aws

Guide du développeur

AWS Partner Central

AWS Partner Central: Guide du développeur

Table of Contents

..... vi
Bienvenue 1
Utilisation des AWS SDK 1
Configuration et authentification 2
Lier votre Compte AWS vers Partner Central 2
Configuration de IAM 3
Authentification des appels d'API 3
Signature de vos appels avec un agent utilisateur personnalisé 5
Agents de Partner Central (MCP Server) 7
Présentation de 7
Fonctionnalités de l'agent 7
Principaux avantages 8
Exemples d’utilisation 9
Création d'opportunités 9
Clonage d'opportunités 9
Informations sur les pipelines 10
Résumé de l'opportunité 10
Génération de jeux de ventes 10
Création d'un profil client 10
Recommandation de solution 11
Recommandation de financement 11
Recommandations pour la prochaine étape 11
Progression des opportunités 12
Expérience agentic pour l'intégration des partenaires 12
Automatisation du profil des partenaires 13
Conseils de configuration pour les vendeurs 13
Conseils de conformité PRM 13
Prise en main 14
Prise en main 14
Conditions préalables 14
Étape 1 : configurer les autorisations IAM 14
Étape 2 : Connectez votre client MCP 21
Signature de vos appels avec un en-tête MCP 23
Étape 3 : vérifier votre configuration 24

Étape 4 : Exécutez vos premières tâches 25
Considérations sur la sécurité 27
Étapes suivantes 28
Référence de configuration 28
Endpoint 28
Autorisations IAM 29
Environnements de catalogue 33
Gestion de session 34
Chargement d'un fichier 34
Codes d’erreur 35
Limites de débit 36
Types d'événements de streaming (SSE) 36
Étapes suivantes 37
Référence des outils 37
Vue d'ensemble des outils 38
`Envoyer un message` 38
`Obtenir une session` 46
Gestion des erreurs 48
À propos des API 50
Utilisation des API 50
Utilisation de l'API du compte 50
Utilisation de l'API de vente 64
Utilisation de l'API Benefits 122
Utilisation de l'API Channel 137
Utilisation de l'API de mesure des recettes 140
Régions prises en charge 152
Régions pour l'API du compte 152
Régions pour l'API de vente 152
Régions pour l'API Benefits 153
Régions pour l'API Channel 153
Régions pour l'API de mesure des recettes 154
Permissions 155
Accès à l'API du compte 155
Accès à l'API de vente 158
Accès à l'API Benefits 164
Accès à l'API Channel 167

Accès à l'API de mesure des recettes 176
Quotas 185
Quotas pour l'API du compte 185
Quotas pour l'API de vente 189
Quotas pour l'API Benefits 191
Quotas pour l'API Channel 193
Quotas pour l'API de mesure des recettes 199
Logging 201
Journalisation de l'API du compte 201
Journalisation de l'API de vente 205
Journalisation de l'API Benefits 206
Journalisation de l'API du canal 209
Enregistrement de l'API de mesure des recettes 211
Notifications 214
Notifications pour AWS API du compte Partner Central 215
Notifications pour AWS API de vente centralisée pour les partenaires 227
Notifications pour AWS API Partner Central Benefits 242
Notifications pour AWS API Partner Central Channel 256
Tester dans un bac à sable 266
Accès à l'environnement sandbox 267
Informations importantes concernant l'environnement sandbox 267
Test de l'API du compte dans un bac à sable 267
Test de l'API de vente dans un bac à sable 271
Tester l'API Benefits dans un bac à sable 275
Test dans un sandbox pour l'API Channel 280
Test dans un bac à sable pour l'API de mesure des recettes 284
Exemples de code 289
Principes de base 290
Actions 291
Scénarios 357
Mise à jour de l’entité associée à une opportunité 358
Notes de mise à jour 364
Historique du document 397

La référence AWS Partner Central d'API a été restructurée. Pour plus d'informations sur les opérations d'API prises en charge, consultez la référence des AWS Partner Central API.

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.

Bienvenue sur le AWS Guide du développeur Partner Central

Ce guide du développeur décrit comment les AWS partenaires peuvent utiliser les API de service pour intégrer et gérer les activités de création, de commercialisation, de vente et de développement avec AWS.

