à vos employés pour les féliciter à leur date d'anniversaire.

Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Cet exemple est également disponible dans le AWS SDK for JavaScript guide du développeur v3.

Les services utilisés dans cet exemple

- DynamoDB
- EventBridge
- Lambda
- Amazon SNS

Python

SDK pour Python (Boto3)

Cet exemple montre comment enregistrer une AWS Lambda fonction en tant que cible d'un EventBridge événement Amazon planifié. Le gestionnaire Lambda écrit un message convivial et les données complètes de l'événement dans Amazon CloudWatch Logs pour une récupération ultérieure.

- Déploie une fonction Lambda.
- Crée un événement EventBridge planifié et fait de la fonction Lambda la cible.
- Accorde l'autorisation de laisser EventBridge invoquer la fonction Lambda.
- Imprime les dernières données des CloudWatch journaux pour afficher le résultat des appels planifiés.
- Nettoie toutes les ressources créées lors de la démonstration.

Il est préférable de visionner cet exemple sur GitHub. Pour obtenir le code source complet et les instructions de configuration et d'exécution, consultez l'exemple complet sur GitHub.

Les services utilisés dans cet exemple

- CloudWatch Journaux
- EventBridge
- Lambda

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation EventBridge avec un AWS SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Amazon EventBridge sécurité

Amazon EventBridge utilisés AWS Identity and Access Management pour contrôler l'accès à d'autres AWS services et ressources. Pour obtenir un vue d’ensemble du fonctionnement d’IAM, consultez Présentation de la gestion des accès dans le Guide de l’utilisateur IAM. Pour obtenir une présentation des informations d'identification de sécurité, consultez Informations d'identification de sécuritéAWS dans le Référence générale d'Amazon Web Services.

Rubriques

Protection des données sur Amazon EventBridge

Le modèle de responsabilité AWS partagée de s'applique à la protection des données dans Amazon EventBridge. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec EventBridge ou d'autres Services AWS utilisateurs de la console, de l'API ou AWS des SDK. AWS CLI Toutes les données que vous

entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Chiffrement des données pour les bus EventBridge d'événements

EventBridge fournit à la fois le chiffrement au repos et le chiffrement en transit pour protéger les données de vos événements :

- Chiffrement au repos

  EventBridge s'intègre à AWS Key Management Service (KMS) pour chiffrer les données d'événements stockées sur les bus d'événements. Par défaut, EventBridge utilise an Clé détenue par AWS pour chiffrer les données des événements. Vous pouvez également spécifier d'utiliser un EventBridge clé gérée par le client pour les événements personnalisés et ceux des partenaires à la place.
- Chiffrement en transit

  EventBridge chiffre les données qui transitent entre EventBridge et d'autres services à l'aide du protocole TLS (Transport Layer Security). Pour les bus d'événements, cela inclut lors de l'envoi d'un événement à une cible de règles EventBridge, ainsi que lors de l' EventBridge envoi d'un événement à une cible de règles.

Chiffrement au repos pour les bus dédiés aux événements

EventBridge fournit un chiffrement transparent côté serveur en s'intégrant à AWS Key Management Service (KMS). Le chiffrement des données au repos par défaut permet de réduire les frais opérationnels et la complexité liés à la protection des données sensibles. Dans le même temps, il vous permet de créer des applications sécurisées qui répondent aux exigences réglementaires et de conformité strictes en matière de chiffrement.

Le EventBridge chiffrement des données du bus d'événements au repos inclut :

- Données relatives aux événements AWS, aux événements personnalisés et aux événements organisés par les partenaires.

  Pour les bus d'événements, les données d'événement incluent tous les champs contenus dans l'???élément de l'événement.

EventBridge ne chiffre pas les métadonnées des événements. Pour plus d'informations sur les métadonnées des événements, consultez???.

- Schémas d'événements
- Transformateurs d'entrée

Par défaut, EventBridge utilise an Clé détenue par AWS pour chiffrer les données des événements. Vous pouvez également spécifier d'utiliser un EventBridge clé gérée par le client pour les événements personnalisés et ceux des partenaires à la place.

Considérations relatives à la sécurité pour le chiffrement du bus d'événements

Nous vous recommandons vivement de ne jamais saisir d'informations confidentielles ou sensibles dans les champs suivants, car elles ne sont pas cryptées au repos :

- Noms des bus d'événements
- Noms des règles
- Ressources partagées telles que les tags

KMS key options pour le chiffrement du bus d'événements

EventBridge utilise an Clé détenue par AWS pour chiffrer les événements AWS de service stockés sur les bus d'événements.

Pour chaque bus d'événements, vous pouvez choisir le type d' KMS key EventBridge utilisation à utiliser pour chiffrer les événements personnalisés et ceux des partenaires stockés sur ce bus :

- Clé détenue par AWS

  Par défaut, EventBridge chiffre les données à l'aide de la norme de chiffrement avancée 256 bits (AES-256) sous un Clé détenue par AWS, ce qui permet de protéger vos données contre tout accès non autorisé.

  Vous ne pouvez ni afficher, ni gérer, ni utiliser Clés détenues par AWS, ni auditer leur utilisation. Toutefois, vous n'avez pas besoin de prendre de mesure ou de modifier les programmes pour protéger les clés qui chiffrent vos données.

  En général, à moins que vous ne soyez obligé d'auditer ou de contrôler la clé de chiffrement qui protège vos ressources, une Clé détenue par AWS est un bon choix. Clés détenues par AWS sont

totalement gratuits (pas de frais mensuels ni de frais d'utilisation), et ils ne sont pas pris en compte dans les AWS KMS quotas de votre compte. Vous n'avez pas besoin de créer ou de maintenir la clé ou sa politique de clé.

Pour plus d'informations, consultez Clés détenues par AWS dans le Guide du développeur AWS Key Management Service .

- Clé gérée par le client

  EventBridge prend en charge l'utilisation d'un système symétrique clé gérée par le client que vous créez, possédez et gérez. Comme vous avez le contrôle total de ce type de KMS key, vous pouvez effectuer des tâches telles que :

  - Établissement et gestion des stratégies de clé
  - Établissement et gestion des politiques IAM et des octrois
  - Activation et désactivation des stratégies de clé
  - Rotation des matériaux de chiffrement de clé
  - Ajout de balises
  - Création d'alias de clé
  - Planification des clés pour la suppression

  Pour plus d'informations, consultez Clés gérées par le client dans le Guide du développeur AWS Key Management Service .

  EventBridge prend en charge les clés multirégionales et l'accès aux clés entre comptes.

  Clés gérées par le client encourir des frais mensuels. Pour plus de détails, consultez la section AWS Key Management Service Tarification et quotas dans le guide du AWS Key Management Service développeur.

  > **Note**
  >
  > EventBridge ne prend pas en charge les fonctionnalités suivantes sur les bus d'événements chiffrés à l'aide de clés gérées par le client :
  >
  > - Archives
  > - Découverte de schémas

Pour plus d’informations, consultez ???.

Chiffrer les événements avec clés gérées par le client

Vous pouvez spécifier d' EventBridge utiliser a AWS KMS clé gérée par le client pour chiffrer vos données (événements personnalisés et partenaires) stockées sur un bus d'événements, plutôt que d'utiliser un Clé détenue par AWS as par défaut. Vous pouvez spécifier clé gérée par le client quand vous créez ou mettez à jour un bus d'événements. Vous pouvez également mettre à jour le bus d'événements par défaut afin d'en utiliser un clé gérée par le client pour les événements personnalisés et ceux des partenaires. Pour plus d’informations, consultez ???.

Si vous spécifiez un clé gérée par le client pour un bus d'événements, vous avez la possibilité de spécifier une file d'attente de lettres mortes (DLQ) pour le bus d'événements. EventBridge transmet ensuite à ce DLQ tout événement personnalisé ou lié à un partenaire qui génère des erreurs de chiffrement ou de déchiffrement. Pour plus d’informations, consultez ???.

Spécifier un clé gérée par le client pour le chiffrement lors de la création d'un bus d'événements (à l'aide de la console)

- Suivez les instructions ci-dessous :

  ???.

Spécifier un clé gérée par le client pour le chiffrement lors de la création d'un bus d'événements (à l'aide de la CLI)

- Lors de l'appel`create-event-bus`, utilisez l'`kms-key-identifier`option pour spécifier la clé gérée par le client forme EventBridge à utiliser pour le chiffrement sur le bus d'événements.

  Utilisez-le éventuellement `dead-letter-config` pour spécifier une file d'attente de lettres mortes (DLQ).

Mettre à jour un bus d'événements pour utiliser un clé gérée par le client pour le chiffrement (à l'aide de la console)

- Suivez les instructions ci-dessous :

???.

Mettre à jour un bus d'événements pour utiliser un clé gérée par le client pour le chiffrement (à l'aide de la CLI)

- Lors de l'appel`update-event-bus`, utilisez l'`kms-key-identifier`option pour spécifier la clé gérée par le client forme EventBridge à utiliser pour le chiffrement sur le bus d'événements.

  Utilisez-le éventuellement `dead-letter-config` pour spécifier une file d'attente de lettres mortes (DLQ).

Mise à jour du bus d'événements par défaut pour utiliser un clé gérée par le client pour le chiffrement à l'aide de CloudFormation

Étant donné EventBridge que le bus d'événements par défaut est automatiquement intégré à votre compte, vous ne pouvez pas le créer à l'aide d'un CloudFormation modèle, comme vous le feriez normalement pour toute ressource que vous souhaitez inclure dans une CloudFormation pile. Pour inclure le bus d'événements par défaut dans une CloudFormation pile, vous devez d'abord l'importer dans une pile. Une fois que vous avez importé le bus d'événements par défaut dans une pile, vous pouvez mettre à jour les propriétés du bus d'événements comme vous le souhaitez.

- Suivez les instructions ci-dessous :

  ???.

Autoriser l'utilisation EventBridge d'un clé gérée par le client

Si vous utilisez un clé gérée par le client identifiant dans votre compte pour protéger le bus de votre EventBridge événement, les politiques en la matière KMS key doivent EventBridge autoriser son utilisation en votre nom. Vous fournissez ces autorisations dans une politique clé.

EventBridge n'a pas besoin d'autorisation supplémentaire pour utiliser la valeur par défaut Clé détenue par AWS afin de protéger les EventBridge ressources de votre AWS compte.

EventBridge nécessite les autorisations suivantes sur un clés gérées par le client :

- `kms:DescribeKey`

EventBridge nécessite cette autorisation pour récupérer l' KMS key ARN de l'identifiant de clé fourni et pour vérifier que la clé est symétrique.

- kms:GenerateDataKey

  EventBridge nécessite cette autorisation pour générer une clé de données en tant que clé de chiffrement pour les données d'événement.

- kms:Decrypt

  EventBridge nécessite cette autorisation pour déchiffrer la clé de données chiffrée et stockée avec les données d'événement chiffrées.

  EventBridge l'utilise pour faire correspondre les règles ; les utilisateurs n'ont jamais accès aux données.

L'exemple de politique clé suivant fournit les autorisations requises :

```
{
  "Sid": "Allow EventBridge to encrypt events",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ]
  "Resource": "*",
  "Condition": {
    "StringEquals": {
        "kms:EncryptionContext:aws:events:event-bus:arn":
 "arn:aws:events:region:account-id:event-bus/event-bus-arn",
        "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-name"
    }
  }
}
```

Sécurité lors de l'utilisation clés gérées par le client pour le chiffrement du bus d' EventBridge événements

La meilleure pratique en matière de sécurité consiste à ajouter une clé `aws:SourceArnaws:sourceAccount`, ou une clé de `kms:EncryptionContext:aws:events:event-bus:arn` condition à la politique AWS KMS clé. La clé de condition IAM globale permet de garantir que la clé KMS est EventBridge utilisée uniquement pour le bus ou le compte spécifié.

L'exemple suivant montre comment appliquer cette bonne pratique dans votre IAM politique :

```
{
      "Sid": "Allow the use of key",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt"
      ],
      "Resource": "*",
      "Condition" : {
        "StringEquals": {
          "aws:SourceAccount": "arn:aws:events:region:account-id",
          "aws:SourceArn": "arn:aws:events:region:account-id:event-bus/event-bus-name",
          "kms:EncryptionContext:aws:events:event-bus:arn":
 "arn:aws:events:region:account-id:event-bus/event-bus-arn"
        }
      }
```

Gestion du clés gérées par le client chiffrement du bus d' EventBridge événements

Pour garantir qu' EventBridge il conserve toujours l'accès au nécessaire clé gérée par le client :

- Ne supprimez pas un clé gérée par le client tant que vous n'êtes pas certain que tous les événements chiffrés avec celui-ci ont été traités.

  Lorsque vous effectuez l'une des opérations suivantes, conservez le contenu clé précédent afin de EventBridge pouvoir continuer à l'utiliser pour des événements précédemment chiffrés :

  - Rotation automatique des clés

- Rotation manuelle des touches
- Mettre à jour un alias de clé

  En général, si vous envisagez de supprimer une AWS KMS clé, désactivez-la d'abord et configurez une CloudWatch alarme ou un mécanisme similaire pour être certain de ne jamais avoir à utiliser la clé pour déchiffrer des données chiffrées.
- Ne supprimez pas la politique clé qui fournit EventBridge les autorisations d'utilisation de la clé.

Parmi les autres considérations, mentionnons :

- Spécifiez clés gérées par le client les cibles des règles, le cas échéant.

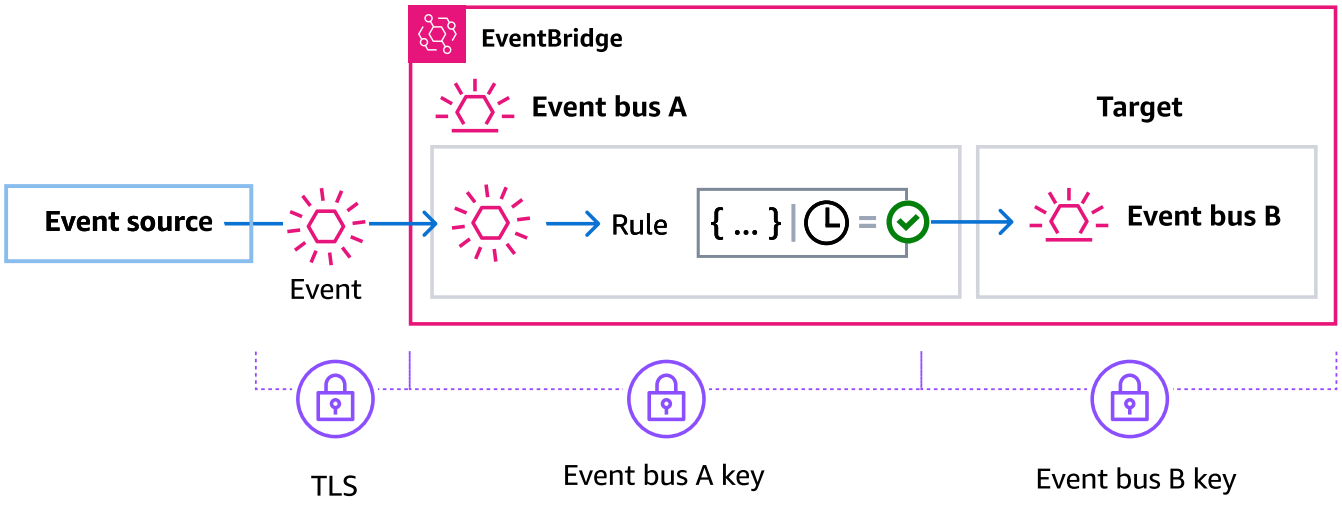
  Lors de l' EventBridge envoi d'un événement à une cible de règles, l'événement est envoyé à l'aide du protocole TLS (Transport Layer Security). Toutefois, le type de chiffrement appliqué à l'événement tel qu'il est stocké sur la cible dépend du chiffrement que vous avez configuré sur la cible elle-même.

Chiffrement des événements lorsqu'un bus d'événements est la cible de la règle

Lorsqu'un événement personnalisé ou partenaire est envoyé à un bus d'événements, EventBridge chiffre cet événement conformément à la configuration de la clé KMS de chiffrement au repos pour ce bus d'événements, soit par défaut, Clé détenue par AWS soit par un clé gérée par le client, si une clé a été spécifiée. Si un événement correspond à une règle, EventBridge chiffre l'événement avec la configuration de clé KMS pour ce bus d'événements jusqu'à ce que l'événement soit envoyé à la cible de la règle, sauf si la cible de la règle est un autre bus d'événements.

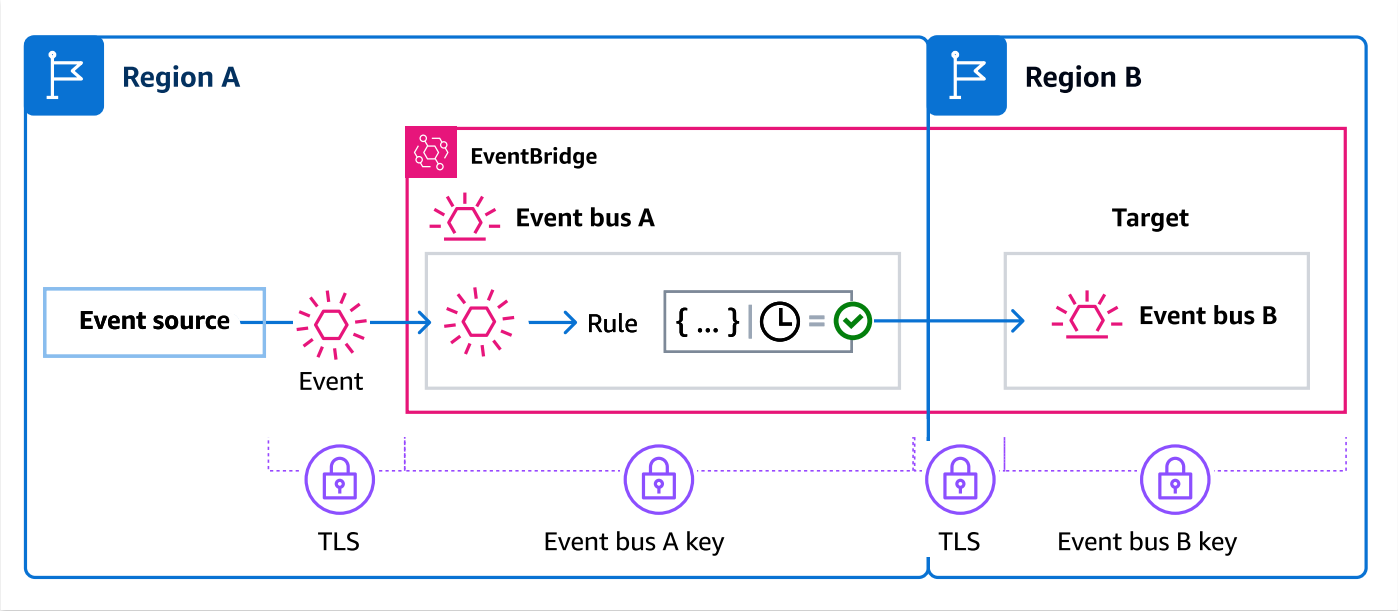
- Si la cible d'une règle est un autre bus d'événements dans la même AWS région :

  Si le bus d'événements cible a une valeur spécifiée clé gérée par le client, EventBridge chiffre l'événement avec le bus clé gérée par le client d'événements cible pour la livraison à la place.



- Si la cible d'une règle est un autre bus d'événements dans une autre AWS région :

  EventBridge chiffre l'événement au repos conformément à la configuration de la clé KMS sur le premier bus d'événements. EventBridge utilise le protocole TLS pour envoyer l'événement au deuxième bus d'événements de la région différente, où il est ensuite chiffré conformément à la configuration de clé KMS spécifiée pour le bus d'événements cible.