Rubriques

- Utilisation AWS API Partner Central avec un AWS Kit SDK
- Configuration et authentification

Utilisation AWS API Partner Central avec un AWS Kit SDK

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

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

Documentation SDK	Exemples de code
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour Ruby exemples de code
AWS SDK pour Rust	AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
AWS SDK pour Swift	AWS SDK pour Swift exemples de code

> **Exemple de disponibilité**
>
> Vous n’avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Provide feedback (Fournir un commentaire) en bas de cette page.

Configuration et authentification

La configuration et l'authentification avec l'API AWS Partner Central s'effectuent en trois étapes. Voici un aperçu du processus :

1. Associez votre compte AWS Marketplace vendeur à Partner Central.
2. Configurez les autorisations à l'aide d'IAM.
3. Authentifiez vos appels d'API à l'aide de Signature Version 4 (SigV4).

Lier votre Compte AWS vers Partner Central

La connexion Compte AWSà Partner Central est une condition préalable à l'utilisation de l'API. Pour plus d'informations, consultez Lier les comptes AWS Partner Central aux comptes AWS Marketplace vendeur. Vous devez vous connecter à Partner Central avec un compte disposant des autorisations de responsable d'alliance ou d'administrateur du cloud, accéder à la **`Account Linking`** section et suivre les instructions.

Configuration de IAM

Pour utiliser l'API AWS Partner Central, vous aurez besoin d'un rôle Gestion des identités et des accès AWS(IAM) ou d'un utilisateur IAM pour commencer à passer des appels. Pour plus d'informations, voir Quand utiliser IAM ? . Pour cela, suivez les étapes de création de rôles IAM et de création d'un utilisateur IAM décrites dans vos Compte AWS guides. Vous devez créer cet IAM Role/User dans votre compte Central-linked AWS Marketplace vendeur partenaire. role/user La création d'IAM n'entraîne aucun coût.

1. Création d'un IAM Role/User

   Connectez-vous au AWS Management Console, accédez au service IAM et suivez les étapes pour créer un rôle IAM ou un utilisateur IAM.
2. Attribuer des politiques :

   Joignez des politiques gérées ou créez des politiques personnalisées selon vos besoins. Pour modifier ou étendre les autorisations, appliquez des politiques supplémentaires au rôle IAM au lieu de copier et de combiner le contenu `AWSPartnerCentralOpportunityManagement` avec d'autres autorisations. Évitez de dupliquer les politiques gérées, car cela vous empêchera d'accéder automatiquement aux nouvelles fonctionnalités au fur et à mesure de leur publication, et vous devrez mettre à jour vos politiques manuellement à l'avenir. Pour plus de détails sur les politiques d'accès, consultez la documentation sur le contrôle d'accès.

Gestion AWS Marketplace offres

Pour gérer les AWS Marketplace offres et les associer à des opportunités, les partenaires doivent autoriser le rôle IAM à accéder aux API du catalogue. Assurez-vous qu' role/user il dispose d'autorisations, telles que `aws-marketplace:ListEntities` et`aws-marketplace:SearchAgreements`.

Authentification des appels d'API

AWS L'API Partner Central utilise Signature Version 4 (SigV4) pour l'authentification. Voici comment l'implémenter :

Utilisation de AWS Kit SDK

AWS Les SDK gèrent automatiquement la signature des demandes. Entrez vos AWS informations d'identification et le SDK s'occupe du reste.

1. Pour Java, voir Fournir des informations d'identification temporaires au AWS SDK for Java.
2. Pour Python (Boto3), voir Credentials.
3. Pour JavaScript (Node.js), voir Configuration des informations d'identification dans Node.js.
4. Pour .NET, voir Credential and profile resolution.
5. Pour d'autres langages de programmation et d'autres exemples, consultez les Outils sur lesquels vous pouvez vous appuyer AWS.

Authentification sans utiliser le AWS Kit SDK

Si aucun AWS SDK n'est disponible pour le langage de programmation que vous avez choisi, l'authentification consiste à créer manuellement une demande canonique, à signer la demande et à gérer les jetons de session.AWS propose des conseils complets sur l'utilisation de la signature SigV4. Cependant, veuillez noter que l'utilisation du AWS SDK est recommandée car la signature manuelle des demandes augmente la complexité et nécessite une gestion minutieuse des jetons de sécurité.

Chaque demande pour le protocole AWSJSON1_0 DOIT être envoyée à l'URL racine (/) en utilisant la méthode HTTP « POST » en utilisant les en-têtes suivants.

En-têtes obligatoires

Chaque demande d'API pour le protocole AWSJSON1_0 DOIT être envoyée à l'URL racine (/) en utilisant la méthode HTTP « POST » en utilisant les en-têtes suivants.