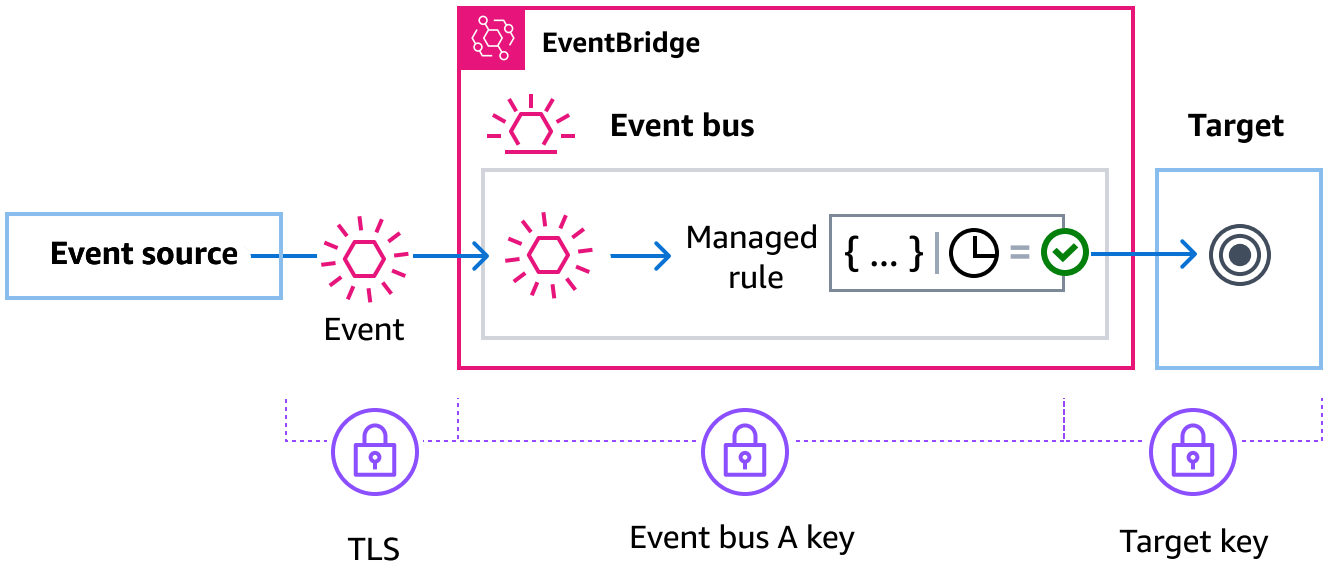


Chiffrement des événements pour les règles gérées

AWS les services peuvent créer et gérer les règles du bus d'événements dans votre AWS compte qui sont nécessaires à certaines fonctions de ces services. Dans le cadre d'une règle gérée, le AWS

service peut spécifier d' EventBridge utiliser la valeur clé gérée par le client spécifiée pour la cible de la règle. Cela vous donne la possibilité de spécifier laquelle clé gérée par le client utiliser en fonction de la cible de la règle.

Dans ces cas, une fois qu'un événement personnalisé ou partenaire correspond à la règle gérée, EventBridge utilise la cible clé gérée par le client spécifiée par la règle gérée pour chiffrer l'événement jusqu'à ce qu'il soit envoyé à la cible de la règle. Cela vaut indépendamment du fait que le bus d'événements ait été configuré pour utiliser le sien à clé gérée par le client des fins de chiffrement. C'est le cas même si la cible de la règle gérée est un autre bus d'événements et que ce bus d'événements possède ses propres clé gérée par le client spécifications de chiffrement. EventBridge continue d'utiliser la cible clé gérée par le client spécifiée dans la règle gérée jusqu'à ce que l'événement soit envoyé à une cible qui n'est pas un bus d'événements.



Dans les cas où la cible de la règle est un bus d'événements dans une autre région, vous devez fournir une clé multirégionale. Le bus d'événements de la première région chiffre l'événement à l'aide de ce qui est clé gérée par le client spécifié dans la règle gérée. Il envoie ensuite l'événement au bus d'événements cible dans la deuxième région. Ce bus d'événements doit pouvoir continuer à utiliser le clé gérée par le client jusqu'à ce qu'il envoie l'événement à sa cible.

EventBridge contexte de chiffrement du bus d'événements

Un contexte de chiffrement est un ensemble de paires clé-valeur qui contiennent des données non secrètes arbitraires. Lorsque vous incluez un contexte de chiffrement dans une demande de

chiffrement de données, AWS KMS lie de manière chiffrée le contexte de chiffrement aux données chiffrées. Pour déchiffrer les données, vous devez transmettre le même contexte de chiffrement.

Vous pouvez également utiliser le contexte de chiffrement comme condition d'autorisation dans les politiques et les autorisations.

Pour les bus d'événements, EventBridge utilise le même contexte de chiffrement dans toutes les opérations AWS KMS cryptographiques. Si vous utilisez une clé gérée par le client pour protéger vos EventBridge ressources, vous pouvez utiliser le contexte de chiffrement pour identifier l'utilisation de cette clé KMS key dans les enregistrements et les journaux d'audit. Il apparaît également en texte brut dans les journaux, tels que AWS CloudTrail et Amazon CloudWatch Logs.

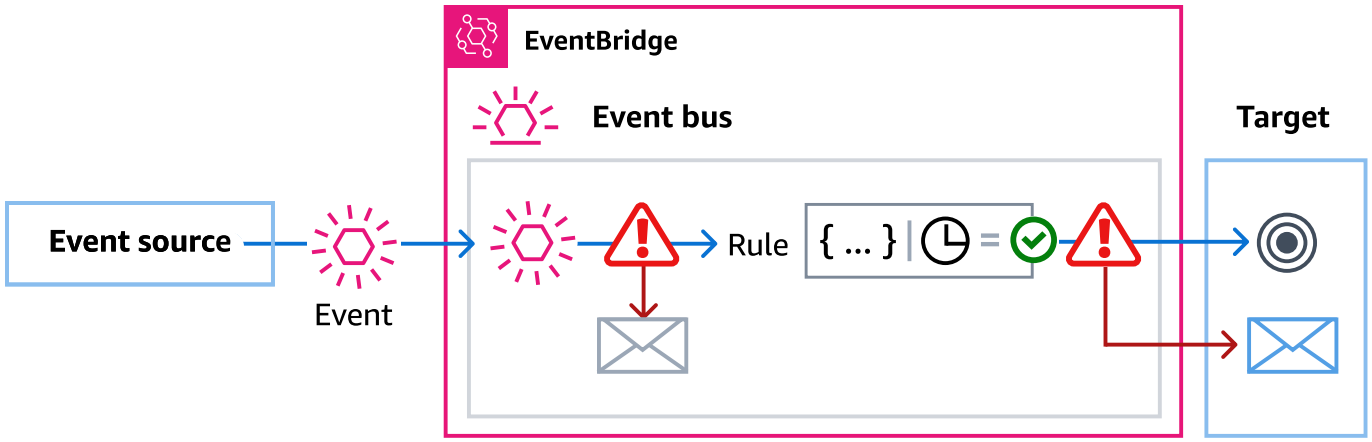
Dans ses demandes à AWS KMS, EventBridge utilise un contexte de chiffrement avec une seule paire clé-valeur, qui contient l'ARN du bus d'événements :

```
"encryptionContext": {
    "kms:EncryptionContext:aws:events:event-bus:arn": "event-bus-arn"
}
```

Utilisation de files d'attente contenant des lettres mortes pour capturer les erreurs liées aux événements chiffrés

Si vous configurez le clé gérée par le client chiffrement sur un bus d'événements, nous vous recommandons de spécifier une file d'attente de lettres mortes (DLQ) pour ce bus d'événements. EventBridge envoie des événements personnalisés et partenaires à ce DLQ s'il rencontre une erreur irrécupérable lors du traitement de l'événement sur le bus d'événements. Une erreur non récupérable est une erreur nécessitant une action de l'utilisateur pour résoudre le problème sous-jacent, tel que la désactivation ou l'absence clé gérée par le client du paramètre spécifié.

- Si une erreur de chiffrement ou de déchiffrement non récupérable se produit lors EventBridge du traitement de l'événement sur le bus d'événements, l'événement est envoyé au DLQ pour le bus d'événements, s'il en est spécifié un.
- Si une erreur de chiffrement ou de déchiffrement non récupérable se produit lors EventBridge de la tentative d'envoi de l'événement à une cible, l'événement est envoyé au DLQ de la cible, s'il en est spécifié un.



Pour plus d'informations, notamment les considérations relatives à l'utilisation des DLQ et les instructions relatives à la définition des autorisations, consultez???.

Décryptage d'événements dans des files d'attente contenant des lettres mortes EventBridge

Une fois que vous avez résolu le problème sous-jacent à l'origine d'une erreur non récupérable, vous pouvez traiter les événements envoyés au bus d'événements ou aux DLQ cibles. Pour les événements chiffrés, vous devez d'abord les déchiffrer afin de pouvoir les traiter.

L'exemple suivant montre comment déchiffrer un événement transmis à un bus d'événements ou à une DLQ cible. EventBridge

```
// You will receive an encrypted event in the following json format.
        // ```
        //   {
        //     "version": "0",
        //     "id": "053afa53-cdd7-285b-e754-b0dfd0ac0bfb",  // New event id not the
 same as the original one
        //     "account": "123456789012",
        //     "time": "2020-02-10T10:22:00Z",
        //     "resources": [ ],
        //     "region": "us-east-1",
        //     "source": "aws.events",
        //     "detail-type": "Encrypted Events",
        //     "detail": {
        //       "event-bus-arn": "arn:aws:events:region:account:event-bus/bus-name",
        //       "rule-arn": "arn:aws:events:region:account:event-bus/bus-name/rule-
name",
        //       "kms-key-arn": "arn:aws:kms:region:account:key/key-arn",
```

```
        //       "encrypted-payload": "AgR4qiru/XNwTUyCgRHqP7rbbHn/
xpmVeVeRIAd12TDYYVwAawABABRhd3M6ZXZlbnRzOmV2ZW50LWJ1cwB
        //
 RYXJuOmF3czpldmVudHM6dXMtZWFzdC0xOjE0NjY4NjkwNDY3MzpldmVudC1idXMvY21rbXMtZ2EtY3Jvc3
        //
 MtYWNjb3VudC1zb3VyY2UtYnVzAAEAB2F3cy1rbXMAS2Fybjphd3M6a21zOnVzLWVhc3QtMToxNDY2ODY5"
        //     }
        //   }
        // ```

        // Construct an AwsCrypto object with the encryption algorithm
`ALG_AES_256_GCM_HKDF_SHA512_COMMIT_KEY` which
        // is used by EventBridge for encryption operation. This object is an entry
point for decryption operation.
        // It can later use decryptData(MasterKeyProvider, byte[]) method to decrypt
data.
        final AwsCrypto crypto = AwsCrypto.builder()

.withEncryptionAlgorithm(CryptoAlgorithm.ALG_AES_256_GCM_HKDF_SHA512_COMMIT_KEY)
                .build();

        // Construct AWS KMS master key provider with AWS KMS Client Supplier and AWS
KMS Key ARN. The KMS Client Supplier can
        // implement a RegionalClientSupplier interface. The AWS KMS Key ARN can be
fetched from kms-key-arn property in
        // encrypted event json detail.
        final KmsMasterKeyProvider kmsMasterKeyProvider =
KmsMasterKeyProvider.builder()
                .customRegionalClientSupplier(...)
                .buildStrict(KMS_KEY_ARN);

        // The string of encrypted-payload is base64 encoded. Decode it into byte
array, so it can be furthur
        // decrypted. The encrypted payload can be fetched from encrypted-payload field
in encrypted event json detail.
        byte[] encryptedByteArray = Base64.getDecoder().decode(ENCRYPTED_PAYLOAD);

        // The decryption operation. It retrieves the encryption context and encrypted
data key from the cipher
        // text headers, which is parsed from byte array encrypted data. Then it
decrypts the data key, and
        // uses it to finally decrypt event payload. This encryption/decryption
strategy is called envelope
```

```
        // encryption, https://docs.aws.amazon.com/kms/latest/developerguide/
concepts.html#enveloping
        final CryptoResult<byte[], KmsMasterKey> decryptResult =
 crypto.decryptData(kmsMasterKeyProvider, encryptedByteArray);

        final byte[] decryptedByteArray = decryptResult.getResult();

        // Decode the event json plaintext from byte array into string with UTF_8
 standard.
        String eventJson = new String(decryptedByteArray, StandardCharsets.UTF_8);
```

Politiques basées sur des balises

Dans Amazon EventBridge, vous pouvez utiliser des politiques basées sur les balises afin de contrôler l'accès aux ressources.

Par exemple, vous pouvez limiter l'accès aux ressources qui incluent une balise avec la clé `environment` et la valeur `production`. L'exemple de politique suivant refuse à toute ressource dotée de cette balise la possibilité de créer, supprimer ou des balises, des règles ou des bus d'événements pour les ressources que vous avez balisées avec `environment/production`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "events:PutRule",
                "events:DescribeRule",
                "events:DeleteRule",
                "events:CreateEventBus",
                "events:DescribeEventBus"
                "events:DeleteEventBus"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/environment": "production"}
            }
        }
    ]
}
```

Pour plus d'informations sur le balisage, consultez les ressources suivantes :

- EventBridge Balises Amazon
- Contrôle de l'accès à l'aide de balises IAM

Amazon EventBridge et AWS Identity and Access Management

Pour accéder à Amazon EventBridge, vous avez besoin d'informations d'identification qui AWS peuvent être utilisées pour authentifier vos demandes. Vos informations d'identification doivent être autorisées à accéder aux ressources AWS, par exemple pour extraire les données d'événements d'autres ressources AWS. Les sections suivantes fournissent des informations détaillées sur la manière dont vous pouvez utiliser AWS Identity and Access Management(IAM) et vous aider EventBridge à sécuriser vos ressources en contrôlant qui peut y accéder.

Rubriques

Authentification

Vous pouvez utiliser les types d'identité suivants pour accéder à AWS :

- Utilisateur root du compte AWS : lorsque vous vous inscrivez à AWS, vous fournissez l'adresse e-mail et le mot de passe qui sont associés à votre compte. Il s'agit de vos informations d'identification racine et elles fournissent un accès complet à l'ensemble de vos ressources AWS.

> **Important**
>
> Pour des raisons de sécurité, nous vous recommandons d'utiliser les informations d'identification root uniquement pour créer un administrateur, qui est un utilisateur IAM disposant d'autorisations complètes sur votre compte. Vous pouvez ensuite utiliser cet administrateur pour créer d'autres utilisateurs et rôles dotés d'autorisations limitées.

Pour plus d'informations, consultez Bonnes pratiques IAM et Création d'un utilisateurs administrateur et d'un groupe dans le Guide de l'utilisateur IAM.

- Utilisateur IAM — Un utilisateur IAM est une identité au sein de votre compte qui dispose d'autorisations spécifiques, par exemple l'autorisation d'envoyer des données d'événements à une cible dans. EventBridge Vous pouvez utiliser des informations d'identification de connexion IAM pour vous connecter aux pages web AWS sécurisées, comme la AWS Management Console, les forums de discussion AWS ou le centre AWS Support.

  En plus des informations d'identification de connexion, vous pouvez également générer des clés d'accès pour chaque utilisateur. Vous pouvez utiliser ces clés lorsque vous accédez aux services AWS par programmation pour signer votre demande par chiffrement, que ce soit par l'intermédiaire de l'un des kits SDK ou en utilisant l'AWS Command Line Interface (AWS CLI). Si vous n'utilisez pas les outils AWS, vous devez vous-même signer la demande avec Signature Version 4, protocole destiné à authentifier les demandes d'API entrantes. Pour plus d'informations sur l'authentification des demandes, consultez Processus de signature Signature Version 4 dans le document Référence générale d'Amazon Web Services.

- Rôle IAM : un rôle IAM est une autre identité IAM que vous pouvez créer dans votre compte et qui dispose d'autorisations spécifiques. S'il est comparable à un utilisateur IAM, il n'est toutefois pas associé à une personne déterminée. En utilisant un rôle IAM, vous pouvez obtenir des clés d'accès temporaires pour accéder à des ressources et services AWS. Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

  - Accès d'un utilisateur fédéré : au lieu de créer un utilisateur, vous pouvez utiliser des identités provenant d'AWS Directory Service, de l'annuaire des utilisateurs de votre entreprise ou d'un fournisseur d'identité (IdP) web. Ceux-ci sont connus sous le nom d'utilisateurs fédérés. AWS attribue un rôle à un utilisateur fédéré lorsque l'utilisateur demande un accès via un fournisseur d'identité. Pour plus d'informations sur les utilisateurs fédérés, consultez Utilisateurs fédérés et rôles dans le Guide de l'utilisateur IAM.
  - Accès intercompte : vous pouvez utiliser un rôle IAM de votre compte pour autoriser un autre compte à accéder aux ressources de votre compte. Pour obtenir un exemple, consultez le Didacticiel : déléguer l'accès entre des comptes AWS à l'aide de rôles IAM du Guide de l'utilisateur IAM.
  - Accès d'un service AWS : vous pouvez utiliser un rôle IAM de votre compte pour autoriser un service AWS à accéder aux ressources de votre compte. Par exemple, vous pouvez créer un rôle qui autorise Amazon Redshift à charger les données stockées dans un compartiment Amazon S3 dans un cluster Amazon Redshift. Pour plus d'informations, veuillez consulter

Création d'un rôle pour la délégation d'autorisations à un service AWS dans le Guide de l'utilisateur IAM.

- Applications exécutées sur Amazon EC2 : pour les applications Amazon EC2 qui ont besoin d'accéder EventBridge à, vous pouvez soit stocker les clés d'accès dans l'instance EC2, soit utiliser un rôle IAM pour gérer les informations d'identification temporaires. Pour attribuer un rôle AWS à une instance EC2, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle, et il fournit des informations d'identification temporaires aux applications s'exécutant sur l'instance EC2. Pour plus d'informations, consultez Utilisation de rôles pour des applications s'exécutant sur Amazon EC2 dans le Guide de l'utilisateur IAM.

Contrôle d'accès

Pour créer des EventBridge ressources ou y accéder, vous devez disposer d'informations d'identification et d'autorisations valides. Par exemple, pour invoquer des cibles AWS Lambda, Amazon Simple Notification Service (Amazon SNS) et Amazon Simple Queue Service (Amazon SQS), vous devez disposez d'autorisations d'accès à ces services.

Gestion des autorisations d'accès à vos ressources Amazon EventBridge

Vous pouvez gérer l'accès aux ressources EventBridge, telles que les règles ou les événements, à l'aide de politiques basées sur l'identité ou basées sur les ressources.

Ressources EventBridge

Chaque ressource et sous-ressource EventBridge est associée à un Amazon Resource Name (ARN) unique. Dans EventBridge, les ARN vous permettent de créer des modèles d'événements. Pour plus d'informations générales sur les ARN, consultez Amazon Resource Name (ARN)AWS et Espaces de noms dans le manuel Référence générale d'Amazon Web Services.

Pour voir la liste des opérations EventBridge qui permettent d'utiliser des ressources, consultez Informations de référence sur les autorisations Amazon EventBridge.

> **Note**
>
> La plupart des services AWS interprètent de la même manière les signes deux points (`:`) et barre oblique (`/`) dans les noms ARN. Cependant, EventBridge utilise une correspondance exacte dans les règles et les modèles 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 que vous souhaitez faire correspondre.

Le tableau suivant présente les ressources EventBridge.

Type de ressource	Format ARN
Archivage	`arn:aws:events:` *`region`*`:`*`account`*`:archive/` *`archive-name`*
Relire	`arn:aws:events:` *`region`*`:`*`account`*`:replay/`*`replay-name`
Règle	`arn:aws:events:` *`region`*`:`*`account`*`:rule/[event-bus-name]/`*`rule-name`
Bus d'événement	`arn:aws:events:` *`region`*`:`*`account`*`:event-bus/` *`event-bus-name`*

Type de ressource	Format ARN
Toutes les ressources EventBridge	`arn:aws:events:*`
Toutes les ressources EventBridge dont est propriétaire le compte spécifié dans la région indiquée	`arn:aws:events: region:account:*`

L'exemple suivant vous montre comment indiquer une règle spécifique (*myRule*) dans votre déclaration à l'aide de son ARN.

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/myRule"
```

Pour spécifier toutes les règles qui appartiennent à un compte spécifique, utilisez le caractère générique astérisque (*) comme suit.

```
"Resource": "arn:aws:events:us-east-1:123456789012:rule/*"
```

Pour spécifier toutes les ressources, ou si une action d'API spécifique ne prend pas en charge les ARN, utilisez le caractère générique astérisque (*) dans l'élément `Resource` comme suit.

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

Pour spécifier plusieurs ressources ou `PutTargets` dans une même déclaration, séparez leurs ARN par des virgules comme suit.

```
"Resource": ["arn1", "arn2"]
```

Propriété des ressources

Les ressources sont la propriété d'un compte, indépendamment des personnes qui les ont créées. Le propriétaire d'une ressource est le compte de l'entité du principal, l'utilisateur root du compte, un utilisateur ou un rôle IAM qui authentifie la demande de création de la ressource. Les exemples suivants illustrent comment cela fonctionne :

- Si vous utilisez les informations d'identification de l'utilisateur root de votre compte pour créer une règle, votre compte est le propriétaire de la ressource EventBridge.
- Si vous créez un utilisateur dans votre compte et que vous lui accordez les autorisations permettant de créer des ressources EventBridge, cet utilisateur peut créer des ressources EventBridge. Cependant, votre compte, auquel l'utilisateur appartient, est le propriétaire des ressources EventBridge.
- Si vous créez un rôle IAM dans votre compte avec des autorisations de création de ressources EventBridge, toute personne pouvant endosser le rôle peut créer des ressources EventBridge. Votre compte, auquel le rôle appartient, est propriétaire des ressources EventBridge.

Gestion de l'accès aux ressources

Une politique d'autorisation décrit qui a accès à quoi. La section suivante explique les options disponibles pour créer des politiques d'autorisations.

> **Note**
>
> Cette section traite de l'utilisation d'IAM dans le contexte d'EventBridge. Elle ne fournit pas d'informations détaillées sur le service IAM. Pour une documentation complète sur IAM, consultez la rubrique Qu'est-ce que IAM ? dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la syntaxe et les descriptions des politiques IAM, consultez la Référence des politiques IAM dans le Guide de l'utilisateur IAM.

Les politiques attachées à une identité IAM sont appelées politiques basées sur une entité (politiques IAM) et les politiques attachées à une ressource sont appelées politiques basées sur une ressource. Dans EventBridge, vous pouvez utiliser à la fois des politiques basées sur l'identité (politiques IAM) et des politiques basées sur les ressources.

Rubriques

- Politiques basées sur une identité (politiques IAM)
- Politiques basées sur les ressources (politiques IAM)

Politiques basées sur une identité (politiques IAM)

Vous pouvez attacher des politiques à des identités IAM. Par exemple, vous pouvez effectuer les opérations suivantes :

- Attacher une politique d'autorisations à un utilisateur ou à un groupe de votre compte : pour accorder à un utilisateur une autorisation de consultation de règles dans la console CloudWatch, attachez une politique d'autorisations à l'utilisateur ou à un groupe auquel l'utilisateur appartient.
- Attacher une politique d'autorisations à un rôle (accorder des autorisations entre comptes) : vous pouvez attacher une politique d'autorisation basée sur une identité à un rôle IAM afin d'accorder des autorisations entre comptes. Par exemple, l'administrateur du compte A peut créer un rôle pour accorder des autorisations intercomptes à un autre compte B ou à un service AWS comme suit :

  1. L'administrateur du compte A crée un rôle IAM et attache une politique d'autorisations à ce rôle qui accorde une autorisation sur les ressources du compte A.
  2. L'administrateur du compte A lie une politique d'approbation au rôle identifiant le compte B comme principal pouvant assumer ce rôle.
  3. L'administrateur du compte B peut ensuite déléguer des autorisations à des utilisateurs du compte B de sorte qu'ils endossent le rôle. Les utilisateurs du compte B sont alors autorisés à créer des ressources dans le compte A ou à y accéder. Le principal de la politique d'approbation peut aussi être un principal de service AWS pour accorder à un service AWS l'autorisation nécessaire pour endosser le rôle.

  Pour en savoir plus sur l'utilisation d'IAM pour déléguer des autorisations, consultez Gestion des accès dans le Guide de l'utilisateur IAM.

Vous pouvez créer des politiques IAM spécifiques pour restreindre les appels et les ressources auxquels les utilisateurs de votre compte ont accès et attacher ensuite ces politiques aux utilisateurs. Pour savoir comment créer des rôles IAM et explorer des exemples de déclarations de politique IAM pour EventBridge, consultez Gestion des autorisations d'accès à vos ressources Amazon EventBridge.

Politiques basées sur les ressources (politiques IAM)

Lorsqu'une règle s'exécute dans EventBridge, toutes les cibles associées à la règle sont invoquées, ce qui recouvre l'invocation des fonctions AWS Lambda, la publication dans les rubriques Amazon SNS ou le relais de l'événement vers les flux Amazon Kinesis. Pour effectuer des appels sur les ressources dot vous êtes propriétaire, EventBridge doit disposer de l'autorisation appropriée. Pour les ressources Lambda, Amazon SNS et Amazon SQS, EventBridge utilise des politiques basées sur les ressources. Pour les flux Kinesis, EventBridge utilise des rôles IAM.

Pour savoir comment créer des rôles IAM et explorer des exemples de déclarations de politique basée sur les ressources pour EventBridge, consultez Utilisation de politiques basées sur les ressources pour Amazon EventBridge.

Spécification des éléments d'une politique : actions, effets et principaux

Pour chaque ressource EventBridge, EventBridge définit un ensemble d'opérations d'API. Pour accorder des autorisations pour ces opérations d'API, EventBridge définit un ensemble d'actions que vous pouvez spécifier dans une politique. Certaines opérations d'API peuvent nécessiter des autorisations pour plusieurs actions. Pour plus d'informations sur les ressources et les opérations de l'API, consultez Ressources EventBridge et Informations de référence sur les autorisations Amazon EventBridge.

Voici les éléments de base d'une politique :

- Ressource – Utilisez un nom Amazon Resource Name (ARN) pour identifier la ressource à laquelle s'applique la politique. Pour de plus amples informations, veuillez consulter Ressources EventBridge.
- Action : utilisez des mots-clés pour identifier les opérations de ressource que vous voulez accorder ou refuser. Par exemple, l'autorisation `events:Describe` permet à l'utilisateur d'effectuer l'opération `Describe`.
- Effet : spécifiez allow ou deny. Si vous n'accordez pas explicitement l'accès (allow) à une ressource, l'accès est refusé. Vous pouvez aussi refuser explicitement l'accès à une ressource dans le but d'empêcher un utilisateur d'y accéder, même si une politique différente accorde l'accès.
- Principal : dans les politiques basées sur une identité (politiques IAM), l'utilisateur auquel la politique est attachée est le principal implicite. Pour les politiques basées sur une ressource, vous spécifiez l'utilisateur, le compte, le service ou une autre entité qui doit recevoir les autorisations (s'applique uniquement aux politiques basées sur une ressource).

Pour plus d'informations sur les politiques IAM et leur syntaxe, consultez Référence de politique JSON IAM dans le Guide de l'utilisateur IAM.

Pour en savoir plus sur les actions d'API EventBridge et les ressources auxquelles elles s'appliquent, consultez Informations de référence sur les autorisations Amazon EventBridge.

Spécification de conditions dans une politique

Lorsque vous accordez des autorisations, vous pouvez utiliser le langage d'access policy pour spécifier les conditions définissant quand une politique doit prendre effet. Par exemple, il est possible d'appliquer une politique après seulement une date spécifique. Pour plus d'informations sur la spécification de conditions dans un langage de politique, consultez Condition dans le Guide de l'utilisateur IAM.

Pour définir des conditions, vous devez utiliser des clés de condition. Il existe des clés de condition AWS et des clés propres à EventBridge que vous pouvez utiliser selon les besoins. Pour obtenir la liste complète des clés AWS, consultez Clés disponibles pour les conditions dans le Guide de l'utilisateur IAM. Pour obtenir la liste complète des clés propres à EventBridge, consultez Utilisation de conditions de politique IAM pour un contrôle d'accès précis.

Utilisation de politiques basées sur l'identité (politiques IAM) pour Amazon EventBridge

Les politiques basées sur l’identité sont des politiques d’autorisations que vous attachez à des identités IAM.

Rubriques

AWS politiques gérées pour EventBridge

AWS répond à de nombreux cas d'utilisation courants en fournissant des politiques IAM autonomes créées et administrées par. AWS Les politiques gérées, ou prédéfinies, accordent les autorisations nécessaires pour les cas d’utilisation courants, ce qui vous évite d’avoir à déterminer quelles autorisations sont nécessaires. Pour plus d’informations, consultez la section Politiques gérées par AWS dans le Guide de l’utilisateur IAM.

Les politiques AWS gérées suivantes que vous pouvez associer aux utilisateurs de votre compte sont spécifiques à EventBridge :

- AmazonEventBridgeFullAccess— Accorde un accès complet à EventBridge, y compris à EventBridge Pipes, EventBridge Schemas et EventBridge Scheduler.
- AmazonEventBridgeReadOnlyAccess— Accorde un accès en lecture seule à EventBridge, y compris à EventBridge Pipes, EventBridge Schemas et Scheduler. EventBridge

AmazonEventBridgeFullAccess politique

La AmazonEventBridgeFullAccess politique accorde des autorisations pour utiliser toutes les EventBridge actions, ainsi que les autorisations suivantes :

- `iam:CreateServiceLinkedRole`— EventBridge nécessite cette autorisation pour créer le rôle de service dans votre compte pour les destinations d'API. Cette autorisation accorde uniquement les autorisations du service IAM nécessaires pour créer un rôle dans votre compte, plus particulièrement pour les destinations d’API.

- `iam:PassRole`— EventBridge nécessite cette autorisation pour transmettre un rôle d'invocation EventBridge à invoquer la cible d'une règle.
- Autorisations du Gestionnaire de secrets EventBridge : ces autorisations sont nécessaires pour gérer les secrets de votre compte lorsque vous fournissez des informations d'identification via la ressource de connexion pour autoriser les destinations API.

Le JSON suivant montre la AmazonEventBridgeFullAccess politique.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "EventBridgeActions",
            "Effect": "Allow",
            "Action": [
                "events:*",
                "schemas:*",
                "scheduler:*",
                "pipes:*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "IAMCreateServiceLinkedRoleForApiDestinations",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
AmazonEventBridgeApiDestinationsServiceRolePolicy",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "apidestinations.events.amazonaws.com"
                }
            }
        },

        {
            "Sid": "SecretsManagerAccessForApiDestinations",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:CreateSecret",
                "secretsmanager:UpdateSecret",
                "secretsmanager:DeleteSecret",
```

```
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:events!*"
        },
        {
            "Sid": "IAMPassRoleAccessForEventBridge",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "events.amazonaws.com"
                }
            }
        },
        {
            "Sid": "IAMPassRoleAccessForScheduler",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "scheduler.amazonaws.com"
                }
            }
        },
        {
            "Sid": "IAMPassRoleAccessForPipes",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/*",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "pipes.amazonaws.com"
                }
            }
        }
    ]
}
```

> **Note**
>
> Les informations contenues dans cette section s'appliquent également à la politique `CloudWatchEventsFullAccess`. Cependant, il est fortement recommandé d'utiliser Amazon EventBridge au lieu d'Amazon CloudWatch Events.

AmazonEventBridgeReadOnlyAccess politique

La AmazonEventBridgeReadOnlyAccess politique accorde des autorisations pour utiliser toutes les EventBridge actions de lecture.

Le JSON suivant montre la AmazonEventBridgeReadOnlyAccess politique.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "events:DescribeRule",
                "events:DescribeEventBus",
                "events:DescribeEventSource",
                "events:ListEventBuses",
                "events:ListEventSources",
                "events:ListRuleNamesByTarget",
                "events:ListRules",
                "events:ListTargetsByRule",
                "events:TestEventPattern",
                "events:DescribeArchive",
                "events:ListArchives",
                "events:DescribeReplay",
                "events:ListReplays",
                "events:DescribeConnection",
                "events:ListConnections",
                "events:DescribeApiDestination",
                "events:ListApiDestinations",
                "events:DescribeEndpoint",
                "events:ListEndpoints",
                "schemas:DescribeCodeBinding",
                "schemas:DescribeDiscoverer",
                "schemas:DescribeRegistry",
```

```
                "schemas:DescribeSchema",
                "schemas:ExportSchema",
                "schemas:GetCodeBindingSource",
                "schemas:GetDiscoveredSchema",
                "schemas:GetResourcePolicy",
                "schemas:ListDiscoverers",
                "schemas:ListRegistries",
                "schemas:ListSchemas",
                "schemas:ListSchemaVersions",

                "schemas:ListTagsForResource",
                "schemas:SearchSchemas",
                "scheduler:GetSchedule",
                "scheduler:GetScheduleGroup",
                "scheduler:ListSchedules",
                "scheduler:ListScheduleGroups",
                "scheduler:ListTagsForResource",
                "pipes:DescribePipe",
                "pipes:ListPipes",
                "pipes:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Les informations contenues dans cette section s'appliquent également à la politique `CloudWatchEventsReadOnlyAccess`. Cependant, il est fortement recommandé d'utiliser Amazon EventBridge au lieu d'Amazon CloudWatch Events.

EventBridge Politiques gérées spécifiques au schéma

Un schéma définit la structure des événements envoyés à EventBridge. EventBridge fournit des schémas pour tous les événements générés par les AWS services. Les politiques AWS gérées spécifiques aux EventBridge schémas suivantes sont disponibles :

- AmazonEventBridgeSchemasServiceRolePolicy
- AmazonEventBridgeSchemasFullAccess

- AmazonEventBridgeSchemasReadOnlyAccess

EventBridge Politiques gérées spécifiques au planificateur

Amazon EventBridge Scheduler est un planificateur sans serveur qui vous permet de créer, d'exécuter et de gérer des tâches à partir d'un service géré centralisé. Pour les politiques AWS gérées spécifiques au EventBridge planificateur, voir les politiques AWS gérées pour le planificateur dans le guide de l'utilisateur du EventBridge EventBridge planificateur.

EventBridge Politiques gérées spécifiques aux tuyaux

Amazon EventBridge Pipes connecte les sources d'événements aux cibles. Ils réduisent le besoin de connaissances spécialisées et de code d'intégration lors du développement d'architectures pilotées par les événements. Cela permet d'assurer la cohérence dans les applications de votre entreprise. Les politiques AWS gérées suivantes spécifiques à EventBridge Pipes sont disponibles :

- AmazonEventBridgePipesFullAccess

  Fournit un accès complet à Amazon EventBridge Pipes.

  > **Note**
  >
  > Cette politique prévoit que `iam:PassRole` — EventBridge Pipes a besoin de cette autorisation pour transmettre un rôle d'invocation EventBridge à la création et au démarrage de canaux.

- AmazonEventBridgePipesReadOnlyAccess

  Fournit un accès en lecture seule à Amazon EventBridge Pipes.

- AmazonEventBridgePipesOperatorAccess

  Fournit un accès en lecture seule et aux opérateurs (c'est-à-dire la possibilité d'arrêter et de démarrer l'exécution de Pipes) à Amazon EventBridge Pipes.

Rôles IAM pour l'envoi d'événements

Pour relayer des événements vers des cibles, un rôle IAM est EventBridge nécessaire.

Pour créer un rôle IAM pour envoyer des événements à EventBridge

1. Ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Pour créer un rôle IAM, suivez les étapes décrites dans la section Création d'un rôle pour déléguer des autorisations à un AWS service dans le Guide de l'utilisateur IAM. Au cours de ces étapes, procédez comme suit :

   - Dans Nom du rôle, utilisez un nom unique au sein de votre compte.
   - Dans Sélectionner le type de rôle, choisissez AWS Service Rôles, puis Amazon EventBridge. Cela donne EventBridge les autorisations nécessaires pour assumer le rôle.
   - Dans Attach Policy, sélectionnez AmazonEventBridgeFullAccess.

Vous pouvez également créer vos propres politiques IAM personnalisées pour autoriser les EventBridge actions et les ressources. Vous pouvez attacher ces politiques personnalisées aux utilisateurs ou groupes IAM qui nécessitent ces autorisations. Pour plus d'informations sur les politiques IAM, consultez Présentation des politiques IAM dans le Guide de l'utilisateur IAM. Pour plus d'informations sur la gestion et la création de politiques IAM personnalisées, consultez Gestion des politiques IAM dans le Guide de l'utilisateur IAM.

Autorisations requises pour accéder EventBridge aux cibles à l'aide de rôles IAM

EventBridge les cibles nécessitent généralement des rôles IAM qui accordent l'autorisation EventBridge d'invoquer la cible. Voici quelques exemples de différents AWS services et cibles. Pour les autres, utilisez la EventBridge console pour créer une règle et un nouveau rôle qui sera créé avec une politique avec des autorisations bien définies préconfigurées.

Amazon SQS, Amazon SNS, CloudWatch Lambda, Logs et les cibles de bus n'utilisent pas de rôles EventBridge , et les EventBridge autorisations doivent être accordées via une politique de ressources. Les cibles API Gateway peuvent utiliser des politiques de ressource ou des rôles IAM.

Si la cible est une destination d'API, le rôle que vous spécifiez doit inclure la politique suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "events:InvokeApiDestination" ],
            "Resource": [ "arn:aws:events:::api-destination/*" ]
```

```
        }
     ]
}
```

Si la cible est un flux Kinesis, le rôle utilisé pour envoyer les données d'événements à cette cible doit inclure la politique suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Si la cible est la fonctionnalité Exécuter la commande de Systems Manager et que vous spécifiez une ou plusieurs valeurs `InstanceIds` pour la commande, le rôle que vous spécifiez doit inclure la politique suivante.

```
{
     "Version": "2012-10-17",
     "Statement": [
          {
               "Action": "ssm:SendCommand",
               "Effect": "Allow",
               "Resource": [
                    "arn:aws:ec2:region:accountId:instance/instanceIds",
                    "arn:aws:ssm:region:*:document/documentName"
               ]
          }
     ]
}
```

Si la cible est la fonctionnalité Exécuter la commande de Systems Manager et que vous spécifiez une ou plusieurs balises pour la commande, le rôle que vous spécifiez doit inclure la politique suivante.

```
{
```

```
     "Version": "2012-10-17",
     "Statement": [
           {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ec2:region:accountId:instance/*"
                ],
                "Condition": {
                    "StringEquals": {
                        "ec2:ResourceTag/*": [
                            "[[tagValues]]"
                        ]
                    }
                }
           },
           {
                "Action": "ssm:SendCommand",
                "Effect": "Allow",
                "Resource": [
                     "arn:aws:ssm:region:*:document/documentName"
                ]
           }
     ]
}
```

Si la cible est une machine à AWS Step Functions états, le rôle que vous spécifiez doit inclure la politique suivante.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
             "Action": [ "states:StartExecution" ],
            "Resource": [ "arn:aws:states:*:*:stateMachine:*" ]
        }
     ]
}
```

Si la cible est une tâche Amazon ECS, le rôle que vous spécifiez doit inclure la politique suivante.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "ecs:RunTask"
        ],
        "Resource": [
            "arn:aws:ecs:*:account-id:task-definition/task-definition-name"
        ],
        "Condition": {
            "ArnLike": {
                "ecs:cluster": "arn:aws:ecs:*:account-id:cluster/cluster-name"
            }
        }
    },
    {
        "Effect": "Allow",
        "Action":"iam:PassRole",
        "Resource": [
            "*"
        ],
        "Condition": {
            "StringLike": {
                "iam:PassedToService": "ecs-tasks.amazonaws.com"
            }
        }
    }]
}
```

La politique suivante permet aux cibles intégrées EventBridge d'effectuer des actions Amazon EC2 en votre nom. Vous devez utiliser le AWS Management Console pour créer des règles avec des cibles intégrées.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "TargetInvocationAccess",
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:RebootInstances",
                "ec2:StopInstances",
```

```
                "ec2:TerminateInstances",
                "ec2:CreateSnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

La politique suivante permet EventBridge de relayer les événements vers les flux Kinesis de votre compte.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KinesisAccess",
            "Effect": "Allow",
            "Action": [
                "kinesis:PutRecord"
            ],
            "Resource": "*"
        }
    ]
}
```

Exemple de politique gérée par le client : utilisation du balisage pour contrôler l’accès aux règles

L'exemple suivant montre une politique utilisateur qui accorde des autorisations pour les EventBridge actions. Cette politique fonctionne lorsque vous utilisez l' EventBridge API, AWS les SDK ou le AWS CLI.

Vous pouvez autoriser les utilisateurs à accéder à des EventBridge règles spécifiques tout en les empêchant d'accéder à d'autres règles. Pour ce faire, balisez les deux ensembles de règles et utilisez des politiques IAM qui fassent référence à ces balises. Pour plus d'informations sur le balisage EventBridge des ressources, consultez EventBridge Balises Amazon.

Vous pouvez accorder une politique IAM à un utilisateur pour autoriser l’accès aux seules règles disposant d’une balise déterminée. Pour choisir les règles auxquelles vous accordez accès, associez-les à cette balise. Par exemple, la politique suivante accorde un accès utilisateur aux règles dont la clé de balise `Stack` a la valeur `Prod`.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "events:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Stack": "Prod"
                }
            }
        }
    ]
}
```

Pour plus d'informations sur l'utilisation des instructions de politique IAM, consultez Contrôle de l'accès à l'aide des politiques dans le Guide de l'utilisateur IAM.