En-tête	Obligatoire	Description
Content-Type	true	Cet en-tête a une valeur statique de`application/x-amz-json-1.0` .
Content-Length	true	L' Content-Length en-tête standard défini par la RFC 9110 #section -8.6.
X-Amz-Target	vrai pour les demandes	Par exemple, la valeur du fonctionnement `CreatePartner` du service `PartnerCentralAccount` est`PartnerCentralAccount.CreatePartner` .

Signature de vos appels avec un agent utilisateur personnalisé

Lorsque vous envoyez des demandes d'API àAWS Partner Central, nous vous recommandons d'inclure l'`X-Amzn-User-Agent`en-tête pour aider àAWS identifier la source de l'application cliente, à contrôler l'utilisation et à auditer les performances.AWS utilise cet en-tête pour distinguer le type d'application client effectuant l'appel et pour recueillir des informations sur le taux de réussite des différentes implémentations clientes.

En-tête d'agent utilisateur personnalisé

Nom de l'en-tête : `X-Amzn-User-Agent`

Objectif : distingue le type de client qui fait la demande d'API, en catégorisant la source de l'interaction.

Format : `{Integrator}|{Product}`

Exemple de valeur : `AWS|AWS Partner CRM Connector`

L'inclusion de cet en-tête dans chaque demande permet d'AWS analyser les modèles de demandes, de suivre les intégrations et d'améliorer l'expérience API pour les différents systèmes clients.

Utilisation d'en-têtes personnalisés dans les SDK

Pour inclure l'`X-Amzn-User-Agent`en-tête dans les appels du SDK, vous pouvez modifier le comportement des demandes du client avant de lancer l'appel d'API. Vous trouverez ci-dessous un exemple d'API de vente utilisant le AWS SDK pour Python (Boto3) :

```
import boto3

# Define service and endpoint details
service_name = "partnercentral-selling"
endpoint_url = "https://partnercentral-selling.us-east-1.api.aws"

# Create a boto3 client for Partner Central
partner_central_client = boto3.client(
service_name=service_name,
='us-east-1',
endpoint_url=endpoint_url
)
```

```
# Function to add the custom User-Agent header

def add_version_header(params, **kwargs):
params["headers"]['X-Amzn-User-Agent'] = 'AWS|AWS Partner CRM Connector'

# Register the event to modify the request before the call is made
partner_central_client.meta.events.register(
f'before-call.{service_name}.*', add_version_header
)

# Now, whenever an API call is made using this client, the custom User-Agent header
 will be included
```

Cet exemple montre comment enregistrer un événement dans le SDK Boto3 pour ajouter automatiquement l'`X-Amzn-User-Agent`en-tête à chaque demande d'API. La même approche peut être appliquée à d'autres AWS SDK en modifiant leurs mécanismes d'interception des demandes respectifs.

> **Note**
>
> Le format `X-Amzn-User-Agent` d'en-tête a été simplifié. Nous vous recommandons d'utiliser le nouveau format. Le format précédent (`CompanyName` | `ProductName` | `CRMName` | `ProductVersion`) reste pris en charge pour des raisons de rétrocompatibilité. Les intégrations existantes continueront de fonctionner sans modification.

Agents de Partner Central (MCP Server)

Le serveur MCP des agents Partner Central fournit des outils Partner Central via le protocole MCP (Model Context Protocol), permettant à vos agents et outils d'IA de découvrir et d'interagir avec la gestion des opportunités, les informations sur les clients et les programmes de financement en langage naturel.

Le serveur gère l'authentification, la gestion des sessions et l'approbation humaine pour les opérations d'écriture, en maintenant le contrôle tout en rationalisant les flux de travail.

Présentation de

Le serveur MCP des agents Partner Central est un service entièrement géré et AWS hébergé qui connecte les clients MCP-compatible IA aux agents Partner Central. Il utilise le protocole HTTPS JSON-RPC 2.0 avec authentification Sigv4 et prend en charge les Server-Sent événements (SSE) pour les réponses en streaming en temps réel.

Le serveur prend en charge les conversations à plusieurs tours, les pièces jointes pour l'analyse des documents et un flux de travail d'approbation intégré qui nécessite votre consentement explicite avant l'exécution de toute opération d'écriture.