Amazon EventBridge met à jour AWS ses politiques gérées

Consultez les détails des mises à jour des politiques AWS gérées EventBridge depuis que ce service a commencé à suivre ces modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page Historique du EventBridge document.

Modification	Description	Date
AmazonEventBridgeFullAccess— Politique mise à jour	AWS GovCloud (US) Regions uniquement L'autorisation suivante n'est pas incluse, car elle n'est pas utilisée : • `iam:CreateServiceLinkedRole` autorisation pour le registre des EventBridge schémas	9 mai 2024

Modification	Description	Date
AmazonEventBridgeSchemasFullAccess— Politique mise à jour	AWS GovCloud (US) Regions uniquement L'autorisation suivante n'est pas incluse, car elle n'est pas utilisée : • `iam:CreateServiceLinkedRole` autorisation pour le registre des EventBridge schémas	9 mai 2024
AmazonEventBridgePipesFullAccess— Ajout d'une nouvelle politique	EventBridge ajout d'une politique gérée pour les autorisations complètes d'utilisation de EventBridge Pipes.	1er décembre 2022
AmazonEventBridgePipesReadOnlyAccess— Ajout d'une nouvelle politique	EventBridge ajout d'une politique gérée pour les autorisations permettant de consulter les ressources d'information de EventBridge Pipes.	1er décembre 2022
AmazonEventBridgePipesOperatorAccess— Ajout d'une nouvelle politique	EventBridge ajout d'une politique gérée pour les autorisations permettant d'afficher les informations sur les EventBridge tuyaux, ainsi que de démarrer et d'arrêter l'exécution des tuyaux.	1er décembre 2022

Modification	Description	Date
AmazonEventBridgeFullAccess – Mise à jour d'une politique existante	EventBridge a mis à jour la politique pour inclure les autorisations nécessaires à l'utilisation EventBridge des fonctionnalités de Pipes.	1er décembre 2022
AmazonEventBridgeReadOnlyAccess – Mise à jour d'une politique existante	EventBridge autorisations supplémentaires nécessaires pour consulter les ressources d'information de EventBridge Pipes. Les actions suivantes ont été ajoutées : • `pipes:DescribePipe` • `pipes:ListPipes` • `pipes:ListTagsForResource`	1er décembre 2022
CloudWatchEventsReadOnlyAccess – Mise à jour d'une politique existante	Mis à jour pour correspondre AmazonEventBridgeReadOnlyAccess.	1er décembre 2022
CloudWatchEventsFullAccess – Mise à jour d'une politique existante	Mis à jour pour correspondre AmazonEventBridgeFullAccess.	1er décembre 2022

Modification	Description	Date
AmazonEventBridgeFullAccess – Mise à jour d'une politique existante	EventBridge a mis à jour la politique pour inclure les autorisations nécessaires à l'utilisation des schémas et des fonctionnalités du planificateur. Les autorisations suivantes ont été ajoutées : • EventBridge Actions du registre des schémas • EventBridge Actions du planificateur • `iam:CreateServiceLinkedRole` autorisation pour le registre des EventBridge schémas • `iam:PassRole` autorisation pour EventBridge Scheduler	10 novembre 2022

Modification	Description	Date
AmazonEventBridgeReadOnlyAccess – Mise à jour d'une politique existante	EventBridge autorisations supplémentaires nécessaires pour afficher les ressources d'informations sur le schéma et le planificateur. Les actions suivantes ont été ajoutées : • `schemas:DescribeCodeBinding` • `schemas:DescribeDiscoverer` • `schemas:DescribeRegistry` • `schemas:DescribeSchema` • `schemas:ExportSchema` • `schemas:GetCodeBindingSource` • `schemas:GetDiscoveredSchema` • `schemas:GetResourcePolicy` • `schemas:ListDiscoverers` • `schemas:ListRegistries` • `schemas:ListSchemas` • `schemas:ListSchemaVersions`	10 novembre 2022

Modification	Description	Date
	• `schemas:ListTagsForResource` • `schemas:SearchSchemas` • `scheduler:GetSchedule` • `scheduler:GetScheduleGroup` • `scheduler:ListSchedules` • `scheduler:ListScheduleGroups` • `scheduler:ListTagsForResource`	
AmazonEventBridgeReadOnlyAccess – Mise à jour d'une politique existante	EventBridge autorisations supplémentaires nécessaires pour afficher les informations du point de terminaison. Les actions suivantes ont été ajoutées : • `events:ListEndpoints` • `events:DescribeEndpoint`	7 avril 2022

Modification	Description	Date
AmazonEventBridgeReadOnlyAccess – Mise à jour d’une politique existante	EventBridge autorisations supplémentaires nécessaires pour afficher les informations de connexion et de destination de l'API. Les actions suivantes ont été ajoutées : • `events:DescribeConnection` • `events:ListConnections` • `events:DescribeApiDestination` • `events:ListApiDestinations`	4 mars 2021

Modification	Description	Date
AmazonEventBridgeFullAccess – Mise à jour d'une politique existante	EventBridge a mis à jour la politique pour inclure les destinations d'API `iam:CreateServiceLinkedRole` et les AWS Secrets Manager autorisations nécessaires à leur utilisation. Les actions suivantes ont été ajoutées : • `secretsmanager:CreateSecret` • `secretsmanager:UpdateSecret` • `secretsmanager:DeleteSecret` • `secretsmanager:GetSecretValue` • `secretsmanager:PutSecretValue`	4 mars 2021
EventBridge a commencé à suivre les modifications	EventBridge a commencé à suivre les modifications apportées AWS à ses politiques gérées.	4 mars 2021

Utilisation de politiques basées sur les ressources pour Amazon EventBridge

Quand une règle s'exécute dans EventBridge, toutes les cibles qui lui sont associées sont invoquées. Les règles peuvent invoquer des fonctions AWS Lambda, publier dans des rubriques Amazon SNS ou relayer l'événement vers des flux Kinesis. Pour pouvoir effectuer des appels sur les ressources dont vous êtes propriétaire, EventBridge a besoin des autorisations appropriées. Pour les ressources Lambda, Amazon SNS, Amazon SQS et Amazon CloudWatch Logs, EventBridge utilise des politiques basées sur les ressources. Pour les flux Kinesis, EventBridge utilise des politiques basées sur l'identité.

Vous devez utiliser l'interface AWS CLI pour ajouter des autorisations à vos cibles. 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.

Rubriques

Autorisations Amazon API Gateway

Pour invoquer votre point de terminaison Amazon API Gateway à l'aide d'une règle EventBridge, ajoutez l'autorisation suivante à la politique du point de terminaison API Gateway.

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

```
            },
            "Action": "execute-api:Invoke",
            "Condition": {
                "ArnEquals": {
                    "aws:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
                }
            },
            "Resource": [
                "execute-api:/stage/GET/api"
            ]
        }
    ]
}
```

Autorisations CloudWatch Logs

Quand CloudWatch Logs est la cible d'une règle, EventBridge crée des flux de journaux, et CloudWatch Logs stocke le texte des événements sous forme d'entrées de journal. Pour permettre EventBridge de créer les flux de journaux et de journaliser les événements, CloudWatch doit inclure une politique basée sur les ressources qui permette à EventBridge d'écrire dans CloudWatch Logs.

Si vous utilisez la AWS Management Console pour ajouter CloudWatch Logs en tant que cible d'une règle, la politique basée sur les ressources est créée automatiquement. Si vous utilisez l'interface AWS CLI pour ajouter la cible et que la politique n'existe pas déjà, vous devez la créer.

L'exemple suivant autorise EventBridge à écrire dans tous les groupes de journaux dont le nom commence par `/aws/events/`. Si vous utilisez une autre politique de dénomination pour ces types de journaux, ajustez l'exemple en conséquence.

```
{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": ["events.amazonaws.com", "delivery.logs.amazonaws.com"]
            },
            "Resource": "arn:aws:logs:region:account:log-group:/aws/events/*:*",
            "Sid": "TrustEventsToStoreLogEvent"
```

```
        }
    ],
    "Version": "2012-10-17"
}
```

Pour plus d'informations, consultez PutResourcePolicy dans le Guide de référence des API CloudWatch Logs.

Autorisations AWS Lambda

Pour invoquer votre fonction AWS Lambda à l'aide d'une règle EventBridge, ajoutez l'autorisation suivante à la politique de votre fonction Lambda.

```
{
  "Effect": "Allow",
  "Action": "lambda:InvokeFunction",
  "Resource": "arn:aws:lambda:region:account-id:function:function-name",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Condition": {
    "ArnLike": {
      "AWS:SourceArn": "arn:aws:events:region:account-id:rule/rule-name"
    }
  },
  "Sid": "InvokeLambdaFunction"
}
```

Pour ajouter l'autorisation ci-dessus qui permet à EventBridge d'invoquer des fonctions Lambda à l'aide de l'interface AWS CLI

- A partir d'une invite de commande, entrez la commande suivante.

```
aws lambda add-permission --statement-id "InvokeLambdaFunction" \
--action "lambda:InvokeFunction" \
--principal "events.amazonaws.com" \
--function-name "arn:aws:lambda:region:account-id:function:function-name" \
--source-arn "arn:aws:events:region:account-id:rule/rule-name"
```

Pour plus d'informations sur la définition d'autorisations qui permettent à EventBridge d'invoquer des fonctions Lambda, consultez AddPermission et Utilisation de Lambda avec des événements planifiés dans le Guide du développeur AWS Lambda.

Autorisations Amazon SNS

Pour permettre à EventBridge de publier dans une rubrique Amazon SNS, utilisez les commandes `aws sns get-topic-attributes` et `aws sns set-topic-attributes`.

> **Note**
>
> Vous ne pouvez pas utiliser de blocs `Condition` dans les politiques de rubriques Amazon SNS pour EventBridge.

Pour ajouter les autorisations qui permettent à EventBridge de publier dans des rubriques SNS

1. Pour afficher la liste des attributs d'une rubrique SNS, utilisez la commande suivante.

   ```
   aws sns get-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
   ```

   L'exemple suivant montre le résultat d'une nouvelle rubrique SNS.

   ```
   {
       "Attributes": {
           "SubscriptionsConfirmed": "0",
           "DisplayName": "",
           "SubscriptionsDeleted": "0",
           "EffectiveDeliveryPolicy": "{\"http\":{\"defaultHealthyRetryPolicy\":
   {\"minDelayTarget\":20,\"maxDelayTarget\":20,\"numRetries\":3,\"numMaxDelayRetries
   \":0,\"numNoDelayRetries\":0,\"numMinDelayRetries\":0,\"backoffFunction\":\"linear
   \"},\"disableSubscriptionOverrides\":false}}",
           "Owner": "account-id",
           "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
   \"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
   \":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
   \",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",
   \"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\"],\"Resource
   \":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals\":
   {\"AWS:SourceOwner\":\"account-id\"}}}]}",
           "TopicArn": "arn:aws:sns:region:account-id:topic-name",
   ```

```
        "SubscriptionsPending": "0"
    }
}
```

2. Utilisez un convertisseur de JSON en chaîne pour convertir la déclaration suivante en chaîne.

```
{
  "Sid": "PublishEventsToMyTopic",
  "Effect": "Allow",
  "Principal": {
    "Service": "events.amazonaws.com"
  },
  "Action": "sns:Publish",
  "Resource": "arn:aws:sns:region:account-id:topic-name"
}
```

   Après avoir converti la déclaration en chaîne, elle se présente comme suit.

```
{\"Sid\":\"PublishEventsToMyTopic\",\"Effect\":\"Allow\",\"Principal\":
{\"Service\":\"events.amazonaws.com\"},\"Action\":\"sns:Publish\",\"Resource\":
\"arn:aws:sns:region:account-id:topic-name\"}
```

3. Ajoutez la chaîne que vous avez créée à l'étape précédente à la collection `"Statement"` à l'intérieur de l'attribut `"Policy"`.
4. Utilisez la commande `aws sns set-topic-attributes` pour définir la nouvelle politique.

```
aws sns set-topic-attributes --topic-arn "arn:aws:sns:region:account-id:topic-name"
 \
--attribute-name Policy \
--attribute-value "{\"Version\":\"2012-10-17\",\"Id\":\"__default_policy_ID\",
\"Statement\":[{\"Sid\":\"__default_statement_ID\",\"Effect\":\"Allow\",\"Principal
\":{\"AWS\":\"*\"},\"Action\":[\"SNS:GetTopicAttributes\",\"SNS:SetTopicAttributes
\",\"SNS:AddPermission\",\"SNS:RemovePermission\",\"SNS:DeleteTopic\",
\"SNS:Subscribe\",\"SNS:ListSubscriptionsByTopic\",\"SNS:Publish\"],\"Resource
\":\"arn:aws:sns:region:account-id:topic-name\",\"Condition\":{\"StringEquals
\":{\"AWS:SourceOwner\":\"account-id\"}}}, {\"Sid\":\"PublishEventsToMyTopic\",
\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"events.amazonaws.com\"},\"Action
\":\"sns:Publish\",\"Resource\":\"arn:aws:sns:region:account-id:topic-name\"}]}"
```

Pour plus d'informations, consultez l'action SetTopicAttributes dans la Référence des API Amazon Simple Notification Service.

Autorisations Amazon SQS

Pour permettre à une règle EventBridge d'invoquer une file d'attente Amazon SQS, utilisez les commandes `aws sqs get-queue-attributes` et `aws sqs set-queue-attributes`.

Si la politique de la file d'attente SQS est vide, vous devez d'abord créer une politique et y ajouter la déclaration d'autorisations. La politique d'une nouvelle file d'attente SQS est vide.

Si la file d'attente SQS possède déjà une politique, vous devez copier la politique d'origine et la combiner avec une nouvelle déclaration pour y ajouter la déclaration d'autorisations.

Pour ajouter les autorisations qui permettent à des règles EventBridge d'invoquer une file d'attente SQS

1. Pour afficher la liste des attributs d'une file d'attente SQS. A partir d'une invite de commande, entrez la commande suivante.

```
aws sqs get-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attribute-names Policy
```

2. Ajoutez la déclaration suivante.

```
{
      "Sid": "AWSEvents_custom-eventbus-ack-sqs-rule_dlq_sqs-rule-target",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sqs:SendMessage",
      "Resource": "arn:aws:sqs:region:account-id:queue-name",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:events:region:account-id:rule/bus-name/rule-
name"
        }
      }
    }
```

3. Utilisez un convertisseur de JSON en chaîne pour convertir la déclaration précédente en chaîne. Après avoir converti la politique en chaîne, elle se présente comme suit.

```
{\"Sid\": \"EventsToMyQueue\", \"Effect\": \"Allow\", \"Principal\": {\"Service
\": \"events.amazonaws.com\"}, \"Action\": \"sqs:SendMessage\", \"Resource\":
 \"arn:aws:sqs:region:account-id:queue-name\", \"Condition\": {\"ArnEquals\":
 {\"aws:SourceArn\": \"arn:aws:events:region:account-id:rule/rule-name\"}}
```

4. Créez un fichier nommé `set-queue-attributes.json` avec le contenu suivant.

```
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"arn:aws:sqs:region:account-
id:queue-name/SQSDefaultPolicy\",\"Statement\":[{\"Sid\": \"EventsToMyQueue\",
 \"Effect\": \"Allow\", \"Principal\": {\"Service\": \"events.amazonaws.com\"},
 \"Action\": \"sqs:SendMessage\", \"Resource\": \"arn:aws:sqs:region:account-
id:queue-name\", \"Condition\": {\"ArnEquals\": {\"aws:SourceArn\":
 \"arn:aws:events:region:account-id:rule/rule-name\"}}}]}"
}
```

5. Définissez l'attribut policy en utilisant le fichier `set-queue-attributes.json` que vous venez de créer en tant qu'entrée, comme dans la commande suivante.

```
aws sqs set-queue-attributes \
--queue-url https://sqs.region.amazonaws.com/account-id/queue-name \
--attributes file://set-queue-attributes.json
```

Pour plus d'informations, consultez Exemples de politiques Amazon SQS dans le Guide du développeur Amazon Simple Queue Service.

Particularités pour EventBridge Pipes

EventBridge Pipes ne prend pas en charge les politiques basées sur les ressources et ne dispose pas d'API qui prennent en charge les conditions des politiques basées sur les ressources.

Prévention du problème de l'adjoint confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. Dans AWS, l'emprunt d'identité entre services peut entraîner le problème de député confus. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le

service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous vous recommandons d'utiliser les clés de contexte de condition globale `aws:SourceArn` et `aws:SourceAccount` dans les politiques de ressources afin de limiter les autorisations à la ressource octroyées par Amazon EventBridge à un autre service. Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services. Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Si la valeur `aws:SourceArn` ne contient pas l'ID du compte, tel qu'un ARN de compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations.

Bus d'événements

Pour les cibles de la règle de bus d'événements EventBridge, la valeur de `aws:SourceArn` doit être l'ARN de la règle.

L'exemple suivant montre comment utiliser les clés de contexte de condition globale `aws:SourceArn` et `aws:SourceAccount` dans EventBridge afin d'éviter le problème de l'adjoint confus. Cet exemple est destiné à être utilisé dans une politique d'approbation de rôle, pour un rôle utilisé par une règle EventBridge.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
```

```
    "Action": "sts:AssumeRole"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:events:*:123456789012:rule/myRule"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

EventBridge Pipes

Pour EventBridge Pipes, la valeur de `aws:SourceArn` doit être l'ARN du canal.

L'exemple suivant montre comment utiliser les clés de contexte de condition globale `aws:SourceArn` et `aws:SourceAccount` dans EventBridge afin d'éviter le problème de l'adjoint confus. Cet exemple est destiné à être utilisé dans une politique d'approbation de rôle, pour un rôle utilisé par EventBridge Pipes.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:pipe:*:123456789012::pipe/example"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Politiques basées sur les ressources pour les schémas Amazon EventBridge

Le registre de schémas EventBridge prend en charge les politiques basées sur les ressources. Une politique basée sur les ressources est une politique qui est attachée à une ressource et non à une identité IAM. Par exemple, dans Amazon Simple Storage Service (Amazon S3), une politique de ressource est attachée à un compartiment Amazon S3.

Pour plus d'informations sur les schémas EventBridge et les politiques basées sur les ressources, consultez les documents suivants.

- Référence des API REST de schémas Amazon EventBridge
- Politiques basées sur l'identité et politiques basées sur les ressources dans le guide de l'utilisateur IAM

API prises en charge pour les politiques basées sur les ressources

Voici les API que vous pouvez utiliser avec les politiques basées sur les ressources pour le registre de schémas EventBridge.

- `DescribeRegistry`
- `UpdateRegistry`
- `DeleteRegistry`
- `ListSchemas`
- `SearchSchemas`
- `DescribeSchema`
- `CreateSchema`
- `DeleteSchema`
- `UpdateSchema`
- `ListSchemaVersions`
- `DeleteSchemaVersion`
- `DescribeCodeBinding`
- `GetCodeBindingSource`
- `PutCodeBinding`

Exemple de politique accordant toutes les actions prises en charge à un compte AWS

Pour le registre de schémas EventBridge, vous devez toujours attacher une politique basée sur les ressources à un registre. Pour accorder un accès à un schéma, vous devez spécifier l'ARN du schéma et l'ARN du registre dans la politique.

Pour accorder à un utilisateur un accès à toutes les API disponibles pour les schémas EventBridge, utilisez une politique similaire à la suivante, en remplaçant la valeur de `"Principal"` par l'ID du compte auquel vous souhaitez accorder l'accès.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
            "Effect": "Allow",
            "Action": [
                "schemas:*"
            ],
            "Principal": {
                "AWS": [
                    "109876543210"
                ]
            },
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
            ]
        }
    ]
}
```

Exemple de politique accordant les actions en lecture seule à un compte AWS

L'exemple suivant accorde un accès à un compte uniquement pour les API en lecture seule qui concernent les schémas EventBridge.

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

```
            "Effect": "Allow",
            "Action": [
                "schemas:DescribeRegistry",
                "schemas:ListSchemas",
                "schemas:SearchSchemas",
                "schemas:DescribeSchema",
                "schemas:ListSchemaVersions",
                "schemas:DescribeCodeBinding",
                "schemas:GetCodeBindingSource"
            ],
            "Principal": {
                "AWS": [
                    "109876543210"
                ]
            },
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
            ]
        }
    ]
}
```

Exemple de politique accordant toutes les actions à une organisation

Vous pouvez utiliser des politiques basées sur les ressources avec le registre de schémas EventBridge pour accorder un accès à une organisation. Pour plus d'informations, consultez le Guide de l'utilisateur AWS Organizations. L'exemple suivant accorde à l'organisation ayant pour ID `o-a1b2c3d4e5` un accès au registre de schémas.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Test",
            "Effect": "Allow",
            "Action": [
                "schemas:*"
            ],
            "Principal": "*",
            "Resource": [
                "arn:aws:schemas:us-east-1:012345678901:registry/default",
                "arn:aws:schemas:us-east-1:012345678901:schema/default*"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": [
                        "o-a1b2c3d4e5"
                    ]
                }
            }
        }
    ]
}
```

Informations de référence sur les autorisations Amazon EventBridge

Pour spécifier une action dans une politique EventBridge, utilisez le préfixe `events:` suivi du nom de l'opération d'API, comme l'illustre l'exemple suivant.

```
"Action": "events:PutRule"
```

Pour spécifier plusieurs actions dans une seule instruction, séparez-les par des virgules, comme suit :

```
"Action": ["events:action1", "events:action2"]
```

Pour spécifier plusieurs actions, vous pouvez également utiliser des caractères génériques. Par exemple, vous pouvez spécifier toutes les actions qui commencent par le mot `"Put"` comme suit.

```
"Action": "events:Put*"
```

Pour spécifier toutes les actions d'API EventBridge, utilisez le caractère générique `*` comme suit.

```
"Action": "events:*"
```

Le tableau suivant répertorie les opérations d'API EventBridge et les actions correspondantes que vous pouvez spécifier dans une politique IAM.

Opération d'API EventBridge	Autorisations nécessaires	Description
DeleteRule	`events:DeleteRule`	Requise pour supprimer une règle.
DescribeEventBus	`events:DescribeEventBus`	Obligatoire pour répertorier les comptes qui sont autorisés à écrire des événements dans le bus d'événements du compte actuel.
DescribeRule	`events:DescribeRule`	Requise pour répertorier les détails relatifs à une règle.

Opération d'API EventBridge	Autorisations nécessaires	Description
DisableRule	`events:DisableRule`	Requise pour désactiver une règle.
EnableRule	`events:EnableRule`	Requise pour activer une règle.
ListRuleNamesByTarget	`events:ListRuleNamesByTarget`	Requise pour répertorier les règles associées à une cible.
ListRules	`events:ListRules`	Requise pour répertorier toutes les règles de votre compte.
ListTagsForResource	`events:ListTagsForResource`	Requise pour répertorier toutes les balises associées à une ressource EventBridge. Actuellement, seules les règles peuvent être balisées.
ListTargetsByRule	`events:ListTargetsByRule`	Requise pour afficher toutes les cibles associées à une règle.
PutEvents	`events:PutEvents`	Requise pour ajouter des événements personnalisés qui peuvent être associés à des règles.
PutPermission	`events:PutPermission`	Obligatoire pour accorder à un autre compte l'autorisation d'écrire des événements dans le bus d'événement par défaut du compte.
PutRule	`events:PutRule`	Requise pour créer ou mettre à jour une règle.

Opération d'API EventBridge	Autorisations nécessaires	Description
PutTargets	`events:PutTargets`	Requise pour ajouter des cibles à une règle.
RemovePermission	`events:RemovePermission`	Obligatoire pour révoquer à un autre compte l'autorisation d'écrire des événements dans le bus d'événement par défaut du compte.
RemoveTargets	`events:RemoveTargets`	Requise pour supprimer une cible d'une règle.
TestEventPattern	`events:TestEventPattern`	Requise pour tester un modèle d'événement par rapport à un événement donné.

Utilisation de conditions de politique IAM pour un contrôle d'accès précis

Lorsque vous accordez des autorisations, vous utilisez le langage de politique IAM dans une déclaration de politique pour spécifier les conditions d'application d'une politique. Par exemple, vous pouvez faire en sorte qu'une politique ne s'applique qu'après une date donnée.

Dans une politique, une condition est constituée de paires clé-valeur. Les clés de condition ne sont pas sensibles à la casse.

Si vous spécifiez plusieurs conditions ou clés dans une même condition, l'intégralité des conditions et des clés doivent être réunies pour qu'EventBridge accorde l'autorisation. Si vous spécifiez une seule condition avec plusieurs valeurs pour une même clé, EventBridge accorde l'autorisation si l'une des valeurs est respectée.

Vous pouvez aussi utiliser des espaces réservés ou des variables de politique lors de la spécification de conditions. Pour plus d'informations, consultez Variables de stratégie dans le IAM Guide de l'utilisateur. Pour plus d'informations sur la spécification de conditions dans un langage de politique IAM, consultez Condition dans le Guide de l'utilisateur IAM.

Par défaut, les rôles et les utilisateurs IAM ne peuvent pas accéder aux événements relevant de votre compte. Pour accéder à ces événements, un utilisateur doit être autorisé à exécuter l'action d'API `PutRule`. Si un utilisateur ou un rôle IAM est autorisé à exécuter l'action `events:PutRule`, il peut créer une règle qui corresponde à certains événements. Cependant, pour que la règle soit utile, l'utilisateur doit également disposer des autorisations nécessaires pour l'action `events:PutTargets`, car si vous voulez que la règle fasse plus que publier une métrique CloudWatch, vous devez également ajouter une cible à une règle.

Vous pouvez spécifier une condition dans la déclaration de politique d'un utilisateur ou d'un rôle IAM permettant à l'utilisateur ou au rôle de créer une règle qui corresponde uniquement à un ensemble spécifique de sources et de types d'événements. Pour accorder l'accès à des sources et des types d'événements spécifiques, utilisez les clés de condition `events:source` et `events:detail-type`.

De la même façon, vous pouvez spécifier une condition dans la déclaration de politique d'un utilisateur ou d'un rôle IAM permettant à l'utilisateur ou au rôle de créer une règle qui corresponde uniquement à une ressource spécifique dans vos comptes. Pour accorder l'accès à une ressource spécifique, utilisez la clé de condition `events:TargetArn`.

L'exemple suivant est une politique qui permet aux utilisateurs d'accéder à tous les événements dans EventBridge, à l'exception des événements Amazon EC2, comme l'indique l'instruction de refus (Deny) dans l'action d'API `PutRule`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyPutRuleForAllEC2Events",
            "Effect": "Deny",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

Clés de condition EventBridge

Le tableau suivant présente les clés de condition et les paires clé-valeur que vous pouvez utiliser dans une politique dans EventBridge.

Clé de condition	Paire clé-valeur	Types d'évaluation
aws:SourceAccount	Compte dans lequel se trouve la règle spécifiée par `aws:SourceArn` .	Account Id, Null
aws:SourceArn	ARN de la règle qui envoie l'événement.	ARN, Null
events:creatorAccount	`"events:creatorAccount":"` *`creatorAccount`* `"` Pour *`creatorAccount`* , utilisez l'ID du compte dans lequel la règle a été créée. Utilisez cette condition pour autoriser les appels d'API sur les règles d'un compte spécifique.	creatorAccount, Null

Clé de condition	Paire clé-valeur	Types d'évaluation
events:detail-type	`"events:detail-type":" detail-type "` Où *detail-type* est la chaîne littérale pour le champ detail-type de l'événement, par exemple, `"AWS API Call via CloudTrail"` et `"EC2 Instance State-change Notification"` .	Type de détail, null
events: detail.eventTypeCode	`"events:detail.eventTypeCode":" eventTypeCode "` Pour *eventTypeCode* , utilisez la chaîne littérale pour le champ detail.eventTypeCode de l'événement, par exemple `"AWS_ABUSE_DOS_REPORT"` .	eventTypeCode, Null
events: detail.service	`"events:detail.service":" service"` Pour *service*, utilisez la chaîne littérale pour le champ detail.service de l'événement, par exemple `"ABUSE"`.	service, Null

Clé de condition	Paire clé-valeur	Types d'évaluation
events: detail.userIdentity.principalId	`"events:detail.userIdentity.principalId":" principal-id "` Pour *principal-id* , utilisez la chaîne littérale pour le champ detail.userIdentity.principalId de l'événement avec le detail-type `"AWS API Call via CloudTrail"` , par exemple `"AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName."` .	ID du mandataire, null
events:eventBusInvocation	`"events:eventBusInvocation":" boolean"` Pour *boolean*, utilisez true lorsqu'une règle envoie un événement à une cible qui correspond à un bus d'événements dans un autre compte. Utilisez false lorsqu'un appel d'API `PutEvents` est utilisé.	eventBusInvocation, Null
events:ManagedBy	Utilisé en interne par les services AWS. Pour une règle créée par un service AWS en votre nom, la valeur correspond au nom de principal du service qui a créé la règle.	Non destiné à être utilisé dans les politiques des clients.

Clé de condition	Paire clé-valeur	Types d'évaluation
events:source	`"events:source":" source "` Utilisez *source* pour la chaîne littérale du champ source de l'événement, par exemple `"aws.ec2"` et `"aws.s3"`. Pour voir les autres valeurs possibles de *source*, consultez les exemples d'événements dans Événements organisés par AWS les services.	Source, null
events:TargetArn	`"events:TargetArn" :" target-arn "` Pour *target-arn* , utilisez l'ARN de la cible de la règle, par exemple `"arn:aws: lambda:*:*:functio n:*"` .	ArrayOfARN, Null

Pour voir des exemples de déclarations de politique pour EventBridge, consultez Gestion des autorisations d'accès à vos ressources Amazon EventBridge.

Rubriques

- Particularités pour EventBridge Pipes
- Exemple : utilisation de la condition creatorAccount
- Exemple : utilisation de la condition eventBusInvocation
- Exemple : limitation de l'accès à une source spécifique
- Exemple : définition de plusieurs sources pouvant chacune être utilisée individuellement dans un modèle d'événement
- Exemple : définition d'une source et d'un DetailType pouvant être utilisés dans un modèle d'événement

- Exemple : vérification de la définition de la source dans le modèle d'événement
- Exemple : définition d'une liste de sources autorisées dans un modèle d'événement à plusieurs sources
- Exemple : limitation de l'accès de PutRule par detail.service
- Exemple : limitation de l'accès de PutRule par detail.eventTypeCode
- Exemple : vérification que seuls sont autorisés les événements AWS CloudTrail pour les appels d'API émanant d'un certain PrincipalId
- Exemple : limitation de l'accès aux cibles

Particularités pour EventBridge Pipes

EventBridge Pipes ne prend pas en charge d'autres clés de condition de politique IAM.

Exemple : utilisation de la condition **`creatorAccount`**

L'exemple de déclaration de politique ci-dessous montre comment utiliser la condition `creatorAccount` dans une politique pour n'autoriser la création de règles que si le compte spécifié comme `creatorAccount` est le compte dans lequel la règle a été créée.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForOwnedRules",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "events:creatorAccount": "${aws:PrincipalAccount}"
                }
            }
        }
    ]
}
```

Exemple : utilisation de la condition **`eventBusInvocation`**

`eventBusInvocation` indique si l'invocation provient d'une cible intercompte ou d'une demande d'API `PutEvents`. La valeur est true lorsque l'invocation résulte d'une règle incluant une cible intercompte, par exemple lorsque la cible est un bus d'événements dans un autre compte. La valeur est false lorsque l'invocation résulte d'une demande d'API `PutEvents`. L'exemple suivant illustre une invocation en provenance d'une cible intercompte.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCrossAccountInvocationEventsOnly",
      "Effect": "Allow",
      "Action": "events:PutEvents",
      "Resource": "*",
      "Condition": {
        "BoolIfExists": {
          "events:eventBusInvocation": "true"
        }
      }
    }
  ]
}
```

Exemple : limitation de l'accès à une source spécifique

Les politiques suivantes peuvent être attachées à un utilisateur IAM. La politique A autorise l'action d'API `PutRule` pour tous les événements, tandis que la politique B n'autorise `PutRule` que si le modèle d'événement de la règle créée correspond à des événements Amazon EC2.

Politique A : autoriser tous les événements

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEvents",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*"
        }
```

```
    ]
    }
```

Politique B : autoriser les événements d'Amazon EC2 uniquement

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleForAllEC2Events",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2"
                }
            }
        }
    ]
}
```

`EventPattern` est un argument obligatoire pour `PutRule`. Par conséquent, si l'utilisateur utilisant la stratégie B appelle `PutRule` avec un modèle d'événement tel que le suivant.

```
{
    "source": [ "aws.ec2" ]
}
```

La règle sera créée, car la stratégie autorise cette source spécifique, à savoir `"aws.ec2"`. Toutefois, si l'utilisateur avec la stratégie B appelle `PutRule` avec un modèle d'événement tel que le suivant, la création de la règle est refusée, car la stratégie n'autorise pas cette source spécifique : c'est-à-dire, `"aws.s3"`.

```
{
    "source": [ "aws.s3" ]
}
```

Globalement, l'utilisateur soumis à la politique B est autorisé uniquement à créer une règle pouvant correspondre à des événements originaires d'Amazon EC2. Par conséquent, cet utilisateur est autorisé uniquement à accéder aux événements en provenance d'Amazon EC2.

Consultez le tableau suivant pour comparer la stratégie A et la stratégie B.

Modèle d'événement	Autorisé par la stratégie A	Autorisé par la stratégie B
`{ "source": [ "aws.ec2" ] }`	Oui	Oui
`{ "source": [ "aws.ec2", "aws.s3" ] }`	Oui	Non (la source aws.s3 n'est pas autorisée)
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Oui	Oui
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Oui	Non (la source doit être spécifiée)

Exemple : définition de plusieurs sources pouvant chacune être utilisée individuellement dans un modèle d'événement

La politique suivante permet à un utilisateur ou un rôle IAM de créer une règle dont la source dans `EventPattern` est Amazon EC2 ou Amazon ECS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsEC2OrECS",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": [ "aws.ec2", "aws.ecs" ]
                }
            }
        }
    ]
}
```

Le tableau suivant présente des exemples de modèles d'événements qui sont autorisés ou refusés par cette politique.

Modèle d'événement	Autorisé par la politique
`{ "source": [ "aws.ec2" ] }`	Oui
`{ "source": [ "aws.ecs" ] }`	Oui
`{ "source": [ "aws.s3" ] }`	Non
`{ "source": [ "aws.ec2", "aws.ecs" ] }`	Non

Modèle d'événement	Autorisé par la politique

<pre>{
"detail-type": ["AWS API Call via CloudTrail"]
}</pre>
	Non

Exemple : définition d'une source et d'un **`DetailType`** pouvant être utilisés dans un modèle d'événement

La politique suivante autorise les événements uniquement en provenance de la source `aws.ec2` et dont `DetailType` a la valeur `EC2 instance state change notification`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid":
 "AllowPutRuleIfSourceIsEC2AndDetailTypeIsInstanceStateChangeNotification",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:source": "aws.ec2",
                    "events:detail-type": "EC2 Instance State-change Notification"
                }
            }
        }
    ]
}
```

Le tableau suivant présente des exemples de modèles d'événements qui sont autorisés ou refusés par cette politique.

Modèle d'événement	Autorisé par la politique
<pre>{</pre>	Non

Modèle d'événement	Autorisé par la politique
`"source": [ "aws.ec2" ] }`	
`{ "source": [ "aws.ecs" ] }`	Non
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Oui
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance Health Failed" ] }`	Non
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Non

Exemple : vérification de la définition de la source dans le modèle d'événement

La politique suivante permet aux utilisateurs de créer uniquement des règles avec la présence du champ source dans `EventPatterns`. Avec cette politique, un utilisateur ou un rôle IAM ne peut pas créer de règle avec un `EventPattern` qui n'indique pas de source spécifique.

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

```
            "Sid": "AllowPutRuleIfSourceIsSpecified",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

Le tableau suivant présente des exemples de modèles d'événements qui sont autorisés ou refusés par cette politique.

Modèle d'événement	Autorisé par la stratégie
`{ "source": [ "aws.ec2" ], "detail-type": [ "EC2 Instance State-change Notification" ] }`	Oui
`{ "source": [ "aws.ecs", "aws.ec2" ] }`	Oui
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Non

Exemple : définition d'une liste de sources autorisées dans un modèle d'événement à plusieurs sources

La politique suivante permet aux utilisateurs de créer des règles avec plusieurs sources définies dans `EventPatterns`. Chaque source figurant dans le modèle d'événement doit être membre de la liste fournie dans la condition. Lorsque vous utilisez la condition `ForAllValues`, veillez à ce qu'au moins un des éléments de la liste des conditions soit défini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleIfSourceIsSpecifiedAndIsEitherS3OrEC2OrBoth",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "events:source": [ "aws.ec2", "aws.s3" ]
                },
                "Null": {
                    "events:source": "false"
                }
            }
        }
    ]
}
```

Le tableau suivant présente des exemples de modèles d'événements qui sont autorisés ou refusés par cette politique.

Modèle d'événement	Autorisé par la stratégie
`{ "source": [ "aws.ec2" ] }`	Oui
`{ "source": [ "aws.ec2", "aws.s3" ]`	Oui

Modèle d'événement	Autorisé par la stratégie
`}`	
`{ "source": [ "aws.ec2", "aws.autoscaling" ] }`	Non
`{ "detail-type": [ "EC2 Instance State-change Notification" ] }`	Non

Exemple : limitation de l'accès de **`PutRule`** par **`detail.service`**

Vous pouvez restreindre un utilisateur ou un rôle IAM à la simple création de règles pour les événements dont le champ `events:details.service` contient une certaine valeur. La valeur de `events:details.service` n'est pas nécessairement le nom d'un service AWS.

Cette condition de politique est utile lorsque vous utilisez des événements issus d'AWS Health et qui ont un rapport avec la sécurité ou un abus. En utilisant cette condition de stratégie, vous pouvez limiter l'accès à ces alertes sensibles aux utilisateurs qui ont besoin de les voir.

Par exemple, la stratégie suivante autorise la création de règles uniquement pour les événements lorsque la valeur de `events:details.service` est `ABUSE`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleEventsWithDetailServiceEC2",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail.service": "ABUSE"
```

```
                }
            }
        }
    ]
}
```

Exemple : limitation de l'accès de **PutRule** par **detail.eventTypeCode**

Vous pouvez restreindre un utilisateur ou un rôle IAM à la simple création de règles pour les événements dont le champ `events:details.eventTypeCode` contient une certaine valeur. Cette condition de politique est utile lorsque vous utilisez des événements issus d'AWS Health et qui ont un rapport avec la sécurité ou un abus. En utilisant cette condition de stratégie, vous pouvez limiter l'accès à ces alertes sensibles aux utilisateurs qui ont besoin de les voir.

Par exemple, la stratégie suivante autorise la création de règles uniquement pour les événements lorsque la valeur de `events:details.eventTypeCode` est `AWS_ABUSE_DOS_REPORT`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleEventsWithDetailServiceEC2",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail.eventTypeCode": "AWS_ABUSE_DOS_REPORT"
                }
            }
        }
    ]
}
```

Exemple : vérification que seuls sont autorisés les événements AWS CloudTrail pour les appels d'API émanant d'un certain **PrincipalId**

Tous les événements AWS CloudTrail indiquent le PrincipalId de l'utilisateur qui a effectué l'appel d'API dans le chemin `detail.userIdentity.principalId` d'un événement. En utilisant la clé de condition `events:detail.userIdentity.principalId`, vous pouvez limiter l'accès des utilisateurs ou des rôles IAM aux seuls événements CloudTrail provenant d'un compte spécifique.

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutRuleOnlyForCloudTrailEventsWhereUserIsASpecificIAMUser",
            "Effect": "Allow",
            "Action": "events:PutRule",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "events:detail-type": [ "AWS API Call via CloudTrail" ],
                    "events:detail.userIdentity.principalId":
 [ "AIDAJ45Q7YFFAREXAMPLE" ]
                }
            }
        }
    ]
}
```

Le tableau suivant présente des exemples de modèles d'événements qui sont autorisés ou refusés par cette politique.

Modèle d'événement	Autorisé par la politique
`{ "detail-type": [ "AWS API Call via CloudTrail" ] }`	Non
`{ "detail-type": [ "AWS API Call via CloudTrail" ], "detail.userIdentity.principalId": [ "AIDAJ45Q7YFFAREXAMPLE" ] }`	Oui
`{ "detail-type": [ "AWS API Call via CloudTrail" ],`	Non

Modèle d'événement	Autorisé par la politique

<pre>"detail.userIdentity.principalId": ["AROAIDPPEZS35WEXAMPLE:AssumedRoleSessionName"]
}</pre>	

Exemple : limitation de l'accès aux cibles

Si un utilisateur ou un rôle IAM dispose de l'autorisation `events:PutTargets`, il peut ajouter aux règles qu'il est autorisé à accéder n'importe quelle cible sous le même compte. Avec la politique suivante, les utilisateurs sont limités à l'ajout de cibles à une seule règle spécifique : `MyRule` sous le compte `123456789012`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRule",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule"
        }
    ]
}
```

Pour limiter les cibles pouvant être ajoutées à la règle, utilisez la clé de condition `events:TargetArn`. Vous pouvez limiter les cibles aux seules fonctions Lambda, comme dans l'exemple suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowPutTargetsOnASpecificRuleAndOnlyLambdaFunctions",
            "Effect": "Allow",
            "Action": "events:PutTargets",
            "Resource": "arn:aws:events:us-east-1:123456789012:rule/MyRule",
            "Condition": {
```

```
                "ArnLike": {
                    "events:TargetArn": "arn:aws:lambda:*:*:function:*"
                }
            }
        }
    ]
}
```

Utilisation des rôles liés aux services pour EventBridge

Amazon EventBridge utilise des rôles AWS Identity and Access Management (IAM) liés à un service. Un rôle lié à un service est un type unique de rôle IAM lié directement à EventBridge. Les rôles liés à un service sont prédéfinis par EventBridge et comprennent toutes les autorisations nécessaires au service pour appeler d'autres services AWS en votre nom.

Rubriques

- Utilisation de rôles pour créer des secrets pour les destinations d'API
- Utilisation des rôles pour la découverte de schémas

Utilisation de rôles pour créer des secrets pour les destinations d'API

Amazon EventBridge utilise des rôles AWS Identity and Access Management (IAM) liés à un service. Un rôle lié à un service est un type unique de rôle IAM lié directement à EventBridge. Les rôles liés à un service sont prédéfinis par EventBridge et comprennent toutes les autorisations nécessaires au service pour appeler d'autres services AWS en votre nom.

Un rôle lié à un service permet d'utiliser EventBridge plus facilement, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. EventBridge définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul EventBridge peut endosser ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources EventBridge sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services prenant en charge les rôles liés à un service, consultez les AWS services opérationnels avec IAM et recherchez les services présentant la mention

Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Autorisations des rôles liés à un service pour EventBridge

EventBridgeutilise le rôle lié au service nommé AWSServiceRoleForAmazonEventBridgeApiDestinations— Permet d'accéder aux secrets du Gestionnaire de secrets créés par. EventBridge

Le rôle lié à un service AWSServiceRoleForAmazonEventBridgeApiDestinations approuve les services suivants pour endosser le rôle :

- `apidestinations.events.amazonaws.com`

La politique d'autorisations de rôle nommée AmazonEventBridgeApiDestinationsServiceRolePolicy EventBridge permet d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `create, describe, update and delete secrets; get and put secret values` sur `secrets created for all connections by EventBridge`

Vous devez configurer les autorisations de manière à permettre à vos utilisateurs, groupes ou rôles de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour EventBridge

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous créez une connexion dans leAWS Management Console, le ou l'AWSAPIAWS CLI, vous EventBridge créez le rôle lié au service pour vous.

> **Important**
>
> Ce rôle lié à un service peut apparaître dans votre compte si vous avez effectué une action dans un autre service qui utilise les fonctions prises en charge par ce rôle.
> Si vous utilisiez le EventBridge service avant le 11 février 2021, date à laquelle il a commencé à prendre en charge les rôles liés au service, vous avez EventBridge créé le AWSServiceRoleForAmazonEventBridgeApiDestinationsrôle dans votre compte. Pour en savoir plus, consultez la section Un nouveau rôle est apparu dans mon compte Compte AWS.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous créez une connexion, EventBridge crée à nouveau le rôle lié au service pour vous.

Modification d'un rôle lié à un service pour EventBridge

EventBridge ne vous permet pas de modifier le rôle lié à un service AWSServiceRoleForAmazonEventBridgeApiDestinations. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour en savoir plus, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour EventBridge

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer manuellement.

Nettoyer un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez supprimer toutes les ressources utilisées par le rôle.

> **Note**
>
> Si le service EventBridge utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer les ressources EventBridge utilisées par AWSServiceRoleForAmazonEventBridgeApiDestinations (console)

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Sous Intégrations, choisissez les destinations de l'API, puis cliquez sur l'onglet Connexions.
3. Choisissez la connexion, puis choisissez Supprimer.

Pour supprimer les ressources EventBridge utilisées par la AWSServiceRoleForAmazonEventBridgeApiDestinations (interface de ligne de commande AWS)

- Utilisez la commande suivante : `delete-connection`

Pour supprimer les ressources EventBridge utilisées par AWSServiceRoleForAmazonEventBridgeApiDestinations (API)

- Utilisez la commande suivante : `DeleteConnection`

Suppression manuelle du rôle lié au service

Utilisez la console IAM, l'AWS CLI ou l'API AWS pour supprimer le rôle lié à un service AWSServiceRoleForAmazonEventBridgeApiDestinations. Pour plus d'informations, consultez Suppression d'un rôle lié à un service dans le Guide de l'utilisateur IAM.

Régions prises en charge pour les rôles liés à un service EventBridge

EventBridge prend en charge l'utilisation des rôles liés à un service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez AWSRégions et points de terminaison.

Utilisation des rôles pour la découverte de schémas

Amazon EventBridge utilise des rôles AWS Identity and Access Management (IAM) liés à un service. Un rôle lié à un service est un type unique de rôle IAM lié directement à EventBridge. Les rôles liés à un service sont prédéfinis par EventBridge et comprennent toutes les autorisations nécessaires au service pour appeler d'autres services AWS en votre nom.

Un rôle lié à un service permet d'utiliser EventBridge plus facilement, car vous n'avez pas besoin d'ajouter manuellement les autorisations requises. EventBridge définit les autorisations de ses rôles liés à un service et, sauf définition contraire, seul EventBridge peut endosser ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisation. De plus, cette politique d'autorisation ne peut pas être attachée à une autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Vos ressources EventBridge sont ainsi protégées, car vous ne pouvez pas involontairement supprimer l'autorisation d'accéder aux ressources.

Pour plus d'informations sur les autres services prenant en charge les rôles liés à un service, consultez les AWS services opérationnels avec IAM et recherchez les services présentant la mention

Yes (Oui) dans la colonne Service-linked roles (Rôles liés à un service). Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Autorisations des rôles liés à un service pour EventBridge

EventBridgeutilise le rôle lié au service nommé AWSServiceRoleForSchemas— Accorde des autorisations aux règles gérées créées par Amazon EventBridge des schémas.

Le rôle lié à un service AWSServiceRoleForSchemas approuve les services suivants pour endosser le rôle :

- `schemas.amazonaws.com`

La politique d'autorisations de rôle nommée AmazonEventBridgeSchemasServiceRolePolicyEventBridgepermet d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `put, enable, disable, and delete rules; put and remove targets; list targets per rule` sur `all managed rules created by EventBridge`

Vous devez configurer les autorisations de manière à permettre à vos utilisateurs, groupes ou rôles de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour EventBridge

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous effectuez une découverte de schéma dans l'API AWS Management ConsoleAWS CLI, le ou l'AWSAPI, vous EventBridge créez le rôle lié au service pour vous.

> **Important**
>
> Ce rôle lié à un service peut apparaître dans votre compte si vous avez effectué une action dans un autre service qui utilise les fonctions prises en charge par ce rôle. Si vous utilisiez le EventBridge service avant le 27 novembre 2019, date à laquelle il a commencé à prendre en charge les rôles liés au service, vous avez EventBridge créé le AWSServiceRoleForSchemasrôle dans votre compte. Pour en savoir plus, consultez la section Un nouveau rôle est apparu dans mon compte Compte AWS.

Si vous supprimez ce rôle lié à un service et que vous avez ensuite besoin de le recréer, vous pouvez utiliser la même procédure pour recréer le rôle dans votre compte. Lorsque vous effectuez une découverte de schéma, le rôle lié au service est à nouveau EventBridge créé pour vous.

Modification d'un rôle lié à un service pour EventBridge

EventBridge ne vous permet pas de modifier le rôle lié à un service AWSServiceRoleForSchemas. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence à ce rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour en savoir plus, consultez Modification d'un rôle lié à un service dans le guide de l'utilisateur IAM.

Suppression d'un rôle lié à un service pour EventBridge

Si vous n'avez plus besoin d'utiliser une fonction ou un service qui nécessite un rôle lié à un service, nous vous recommandons de supprimer ce rôle. De cette façon, vous n'avez aucune entité inutilisée qui n'est pas surveillée ou gérée activement. Cependant, vous devez nettoyer votre rôle lié à un service avant de pouvoir le supprimer manuellement.

Nettoyer un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez supprimer toutes les ressources utilisées par le rôle.

> **Note**
>
> Si le service EventBridge utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression peut échouer. Si cela se produit, patientez quelques minutes et réessayez.

Pour supprimer les ressources EventBridge utilisées par AWSServiceRoleForSchemas (console)

1. Ouvrez la EventBridge console Amazon à l'adresse https://console.aws.amazon.com/events/.
2. Sous Bus, choisissez Bus d'événements, puis choisissez un bus d'événements.
3. Choisissez Arrêter la découverte.

Pour supprimer les ressources EventBridge utilisées par la AWSServiceRoleForSchemas (interface de ligne de commande AWS)

- Utilisez la commande suivante : `delete-discoverer`

Pour supprimer les ressources EventBridge utilisées par AWSServiceRoleForSchemas (API)

- Utilisez la commande suivante : `DeleteDiscoverer`

Suppression manuelle du rôle lié au service

Utilisez la console IAM, l'AWS CLI ou l'API AWS pour supprimer le rôle lié à un service AWSServiceRoleForSchemas. Pour plus d’informations, consultez Suppression d’un rôle lié à un service dans le Guide de l’utilisateur IAM.

Régions prises en charge pour les rôles liés à un service EventBridge

EventBridge prend en charge l'utilisation des rôles liés à un service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez AWSRégions et points de terminaison.

Enregistrement des appels Amazon EventBridge d'API à l'aide de AWS CloudTrail

Amazon EventBridge est intégré à AWS CloudTrailun service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un Service AWS. CloudTrail capture tous les appels d'API EventBridge sous forme d'événements. Les appels capturés incluent des appels provenant de la EventBridge console et des appels de code vers les opérations de l' EventBridge API. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite EventBridge, l'adresse IP à partir de laquelle la demande a été faite, la date à laquelle elle a été faite et des informations supplémentaires.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer :

- Si la demande a été effectuée avec des informations d'identification d'utilisateur root ou d'utilisateur root.
- Si la demande a été faite au nom d'un utilisateur de l'IAM Identity Center.
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la requête a été effectuée par un autre Service AWS.

CloudTrail est actif dans votre compte Compte AWS lorsque vous créez le compte et vous avez automatiquement accès à l'historique des CloudTrail événements. L'historique des CloudTrail événements fournit un enregistrement consultable, consultable, téléchargeable et immuable des 90 derniers jours des événements de gestion enregistrés dans un. Région AWS Pour plus d'informations, consultez la section Utilisation de l'historique des CloudTrail événements dans le guide de AWS CloudTrail l'utilisateur. La consultation de CloudTrail l'historique des événements est gratuite.

Pour un enregistrement continu des événements de vos 90 Compte AWS derniers jours, créez un magasin de données sur les événements de Trail ou CloudTrailLake.

CloudTrail sentiers

Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Tous les sentiers créés à l'aide du AWS Management Console sont multirégionaux. Vous pouvez créer un parcours à région unique ou multirégionale à l'aide du. AWS CLI Il est recommandé de créer

un parcours multirégional, car vous capturez l'activité dans l'ensemble Régions AWS de votre compte. Si vous créez un parcours à région unique, vous ne pouvez voir que les événements enregistrés dans le parcours. Région AWS Pour plus d'informations sur les sentiers, consultez les sections Création d'un sentier pour votre organisation Compte AWS et Création d'un sentier pour une organisation dans le guide de AWS CloudTrail l'utilisateur.

Vous pouvez envoyer une copie de vos événements de gestion en cours dans votre compartiment Amazon S3 gratuitement CloudTrail en créant un journal. Toutefois, des frais de stockage Amazon S3 sont facturés. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification. Pour obtenir des informations sur la tarification Amazon S3, consultez Tarification Amazon S3.

CloudTrail Stockages de données sur les événements du lac

CloudTrail Lake vous permet d'exécuter des requêtes SQL sur vos événements. CloudTrail Lake convertit les événements existants au format JSON basé sur les lignes au format Apache ORC. ORC est un format de stockage en colonnes qui est optimisé pour une récupération rapide des données. Les événements sont agrégés dans des magasins de données d'événement. Ceux-ci constituent des collections immuables d'événements basées sur des critères que vous sélectionnez en appliquant des sélecteurs d'événements avancés. Les sélecteurs que vous appliquez à un magasin de données d'événement contrôlent les événements qui persistent et que vous pouvez interroger. Pour plus d'informations sur CloudTrail Lake, consultez la section Travailler avec AWS CloudTrail Lake dans le guide de AWS CloudTrail l'utilisateur.

CloudTrail Les stockages et requêtes de données sur les événements de Lake entraînent des coûts. Lorsque vous créez un magasin de données d'événement, vous choisissez l'option de tarification que vous voulez utiliser pour le magasin de données d'événement. L'option de tarification détermine le coût d'ingestion et de stockage des événements, ainsi que les périodes de conservation par défaut et maximale pour le magasin de données d'événement. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification.

EventBridge événements de données dans CloudTrail

Les événements de données fournissent des informations sur les opérations de ressources effectuées sur ou dans une ressource (par exemple, lecture ou écriture de données dans un objet Amazon S3). Ils sont également connus sous le nom opérations de plans de données. Les événements de données sont souvent des activités dont le volume est élevé. Par défaut, CloudTrail

n'enregistre pas les événements liés aux données. L'historique des CloudTrail événements n'enregistre pas les événements liés aux données.

Des frais supplémentaires s'appliquent pour les événements de données. Pour plus d'informations sur la CloudTrail tarification, consultez la section AWS CloudTrail Tarification.

Vous pouvez enregistrer les événements de données pour les types de EventBridge ressources à l'aide de la CloudTrail console ou AWS CLI des opérations de CloudTrail l'API. Pour plus d'informations sur la façon de consigner les événements liés aux données, consultez les sections Enregistrement des événements liés aux données avec le AWS Management Console et Enregistrement des événements liés aux données avec le AWS Command Line Interface dans le Guide de AWS CloudTrail l'utilisateur.

Le tableau suivant répertorie les types de EventBridge ressources pour lesquels vous pouvez enregistrer des événements de données. La colonne Type d'événement de données (console) indique la valeur à choisir dans la liste des types d'événements de données de la CloudTrail console. La colonne de valeur resources.type indique la **`resources.type`** valeur que vous devez spécifier lors de la configuration de sélecteurs d'événements avancés à l'aide des API or. AWS CLI CloudTrail La CloudTrail colonne Data APIs logged to indique les appels d'API enregistrés CloudTrail pour le type de ressource.

Type d'événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
Bus événementiel	`AWS::Events::Event Bus`	• DescribeEventBus
Règle du bus événementiel	`AWS::Events::Rule`	• DeleteRule • DescribeRule • DisableRule • EnableRule • ListRuleNamesByTarget • ListRules • ListTargetsByRule • PutRule • PutTargets • RemoveTargets

Type d'événement de données (console)	valeur resources.type	API de données connectées à CloudTrail
		• TestEventPattern
Tuyau	`AWS::Pipes::Pipe`	• CreatePipe • DeletePipe • DescribePipe • ListPipes • StartPipe • StopPipe • UpdatePipe

Vous pouvez configurer des sélecteurs d'événements avancés pour filtrer les `eventName`, `readOnly`, et `resources.ARN` des champs pour enregistrer uniquement les événements importants pour vous. Pour plus d'informations sur ces champs, consultez AdvancedFieldSelectorla référence de l'AWS CloudTrail API.

EventBridge événements de gestion dans CloudTrail

Les événements de gestion fournissent des informations sur les opérations de gestion effectuées sur les ressources de votre Compte AWS. Ils sont également connus sous le nom opérations de plan de contrôle. Par défaut, CloudTrail enregistre les événements de gestion.

Amazon EventBridge enregistre toutes les opérations EventBridge du plan de contrôle en tant qu'événements de gestion. Pour obtenir la liste des opérations du plan de Amazon EventBridge contrôle auxquelles EventBridge se connecte CloudTrail, consultez la référence de l'Amazon EventBridge API.

EventBridge exemples d'événements

Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'opération d'API demandée, la date et l'heure de l'opération, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics. Les événements n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre un CloudTrail événement illustrant l'`PutRule`opération.

```
{
  "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"
}
```


Pour plus d'informations sur le contenu des CloudTrail enregistrements, voir le contenu des CloudTrail enregistrements dans le Guide de AWS CloudTrail l'utilisateur.

CloudTrail entrées de journal pour les actions entreprises par EventBridge Pipes

EventBridge Pipes assume le rôle IAM fourni lorsqu'il lit des événements provenant de sources, invoque des enrichissements ou invoque des cibles. Pour les CloudTrail entrées relatives aux actions effectuées dans votre compte sur tous les enrichissements, les cibles et les sources Amazon SQS, Kinesis et DynamoDB, les champs et incluront. `sourceIPAddress` `invokedBy` `pipes.amazonaws.com`

Exemple d'entrée de CloudTrail journal pour tous les enrichissements, les cibles et les sources Amazon SQS, Kinesis et DynamoDB


```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "...",
    "arn": "arn:aws:sts::111222333444:assumed-role/...",
    "accountId": "111222333444",
    "accessKeyId": "...",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "...",
        "arn": "...",
        "accountId": "111222333444",
        "userName": "userName"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-09-22T21:41:15Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "pipes.amazonaws.com"
  },
  "eventTime": ",,,",
  "eventName": "...",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "pipes.amazonaws.com",
  "userAgent": "pipes.amazonaws.com",
  "requestParameters": {
```

```
    ...
  },
  "responseElements": null,
  "requestID": "...",
  "eventID": "...",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "...",
  "eventCategory": "Management"
}
```

Pour toutes les autres sources, le `sourceIPAddress` champ des entrées du CloudTrail journal aura une adresse IP dynamique et ne doit pas être utilisé pour une intégration ou une catégorisation d'événements. De plus, ces entrées ne comportent pas le champ `invokedBy`.

Exemple d'entrée de CloudTrail journal pour toutes les autres sources

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    ...
  },
  "eventTime": ",,,",
  "eventName": "...",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Python-httplib2/0.8 (gzip)",
}
```

Validation de conformité dans Amazon EventBridge

Les auditeurs tiers tels que SOC, PCI, FedRAMP et HIPAA évaluent la sécurité et la conformité des services AWS dans le cadre de plusieurs programmes de conformité AWS.

Pour obtenir une liste des services AWS relevant de programmes de conformité spécifiques, consultez Services AWS relevant de programme de conformité. Pour obtenir des renseignements généraux, consultez Programmes de conformité AWS .

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour en savoir plus, veuillez consulter Téléchargement de rapports dans Artifact AWS.

Dans le cadre de l'utilisation d'EventBridge, votre responsabilité en matière de conformité est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et la législation et la réglementation applicables. AWS propose les ressources suivantes en matière de conformité :

- Guides de démarrage rapide pour la sécurité et la conformité : considérations architecturales et procédures de déploiement d'environnements de référence axés sur la sécurité et la conformité sur AWS.
- Conception d'une architecture pour la sécurité et la conformité en vertu de la loi HIPAA (livre blanc) : explique comment les entreprises peuvent utiliser AWS pour créer des applications conformes à la loi HIPAA.
- Ressources de conformité AWS : ensemble de manuels et de guides.
- Évaluation des ressources à l'aide de règles dans le Guide du développeur AWS Config : explique comment AWS Config évalue la conformité de vos configurations de ressources par rapport aux pratiques internes, aux directives sectorielles et aux réglementations.
- AWS Security Hub – Vue complète de l'état de votre sécurité au sein d'AWS, qui vous permet de vérifier votre conformité aux normes du secteur et aux bonnes pratiques de sécurité.

Résilience d'Amazon EventBridge

L'infrastructure mondiale AWS s'articule autours de régions et de zones de disponibilité AWS. AWS Les Régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à haut débit et hautement redondant. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour en savoir plus sur les régions AWS et zones de disponibilité , consultez Infrastructure mondiale AWS.

Sécurité d'infrastructure dans Amazon EventBridge

En tant que service géré, Amazon EventBridge est protégé par la sécurité du réseau mondial AWS. Pour plus d'informations sur les services de sécurité AWS et la manière dont AWS protège l'infrastructure, consultez la section Sécurité du cloud AWS. Pour concevoir votre environnement AWS en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le Security Pillar AWS Well-Architected Framework (Pilier de sécurité de l'infrastructure Well-Architected Framework).

Vous utilisez des appels d'API publiés par AWS pour accéder à EventBridge via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et nous recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Vous pouvez appeler ces opérations d'API de n'importe quel emplacement sur le réseau, et vous pouvez utiliser des stratégies d'accès basées sur les ressources dans EventBridge, qui peuvent inclure des restrictions en fonction de l'adresse IP source. Vous pouvez également utiliser des politiques EventBridge pour contrôler l'accès à partir de points de terminaison Amazon Virtual Private Cloud (Amazon VPC) ou de VPC spécifiques. Concrètement, l'accès réseau à une ressource EventBridge donnée s'en trouve limité au VPC en question au sein du réseau AWS.

Analyse des configurations et des vulnérabilités dans Amazon EventBridge

La configuration et les contrôles informatiques sont une responsabilité partagée entre AWS et vous, notre client. Pour de plus amples informations, veuillez consulter Modèle de responsabilité partagée AWS.

Surveillance d'Amazon EventBridge

EventBridge envoie des métriques à Amazon CloudWatch toutes les minutes, qu'il s'agisse du nombre d'événements correspondants ou du nombre de fois qu'une cible est invoquée par une règle.

La vidéo suivante passe en revue le suivi et l'audit des EventBridge comportements par le biais de CloudWatch : Surveillance et audit des événements

Rubriques

- EventBridge métriques
- Dimensions pour les EventBridge métriques

EventBridge métriques

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

Pour les métriques qui utilisent le nombre comme unité, les statistiques les plus utiles SampleCount ont tendance à être la somme et la somme.

Les métriques qui spécifient uniquement la `RuleName` dimension font référence au bus d'événements par défaut. Les métriques qui spécifient à la fois les `RuleName` dimensions `EventBusName` et font référence à un bus d'événements personnalisé.

Métrique	Description	Dimensions	Unités
`DeadLetterInvocations`	Nombre de fois où la cible d'une règle n'est pas invoquée en réponse à un événement. Cela inclut les invocations qui aboutiraient à l'exécution de la même règle à nouveau, entraînant une boucle infinie.	RuleName	Nombre
`Events`	Le nombre d'événements partenaires ingérés par EventBridge.	EventSourceName	Nombre
`FailedInvocations`	Nombre d'invocations qui ont échoué en permanence. Cela n'inclut pas les invocatio	RuleName	Nombre

Métrique	Description	Dimensions	Unités
	ns qui sont retentées ou les invocations qui ont abouti après une nouvelle tentative. Cela n'inclut pas non plus les invocations qui ont échoué et qui sont comptabilisées dans `DeadLetterInvocations` . **Note** EventBridge envoie uniquement cette métrique à CloudWatch si elle n'est pas nulle.		
`Invocations`	Nombre de fois où une cible est invoquée par une règle en réponse à un événement. Cela inclut les invocations ayant réussi et échoué, mais cela n'inclut pas les tentatives limitées ou renouvelées jusqu'à ce qu'elles échouent en permanence. Cela n'inclut pas `DeadLetterInvocations` . **Note** EventBridge envoie uniquement cette métrique à CloudWatch si elle n'est pas nulle.	Aucune, RuleName	Nombre
`InvocationAttempts`	Nombre de EventBridge tentatives d'invocation d'une cible.	Aucun	Nombre

Métrique	Description	Dimensions	Unités
`InvocationsCreated`	Nombre total d'invocations créées en réponse à chaque événement. Cette métrique est souvent utilisée pour surveiller l'utilisation de la limite d'invocations dans le quota de service de transactions par secondeEventBridge .	Aucun	Nombre
`InvocationsFailedToBeSentToDlq`	Nombre d'invocations qui n'ont pas pu être déplacées vers une file d'attente de lettres mortes. Les erreurs liées à une file d'attente de lettres mortes se produisent en raison d'erreurs d'autorisations, de ressources indisponibles ou de limites de taille. **Note** EventBridge envoie uniquement cette métrique à CloudWatch si elle n'est pas nulle.	RuleName	Nombre
`IngestionToInvocationCompleteLatency`	Temps écoulé entre l'ingestion de l'événement et la fin de la première tentative d'invocation réussie.	EventBusName, Aucun, RuleName	Millisecondes
`IngestionToInvocationStartLatency`	Le temps de traitement des événements, mesuré entre le moment où un événement est ingéré et EventBridge la première invocation d'une cible.	EventBusName, Aucun, RuleName	Millisecondes

Métrique	Description	Dimensions	Unités
`InvocationsSentToDlq`	Nombre d'invocations qui sont déplacées vers une file d'attente de lettres mortes. **Note** EventBridge envoie uniquement cette métrique à CloudWatch si elle n'est pas nulle.	RuleName	Nombre
`MatchedEvents`	Si EventBusName ou EventSourceName est spécifié, le nombre d'événements correspondant à une règle. Si elle RuleName est spécifiée, le nombre d'événements correspondant à une règle spécifique.	EventBusName, EventSourceName, RuleName	Nombre
`RetryInvocationAttempts`	Nombre de fois où l'invocation de la cible a été retentée. **Note** EventBridge envoie uniquement cette métrique à CloudWatch si elle n'est pas nulle.	Aucun	Nombre
`SuccessfulInvocationAttempts`	Nombre de fois où la cible a été invoquée avec succès.	Aucun	Nombre

Métrique	Description	Dimensions	Unités
`ThrottledRules`	Nombre de fois où l'exécution de la règle a été limitée. Les invocations relatives à ces règles peuvent être retardées. Pour plus d'informations, consultez Limite d'invocations dans les transactions par seconde dans ???.	EventBusName, Aucun, RuleName	Nombre
`TriggeredRules`	Nombre de règles exécutées et correspondant à un événement. Vous ne verrez pas cette métrique CloudWatch tant qu'une règle n'est pas déclenchée.	EventBusName, Aucun, RuleName	Nombre

EventBridge PutEvents métriques

L'espace de noms `AWS/Events` inclut les métriques suivantes appartenant aux demandes d'API `PutEvents`.

Pour les métriques qui utilisent le nombre comme unité, les statistiques les plus utiles SampleCount ont tendance à être la somme et la somme.

Métrique	Description	Dimensions	Unités
`PutEventsApproximateCallCount`	Nombre approximatif de demandes `PutEvents` reçues.	Aucun	Nombre
`PutEventsApproximateFailedCount`	Nombre approximatif de demandes `PutEvents` ayant échoué.	Aucun	Nombre
`PutEventsApproxima`	Nombre approximatif de demandes `PutEvents` réussies.	Aucun	Nombre

Métrique	Description	Dimensions	Unités
teSuccessCount			
PutEventsApproximateThrottledCount	Nombre de demandes PutEvents rejetées pour cause de limitation.	Aucun	Nombre
PutEventsEntriesCount	Nombre d'entrées d'événement contenues dans une demande PutEvents.	Aucun	Nombre
PutEventsFailedEntriesCount	Nombre d'entrées d'événement contenues dans une demande PutEvents dont l'ingestion a échoué.	Aucun	Nombre
PutEventsLatency	Temps nécessaire pour chaque demande PutEvents.	Aucun	Millisecondes
PutEventsRequestSize	Taille de la demande PutEvents.	Aucun	Octets

EventBridge PutPartnerEvents métriques

L'espace de noms AWS/Events inclut les métriques suivantes appartenant aux demandes d'API PutPartnerEvents.

> **Note**
>
> EventBridge inclut uniquement les métriques relatives aux PutPartnerEventsdemandes dans les comptes partenaires SaaS qui envoient des événements. Pour plus d'informations, consultez ???.

Pour les métriques qui utilisent le nombre comme unité, les statistiques les plus utiles SampleCount ont tendance à être la somme et la somme.

Métrique	Description	Dimensions	Unités
PutPartnerEventsApproximateCallCount	Nombre approximatif de demandes PutPartnerEvents reçues.	Aucun	Nombre
PutPartnerEventsApproximateFailedCount	Nombre approximatif de demandes PutPartnerEvents ayant échoué.	Aucun	Nombre
PutPartnerEventsApproximateThrottledCount	Nombre de demandes PutPartnerEvents rejetées pour cause de limitation.	Aucun	Nombre
PutPartnerEventsApproximateSuccessCount	Nombre approximatif de demandes PutPartnerEvents réussies.	Aucun	Nombre
PutPartnerEventsEntriesCount	Nombre d'entrées d'événement contenues dans une demande PutPartnerEvents.	Aucun	Nombre
PutPartnerEventsFailedEntriesCount	Nombre d'entrées d'événement contenues dans une demande PutPartnerEvents dont l'ingestion a échoué.	Aucun	Nombre

Métrique	Description	Dimensions	Unités
`PutPartnerEventsLatency`	Temps nécessaire pour chaque demande `PutPartnerEvents` .	Aucun	Millisecondes

Dimensions pour les EventBridge métriques

EventBridge les métriques ont des dimensions, ou des attributs triables, qui sont répertoriés ci-dessous.

Dimension	Description
`EventBusName`	Filtre les métriques disponibles par nom de bus d'événements.
`EventSourceName`	Filtre les métriques disponibles par nom de source d'événement partenaire.
`RuleName`	Filtre les métriques disponibles par nom de règle.

Résolution des problèmes liés à Amazon EventBridge

Vous pouvez suivre les étapes décrites dans cette section pour résoudre les problèmes d'Amazon EventBridge.

Rubriques

Ma règle s’est exécutée, mais ma fonction Lambda n’a pas été invoquée

L’une des raisons pour lesquelles votre fonction Lambda peut ne pas s’exécuter est que vous ne disposez pas des autorisations appropriées.

Pour vérifier vos autorisations pour votre fonction Lambda

1. À l'aide de AWS CLI, exécutez la commande suivante avec votre fonction et votre AWS région :

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

   Le résultat suivant doit s'afficher.

   ```
   {
       "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\"}"
   }
   ```

2. Si le message d'erreur suivant s'affiche.

   ```
   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
   ```

3. Si la sortie contient un champ `SourceAccount`, vous devez le supprimer. Un `SourceAccount` paramètre EventBridge empêche d'invoquer la fonction.

> **Note**
>
> Si la politique est incorrecte, vous pouvez modifier la règle dans la EventBridge console en la supprimant puis en la rajoutant à la règle. La EventBridge console définit ensuite les autorisations correctes sur la cible.
> Si vous utilisez une version ou un alias Lambda spécifique, ajoutez le paramètre `--qualifier` dans les commandes `aws lambda get-policy` et `aws lambda add-permission`, comme illustré dans 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
> --qualifier alias or version
> ```

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

Lorsque vous modifiez une règle ou ses cibles, les événements entrants peuvent ne pas commencer ou arrêter immédiatement 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 laps de temps `TriggeredRulesInvocations`, vérifiez les CloudWatch statistiques et `FailedInvocations` votre règle. Pour plus d'informations sur ces statistiques, consultez Monitoring Amazon EventBridge.

Si la règle est destinée à correspondre à un événement provenant d'un AWS service, effectuez l'une des opérations suivantes :

- Utilisez l'action `TestEventPattern` pour vérifier si le modèle d'événement de votre règle correspond à un événement de test. Pour plus d'informations, consultez TestEventPatternle Amazon EventBridge API Reference.
- Utilisez le bac à sable de la EventBridge console.

Ma règle ne s’est pas exécutée à l’heure que j’avais spécifiée dans `ScheduleExpression`

Assurez-vous que vous avez défini le planning pour la règle dans le fuseau horaire UTC+0. Si le paramètre `ScheduleExpression` est correct, suivez les étapes indiquées dans Je viens de créer ou de modifier une règle, mais elle ne correspond pas à un événement de test.

Ma règle ne s’est pas exécutée à l’heure prévue

EventBridge exécute les règles dans la minute qui suit l'heure de début que vous avez définie. Le compte à rebours pour l'exécution commence dès que la règle est créée.

> **Note**
>
> Les règles planifiées ont le type de livraison `guaranteed`, ce qui signifie que les événements seront déclenchés au moins une fois à chaque heure prévue.

Vous pouvez utiliser une expression cron pour invoquer des cibles à une heure précise. Pour créer une règle qui s’exécute toutes les quatre heures à la 0ème minute, effectuez l’une des opérations suivantes :

- Dans la EventBridge console, vous utilisez l'expression `0 0/4 * * ? *` cron.
- En utilisant le AWS CLI, vous utilisez l'expression`cron(0 0/4 * * ? *)`.

Par exemple, pour créer une règle nommée `TestRule` qui s'exécute toutes les 4 heures à l'aide de AWS CLI, vous devez utiliser la commande suivante.

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

Pour exécuter une règle toutes les cinq minutes, utilisez l’expression cron suivante.

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

La résolution maximale pour une EventBridge règle qui utilise une expression cron est d'une minute. Votre règle planifiée s’exécute au cours de cette minute, mais pas précisément à la seconde exacte.

Étant donné que EventBridge les services cibles sont distribués, il peut y avoir un délai de plusieurs secondes entre le moment où la règle planifiée s'exécute et le moment où le service cible exécute l'action sur la ressource cible.

Ma règle correspond aux appels d'API de service AWS globaux mais elle n'a pas été exécutée

AWS les services internationaux, tels que IAM et Amazon Route 53, ne sont disponibles que dans la région de l'est des États-Unis (Virginie du Nord). Les événements liés aux appels d' AWS API provenant de services internationaux ne sont donc disponibles que dans cette région. Pour plus d'informations, consultez Événements organisés par AWS les services.

Le rôle IAM associé à ma règle est ignoré lors de l'exécution de la règle

EventBridge utilise uniquement les rôles IAM pour les règles qui envoient des événements aux flux Kinesis. Pour les règles qui invoquent des fonctions Lambda ou des rubriques Amazon SNS, vous devez fournir des autorisations basées sur une ressource.

Assurez-vous que vos AWS STS points de terminaison régionaux sont activés, afin qu' EventBridge ils puissent les utiliser lorsqu'ils assument le rôle IAM que vous avez fourni. Pour plus d'informations, consultez la section Activation et désactivation AWS STS dans une AWS région dans le guide de l'utilisateur IAM.

Ma règle a un modèle d'événement censé correspondre à une ressource, mais aucun événement ne correspond

La plupart des services AWS traitent deux points (:)) ou une barre oblique (/) comme le même caractère dans Amazon Resource Names (ARN)., mais EventBridge utilisent une correspondance exacte dans les modèles d'événements et les règles. 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 mettre en correspondance.

Certains événements, tels que les événements AWS d'appel d'API provenant de CloudTrail, n'ont aucun élément dans le champ Ressources.

La livraison de mon événement à la cible a été retardée

EventBridge essaie de transmettre un événement à une cible pendant 24 heures au maximum, sauf dans les scénarios où votre ressource cible est limitée. La première tentative a lieu dès que l'événement arrive dans le flux d'événements. Si le service cible rencontre des problèmes, EventBridge replanifie automatiquement une autre livraison. Si 24 heures se sont écoulées depuis l'arrivée de l'événement, EventBridge arrête d'essayer de livrer l'événement et publie la `FailedInvocations` métrique dans CloudWatch. Nous vous recommandons de configurer une DLQ pour stocker les événements qui n'ont pas pu être livrés avec succès à une cible. Pour plus d'informations, consultez Utilisation de files d'attente de lettres mortes pour traiter les événements non livrés.

Certains événements ne sont pas livrés à ma cible

Si la cible d'une EventBridge règle est limitée pendant une période prolongée, il est EventBridge possible que vous ne réessayiez pas la livraison. Par exemple, si la cible n'est pas configurée pour gérer le trafic d'événements entrant et que le service cible limite les demandes effectuées en votre nom, il est EventBridge possible que la livraison ne EventBridge soit pas relancée.

Ma règle s'est exécutée plusieurs fois en réponse à un événement

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

Prévention des boucles infinies

Dans EventBridge, il est possible de créer une règle qui conduit à des boucles infinies, où la règle s'exécute de manière répétée. Si vous disposez d'une règle qui provoque une boucle infinie, réécrivez-la de sorte que les actions entreprises par la règle ne correspondent pas à la même règle.

Par exemple, une règle qui détecte que les listes ACL ont été modifiées sur un compartiment Amazon S3, puis qui exécute un logiciel pour les faire passer à un nouvel état provoque une boucle infinie. L'un des moyens de résoudre ce problème consiste à réécrire la règle afin qu'elle ne corresponde qu'aux listes ACL dans un état incorrect.

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.

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

Si votre file d’attente Amazon SQS est chiffrée, vous devez créer une clé KMS gérée par le client et inclure la section d’autorisation suivante dans votre stratégie de clé KMS. Pour plus d'informations, consultez la section Configuration AWS KMS des autorisations.

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

Ma règle s’exécute, mais je ne vois aucun message publié dans ma rubrique Amazon SNS

Scénario 1

Vous devez disposer d’une autorisation pour publier des messages dans votre rubrique Amazon SNS. Utilisez la commande suivante en remplaçant us-east-1 par votre région et en utilisant l'ARN de votre sujet. AWS CLI

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

Pour obtenir l’autorisation appropriée, les attributs de votre politique doivent être similaires aux attributs suivants.

```
"{\"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: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 `events.amazonaws.com` n'a pas l'autorisation `Publish` dans votre politique, commencez par copier la politique actuelle, puis ajoutez l'instruction suivante à la liste des instructions.

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

Définissez ensuite les attributs du sujet à l' AWS CLI aide de la commande suivante.

```
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 EventBridge console en la supprimant puis en la rajoutant à la règle. EventBridge définit les autorisations correctes sur la cible.

Scénario 2

Si votre rubrique SNS est chiffrée, vous devez inclure la section suivante dans votre stratégie de clé KMS.

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

Mon sujet Amazon SNS dispose toujours d'autorisations EventBridge même après avoir supprimé la règle associée au sujet Amazon SNS

Lorsque vous créez une règle avec Amazon SNS comme cible, vous ajoutez l'autorisation d' EventBridge accéder à votre rubrique Amazon SNS en votre nom. Si vous supprimez la règle peu après l'avoir créée, il est EventBridge possible que l'autorisation ne soit pas supprimée de votre rubrique Amazon SNS. Si cela se produit, vous pouvez supprimer l'autorisation de la rubrique à l'aide de la commande `aws sns set-topic-attributes`. Pour en savoir plus sur les autorisations basées sur une ressource pour l'envoi d'événements, consultez Utilisation de politiques basées sur les ressources pour Amazon EventBridge.

Quelles clés de condition IAM puis-je utiliser ? EventBridge

EventBridge prend en charge AWS les clés de condition générales (voir les clés contextuelles IAM et de AWS STS condition dans le guide de l'utilisateur IAM), ainsi que les clés répertoriées sur. Utilisation de conditions de politique IAM pour un contrôle d'accès précis

Comment savoir si les EventBridge règles ne sont pas respectées ?

Vous pouvez utiliser l'alarme suivante pour vous avertir lorsque vos EventBridge règles ne sont pas respectées.

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

1. Ouvrez la CloudWatch console à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Sélectionnez Create Alarm (Créer une alerte). Dans le volet CloudWatch Metrics by Category, sélectionnez Events Metrics.
3. Dans la liste des mesures, 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 d'alarme, dans Nom, tapez un nom unique pour l'alarme, par exemple myFailedRules. Pour Description, entrez une description de l’alarme, par exemple : Les règles ne livrent 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 alerte), choisissez State is ALARM (L'état est alerte).
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. Tapez 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 Create Alarm (Créer une alerte).

Quotas Amazon EventBridge

Il existe des quotas pour la plupart des aspects d'EventBridge.

Rubriques

> **Note**
>
> Pour obtenir la liste des quotas pour le planificateur EventBridge, consultez Quotas pour le planificateur EventBridge dans le Guide de l'utilisateur du planificateur EventBridge.

Quotas EventBridge

EventBridge présente les quotas suivants.

La console Service Quotas fournit des informations sur les quotas EventBridge. En plus de visualiser les quotas par défaut, vous pouvez utiliser la console Quotas de service pour demander des augmentations de quotas pour les quotas ajustables.

Nom	Par défaut	Ajusta	Description
Destinations d'API	Chaque région prise en charge : 3 000	Oui	Nombre maximal de destinations d'API par compte et par région.
Connexions	Chaque région prise en charge : 3 000	Oui	Nombre maximal de connexions par compte et par région.

Nom	Par défaut	Ajusta	Description
Limite de CreateEndpoint dans les transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Nombre maximal de demandes par seconde pour l'API CreateEndpoint. Les autres demandes sont bloquées.
Limite de DeleteEndpoint dans les transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Nombre maximal de demandes par seconde pour l'API DeleteEndpoint. Les autres demandes sont bloquées.
Points de terminaison	Chaque Région prise en charge : 100	Oui	Nombre maximal de points de terminaison par compte et par région.
Taille de la politique du bus d'événements	Chaque région prise en charge : 10 240	Oui	Taille maximale de la politique, en caractères. La taille de cette stratégie augmente chaque fois que vous accordez l'accès à un autre compte. Vous pouvez voir votre politique actuelle et sa taille à l'aide de l'API DescribeEventBus.
Bus d'événements	Chaque Région prise en charge : 100	Oui	Nombre maximal de bus d'événements par compte.
Taille du modèle d'événement	Chaque région prise en charge : 2 048	Oui	Taille maximale d'un modèle d'événement, en caractères.

Nom	Par défaut	Ajusta	Description
Limite d'invocations dans les transactions par seconde	us-east-1 : 18 750 par seconde us-east-2 : 4 500 par seconde us-west-1 : 2 250 par seconde us-west-2 : 18 750 par seconde af-south-1 : 750 par seconde ap-northeast-1 : 2 250 par seconde ap-northeast-3 : 750 par seconde ap-southeast-1 : 2 250 par seconde ap-southeast-2 : 2 250 par seconde ap-southeast-3 : 750 par seconde eu-central-1 : 4 500 par seconde eu-south-1 : 750 par seconde	Oui	Une invocation est un événement correspondant à une règle et qui est envoyé vers les cibles de la règle. Une fois la limite atteinte, les invocations sont limitées, c'est-à-dire qu'elles se produisent encore, mais sont retardées.

Nom	Par défaut	Ajusta	Description
	eu-west-1 : 18 750 par seconde eu-west-2 : 2 250 par seconde Chacune des autres régions prises en charge : 1 100 par seconde		
Nombre de règles	af-south-1 : 100 eu-south-1 : 100 Chacune des autres régions prises en charge : 300	Oui	Nombre maximal de règles qu'un compte peut avoir par bus d'événements

Nom	Par défaut	Ajusta	Description
Limite de PutEvents dans les transactions par seconde	us-east-1 : 10 000 par seconde us-east-2 : 2 400 par seconde us-west-1 : 1 200 par seconde us-west-2 : 10 000 par seconde af-south-1 : 400 par seconde ap-northeast-1 : 1 200 par seconde ap-northeast-3 : 400 par seconde ap-southeast-1 : 1 200 par seconde ap-southeast-2 : 1 200 par seconde ap-southeast-3 : 400 par seconde eu-central-1 : 2 400 par seconde eu-south-1 : 400 par seconde	Oui	Nombre maximal de demandes par seconde pour l'API PutEvents. Les autres demandes sont bloquées.

Nom	Par défaut	Ajusta	Description
	eu-west-1 : 10 000 par seconde eu-west-2 : 1 200 par seconde Chacune des autres régions prises en charge : 600 par seconde		
Taux d'invocations par destination d'API	Chaque région prise en charge : 300 par seconde	Oui	Nombre maximal d'invocations par seconde à envoyer à chaque point de terminaison de destination d'API par compte et par région. Une fois le quota atteint, les futures invocations vers ce point de terminaison d'API sont limitées. Les invocations se produiront toujours, mais elles seront retardées.
Cibles par règle	Chaque région prise en charge : 5	Non	Nombre maximal de cibles qui peuvent être associées à une règle

Nom	Par défaut	Ajusta	Description
Limite dans les transactions par seconde	Chaque région prise en charge : 50 par seconde	Oui	Nombre maximal de demandes par seconde pour toutes les opérations d'API EventBridge à l'exception de PutEvents. Les autres demandes sont limitées
Limite d'UpdateEndpoint dans les transactions par seconde	Chaque région prise en charge : 5 par seconde	Non	Nombre maximal de demandes par seconde pour l'API UpdateEndpoint. Les autres demandes sont bloquées.

En outre, EventBridge présente les quotas suivants qui ne sont pas gérés via la console Service Quotas.

Nom	Par défaut	Description
Bus d'événements	Chaque Région prise en charge : 100	Nombre maximal de bus d'événements par compte.
Taille de la politique du bus d'événements	Chaque région prise en charge : 10 240	Taille maximale de la politique, en caractères. La taille de cette stratégie augmente chaque fois que vous accordez l'accès à un autre compte. Vous pouvez voir votre politique actuelle et sa taille à l'aide de l'API `DescribeEventBus` .
Taille du modèle d'événement	Chaque région prise	Taille maximale d'un modèle d'événement, en caractères.

Nom	Par défaut	Description
	en charge : 2 048	Ce quota peut être ajusté jusqu'à 4 096 caractères. Si vous devez augmenter la limite maximale, contactez l'assistance.
Règles contenant des caractères génériques	Chaque région prise en charge : 30 règles par bus d'événements	Nombre maximal de règles, par bus d'événements et par compte, pouvant contenir des filtres d'événements incluant des caractères génériques. Ce quota ne peut pas être ajusté. Pour plus d'informations sur l'utilisation de caractères génériques dans les modèles d'événements, consultez ???.
Niveaux de découverte de schéma	Chaque région prise en charge : 255 niveaux	Nombre maximal de niveaux que la découverte de schéma déduira des événements imbriqués . Tous les événements dépassant 255 niveaux sont ignorés.

Quotas PutPartnerEvents par région

Si vous devez augmenter les limites maximales, contactez l'assistance.

Régions	Transactions par seconde
• AWS GovCloud (US-West) • AWS GovCloud (US-East) • USA Est (Virginie du Nord) • USA Est (Ohio) • USA Ouest (Californie du Nord) • USA Ouest (Oregon) • Afrique (Le Cap) • Asie-Pacifique (Hong Kong) • Asia Pacific (Mumbai)	PutPartnerEvents présente une limite flexible de 1 400 demandes de débit par seconde et de 3 600 demandes en rafales par seconde, par défaut, dans toutes les régions.

Régions	Transactions par seconde
• Asie-Pacifique (Osaka) • Asia Pacific (Seoul) • Asie-Pacifique (Singapour) • Asie-Pacifique (Sydney) • Asie-Pacifique (Tokyo) • Canada (Centre) • Europe (Francfort) • Europe (Irlande) • Europe (Londres) • Europe (Milan) • Europe (Paris) • Europe (Stockholm) • Europe (Milan) • Amérique du Sud (São Paulo) • Chine (Ningxia) • Chine (Beijing)	

Quotas du registre de schémas EventBridge

Le registre de schémas EventBridge présente les quotas suivants.

La console Service Quotas fournit des informations sur les quotas EventBridge. En plus de visualiser les quotas par défaut, vous pouvez utiliser la console Quotas de service pour demander des augmentations de quotas pour les quotas ajustables.

Nom	Par défaut	Ajusta	Description
DiscoveredSchemas	Chaque région prise en charge : 200	Oui	Nombre maximal de schémas que vous pouvez créer dans la région actuelle pour un

Nom	Par défaut	Ajusta	Description
			registre de schémas découvert
Outils de découverte	Par région prise en charge : 10	Oui	Nombre maximal d'outils de découverte que vous pouvez créer dans la région actuelle.
Registres	Par région prise en charge : 10	Oui	Nombre maximal de registres que vous pouvez créer dans la région actuelle.
SchemaVersions	Chaque Région prise en charge : 100	Oui	Nombre maximal de versions par schéma que vous pouvez créer dans la région actuelle.
Schémas	Chaque Région prise en charge : 100	Oui	Nombre maximal de schémas par registre que vous pouvez créer dans la région actuelle. (À l'exception du registre des schémas découvert)

Quotas EventBridge Pipes

EventBridge Pipes présente les quotas suivants. Si vous devez augmenter les limites maximales, contactez l'assistance.

Ressource	Régions	Limite par défaut
Nombre d'exécutions de canal simultanées par compte	• AWS GovCloud (US-West) • AWS GovCloud (US, côte est)	1 000

Ressource	Régions	Limite par défaut
	• Chine (Ningxia) • Chine (Beijing) • Asie-Pacifique (Osaka) • Afrique (Le Cap) • Europe (Milan) • USA Est (Ohio) • Europe (Francfort) • USA Ouest (Californie du Nord) • Europe (Londres) • Asie-Pacifique (Sydney) • Asia Pacific (Tokyo) • Asie-Pacifique (Singapour) • Canada (Centre) • Europe (Paris) • Europe (Stockholm) • Amérique du Sud (São Paulo) • Asie-Pacifique (Séoul) • Asie-Pacifique (Mumbai) • Asie-Pacifique (Hong Kong) • Moyen-Orient (Bahreïn) • Chine (Ningxia) • Chine (Beijing) • Asie-Pacifique (Osaka) • Afrique (Le Cap) • Europe (Milan)	

Ressource	Régions	Limite par défaut
Nombre d'exécutions de canal simultanées par compte	• USA Est (Virginie du Nord) • USA Ouest (Oregon) • Europe (Irlande)	3000
Canaux par compte	Tous	1 000

EventBridge Balises Amazon

Une balise est une étiquette d'attribut personnalisée que vous attribuez ou AWS assignez à une AWS ressource. Dans EventBridge, vous pouvez attribuer des balises aux bus de règles et d'événements. Chaque ressource peut avoir un maximum de 50 balises.

Vous utilisez des balises pour identifier et organiser vos AWS ressources. De nombreux AWS services prennent en charge le balisage. Vous pouvez donc attribuer le même tag aux ressources de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à une EventBridge règle que celle que vous attribuez à une instance EC2.

Une 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.
  - La longueur maximale des clés de balise est de 128 caractères Unicode en UTF-8.
  - Pour chaque ressource, chaque clé de la balise doit être unique.
  - Les caractères autorisés sont les lettres, les espaces et les chiffres représentables en UTF-8, ainsi que les caractères spéciaux suivants : . : + = @ _ / - (tiret).
  - Le `aws:` préfixe est interdit pour les balises car il est réservé à l' AWS usage. Vous ne pouvez pas modifier ni supprimer des clés ou valeurs d'étiquette ayant ce préfixe. Les balises avec ce préfixe ne sont pas comptabilisées comme vos balises pour la limite de ressources.
- Un champ facultatif de valeur de balise, par exemple `111122223333` ou `Production`.
  - Chaque clé de balise ne peut avoir qu'une seule valeur.
  - Les valeurs de balise sont sensibles à la casse.
  - Omettre la valeur de balise équivaut à l'utilisation d'une chaîne vide.
  - 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).

> **Tip**
>
> 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, vous pouvez choisir d'utiliser `Costcenter`, `costcenter` ou `CostCenter` et appliquer ensuite la même convention à toutes les balises.

Vous pouvez utiliser la EventBridge console, l' EventBridge API ou le AWS CLI pour ajouter, modifier ou supprimer des balises. Pour plus d'informations, consultez les ressources suivantes :

- TagResourceUntagResource, et ListTagsForResourcedans le Amazon EventBridge API Reference
- tag-resource, untag-resource, et dans la référence list-tags-for-resourceAWS CLI
- Utilisation de l'éditeur de balises dans le Guide de l'utilisateur Resource Groups

Historique du document

Le tableau suivant décrit les modifications importantes apportées à chaque version du guide de EventBridge l'utilisateur Amazon, à compter de juillet 2019. Pour recevoir les notifications de mise à jour de cette documentation, abonnez-vous à un flux RSS.

Modification	Description	Date de parution
Politiques AWS gérées mises à jour.	AWS GovCloud (US) Regions uniquement `AmazonEventBridgeFullAccess` et `AmazonEventBridgeSchemasFullAccess` les politiques ne l'incluent pas`iam:CreateServiceLinkedRole`, car il n'est pas utilisé. • the section called “Mises à jour des politiques”	9 mai 2024
Générez des AWS CloudFormation modèles à partir des bus d'événements et des règles.	Vous pouvez désormais générer des AWS CloudFormation modèles à partir de vos bus et règles d' EventBridge événements Amazon existants. • Génération d'un modèle AWS CloudFormation à partir d'un bus d'événements Amazon EventBridge	18 novembre 2022
Lancement de la documentation EventBridge Pipes.	Vous pouvez désormais créer des canaux pour connecter les sources aux cibles, avec un filtrage et un enrichissement facultatifs. • Canaux	1er décembre 2022
Générez des AWS CloudFormation modèles à partir des bus d'événements et des règles.	Vous pouvez désormais générer des AWS CloudFormation modèles à partir de vos bus et règles d' EventBridge événements Amazon existants.	18 novembre 2022

Modification	Description	Date de parution
	• Génération d'un modèle AWS CloudFormation à partir d'un bus d'événements Amazon EventBridge	
La AmazonEventBridgePipesFullAccess politique a été ajoutée.	Fournit un accès complet à Amazon EventBridge Pipes. • EventBridge Politiques gérées spécifiques aux tuyaux	1er décembre 2022
La AmazonEventBridgePipesReadOnlyAccess politique a été ajoutée.	Fournit un accès en lecture seule à Amazon EventBridge Pipes. • EventBridge Politiques gérées spécifiques aux tuyaux	1er décembre 2022
La AmazonEventBridgePipesOperatorAccess politique a été ajoutée.	Fournit un accès en lecture seule et aux opérateurs (c'est-à-dire la possibilité d'arrêter et de démarrer l'exécution de Pipes) à Amazon EventBridge Pipes. • EventBridge Politiques gérées spécifiques aux tuyaux	1er décembre 2022
Mise à jour de la CloudWatchEventsFullAccess politique .	Mis à jour pour correspondre à la politique `AmazonEventBridgeFullAccess` . • AmazonEventBridgeFullAccess politique	1er décembre 2022
Mise à jour de la CloudWatchEventsReadOnlyAccess politique.	Mis à jour pour correspondre à la politique `AmazonEventBridgeReadOnlyAccess` . • AmazonEventBridgeReadOnlyAccess politique	1er décembre 2022

Modification	Description	Date de parution
Mise à jour du filtrage de contenu dans les modèles d'événements.	Vous pouvez désormais utiliser les options de filtrage `suffix`, `equals-ignore-case` et `$or` pour créer des modèles d'événements. • Filtrage du contenu dans les modèles EventBridge d'événements Amazon	14 novembre 2022
Mise à jour de la AmazonEventBridgeFullAccess politique.	Autorisations supplémentaires nécessaires à l'utilisation du registre des EventBridge schémas et du EventBridge planificateur. • AmazonEventBridgeFullAccess politique	10 novembre 2022
Mise à jour de la AmazonEventBridgeReadOnlyAccess politique.	Vous pouvez désormais consulter les informations du registre des EventBridge schémas et du EventBridge planificateur. • AmazonEventBridgeReadOnlyAccess politique	10 novembre 2022
Mise à jour du filtrage de contenu dans les modèles d'événements.	Vous pouvez désormais utiliser les options de filtrage `suffix`, `equals-ignore-case` et `$or` pour créer des modèles d'événements. • Filtrage du contenu dans les modèles EventBridge d'événements Amazon	14 novembre 2022
Mise à jour de la AmazonEventBridgeFullAccess politique.	Autorisations supplémentaires nécessaires à l'utilisation du registre des EventBridge schémas et du EventBridge planificateur. • AmazonEventBridgeFullAccess politique	10 novembre 2022

Modification	Description	Date de parution
Mise à jour de la AmazonEventBridgeReadOnlyAccess politique.	Vous pouvez désormais consulter les informations du registre des EventBridge schémas et du EventBridge planificateur. • AmazonEventBridgeReadOnlyAccess politique	10 novembre 2022
Mise à jour de la AmazonEventBridgeReadOnlyAccess politique.	Vous pouvez désormais consulter les informations relatives aux points de terminaison. • AmazonEventBridgeReadOnlyAccess politique	7 avril 2022
Ajout de la prise en charge des points de terminaison globaux.	Amazon prend EventBridge désormais en charge l'utilisation de points de terminaison mondiaux pour rendre votre application tolérante aux pannes régionales, sans frais supplémentaires. Pour en savoir plus, prenez connaissance de ce qui suit : • Rendre les applications tolérantes aux pannes régionales avec des points de terminaison globaux et une réplication d'événements • `CreateEndpoint`	7 avril 2022
Ajout de la prise en charge des archives et des relectures d'événements.	Amazon prend EventBridge désormais en charge l'utilisation d'archives pour stocker les événements et les rediffusions d'événements pour rejouer les événements depuis une archive. Pour en savoir plus, prenez connaissance de ce qui suit : • Archivage des événements Amazon EventBridge . • `CreateArchive` • `StartReplay`	5 novembre 2020

Modification	Description	Date de parution
Ajout de la prise en charge des files d’attente de lettres mortes et de la politique de nouvelle tentative pour les cibles.	Amazon prend EventBridge désormais en charge l'utilisation de files d'attente contenant des lettres mortes et la définition d'une politique de nouvelles tentatives pour les cibles. Pour en savoir plus, prenez connaissance de ce qui suit : • Utilisation de files d'attente de lettres mortes pour traiter les événements non livrés. • `PutTargets`	12 octobre 2020
Ajout de la prise en charge des schémas au format JSONSchema Draft4.	Amazon prend EventBridge désormais en charge les schémas au format JSONSchema Draft 4. Vous pouvez également désormais exporter des schémas à l'aide de l' EventBridge API. Pour en savoir plus, consultez les rubriques suivantes. • EventBridge Schémas Amazon • `Export`dans le EventBridge Schema Registry API Reference.	28 septembre 2020
Politiques basées sur les ressources pour le registre des schémas EventBridge	L'Amazon EventBridge Schema Registry prend désormais en charge les politiques basées sur les ressources. Pour plus d’informations, consultez les rubriques suivantes. • Politiques basées sur les ressources pour les schémas Amazon EventBridge • `Policy`dans la référence de l'API EventBridge Schema Registry • RegistryPolicy Type de ressource dans le guide de AWS CloudFormation l'utilisateur	30 avril 2020

Modification	Description	Date de parution
Balises pour les bus d’événements	Cette version vous permet de créer et de gérer des balises pour les bus d'événements. Vous pouvez ajouter des balises lors de la création d'un bus d'événements, et ajouter ou gérer des balises existantes en appelant l'API associée. Pour plus d’informations, consultez les rubriques suivantes. • EventBridge Balises Amazon • Politiques basées sur des balises • `TagResource` • `UntagResource` • `ListTagsForResource`	24 février 2020
Augmentation des quotas de service	Amazon EventBridge a augmenté les quotas pour les invocations et pour. `PutEvents` Les quotas varient selon les régions et peuvent être augmentés si nécessaire.	11 février 2020

Modification	Description	Date de parution
Ajout d’une nouvelle rubrique sur la transformation de l’entrée cible et ajout d’un lien vers les événements Application Auto Scaling.	Amélioration de la documentation sur le transformateur d'entrée. • Transformation des EventBridge entrées Amazon • Utilisation du transformateur d’entrée pour extraire des données à partir d'un événement et entrer ces données dans la cible • Didacticiel : Utilisation du transformateur d'entrée pour personnaliser les éléments qu'EventBridge transmet à la cible d'événement Ajout d’un lien vers les événements Application Auto Scaling. • Événements d'Application Auto Scaling et EventBridge • Événements organisés par AWS les services	20 décembre 2019
Filtrage basé sur le contenu		19 décembre 2019
Ajout de liens vers des exemples d'événement Amazon Augmented AI	Ajout d'un lien vers la rubrique Amazon Augmented AI dans le manuel Amazon SageMaker Developer Guide qui fournit des exemples d'événements pour Amazon Augmented AI. Pour plus d’informations, consultez les rubriques suivantes. • Utilisation des événements dans Amazon Augmented AI • Événements organisés par AWS les services	13 décembre 2019

Modification	Description	Date de parution
Ajout de liens vers des exemples d'événement Amazon Chime.	Ajout d'un lien vers la rubrique Amazon Chime qui fournit des exemples d'événement pour ce service. Pour plus d'informations, consultez les rubriques suivantes. • Automatiser Amazon Chime avec EventBridge • Événements organisés par AWS les services	12 décembre 2019
EventBridge Schémas Amazon	Vous pouvez désormais gérer les schémas et générer des liaisons de code pour les événements sur Amazon. EventBridge Pour plus d'informations, consultez les rubriques suivantes. • EventBridge Schémas Amazon • EventBridge Référence de l'API Schemas • EventSchemas Référence du type de ressource dans AWS CloudFormation	1 décembre 2019
AWS CloudFormation support pour Event Buses	AWS CloudFormation prend désormais en charge la EventBus ressource. Il prend également en charge le EventBusName paramètre dans les ressources EventBusPolicy et Rule. Pour plus d'informations, consultez Amazon EventBridge Resource Type Reference.	7 octobre 2019
Nouveau service	Version initiale d'Amazon EventBridge.	11 juillet 2019

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.