Fonctionnalités de l'agent

- Informations sur le pipeline : obtenez des informations conversationnelles sur votre pipeline de ventes, y compris les opportunités à risque, la progression des étapes et une analyse des pertes clôturées
- Création d'opportunités : créez de nouvelles opportunités grâce à une conversation en langage naturel, en téléchargeant des notes de réunion, des propositions ou des transcriptions d'appels, ou en clonant une opportunité existante
- Résumé des opportunités : générez des résumés concis en un coup d'œil de toute transaction, couvrant l'étape, les dépenses, la date de clôture, etc.
- Création d'opportunités commerciales : élaborez des stratégies de vente personnalisées en combinant les détails des opportunités, le contexte du secteur et les recommandations de AWS solutions
- Création de profils clients — Générez des profils d'entreprise à l'aide d'informations accessibles au public concernant le secteur, le modèle commercial, la géographie et les développements récents

- Recommandation de solution : Cross-reference vos solutions enregistrées par rapport aux exigences en matière d'opportunités pour trouver la meilleure solution
- Recommandation de financement — Évaluez les opportunités par rapport aux programmes de AWS financement disponibles, estimez les montants et créez des demandes de fonds
- Recommandations pour la prochaine étape — Obtenez des plans d'action hiérarchisés fondés sur les normes de AWS covente et les conseils de progression par étapes
- Progression des opportunités : téléchargez les pièces justificatives, extrayez les données pertinentes et faites progresser les opportunités au cours des étapes du pipeline
- AI-assisted liste de produits : générez des listes de AWS Marketplace produits de haute qualité à partir de vos actifs numériques existants, évaluez la solidité de vos offres par rapport AWS Marketplace aux normes et recevez des recommandations sur le terrain pour améliorer la découvrabilité. Pour plus d'informations, consultez la liste des AI-assisted produits dans le Guide AWS Marketplace du vendeur.

Principaux avantages

- Accès conversationnel à Partner Central : posez des questions en langage naturel au lieu de vous retrouver dans des flux de travail complexes sur console
- Plus de temps à vendre : créez des opportunités grâce à une courte conversation au lieu de remplir un formulaire en plusieurs étapes, ce qui réduit la saisie de données et permet aux équipes commerciales des partenaires de passer plus de temps à vendre, l'agent recommandant des améliorations afin que les partenaires soumettent des opportunités de meilleure qualité et améliorent l'hygiène du pipeline
- Human-in-the-loop sécurité — Toutes les opérations d'écriture nécessitent votre approbation explicite avant d'être exécutées
- Multi-turn conversations — Affinez vos requêtes au cours d'une session sans répéter le contexte
- Analyse de fichiers — Joignez des documents (PDF, DOCX, XLSX, CSV, images) pour que l'agent les analyse parallèlement à vos questions
- Gestion centralisée des accès : contrôlez l'accès par le biais de politiques IAM avec des autorisations précises
- Test du bac à sable : testez les flux de travail dans un environnement de bac à sable isolé avant de toucher aux données de production
- Réponses en streaming — Obtenez des commentaires via SSE pendant que l'agent traite votre demande

Exemples d’utilisation

Toutes les AI-generated informations incluent un identifiant de session pour la traçabilité. Les données sont isolées en fonction des opportunités propres au partenaire connecté. Tout le contenu porte des étiquettes d'information claires et est régi par la politique d'IA AWS responsable.

Création d'opportunités

L'agent crée une nouvelle opportunité à partir d'une description en langage naturel ou d'un document téléchargé (PDF, DOCX, XLSX, TXT). Il extrait le nom du client, l'étendue du projet, la date de clôture prévue et les autres champs obligatoires, enrichit les informations du client à partir de données Web accessibles au public, valide le brouillon par rapport aux exigences de préparation d'AWS et crée l'opportunité via l'API Partner Central Selling après approbation de votre part.

- « Créez une opportunité pour Acme Corp : elle souhaite migrer son entrepôt de données vers Redshift, date de clôture cible fin du troisième trimestre, dépenses AWS mensuelles attendues de 40 000 dollars »
- « Voici les notes que j'ai prises lors de ma réunion d'hier avec GlobalTech — créez une opportunité à partir de cette transcription »
- « Utilisez ce PDF de proposition pour créer une opportunité pour la migration SAP du client »

Clonage d'opportunités

L'agent crée une nouvelle opportunité à partir d'une opportunité existante, intègre les informations relatives au nouveau client ou au projet que vous fournissez et vérifie qu'au moins un champ de différenciation a été modifié avant la soumission afin que les doublons ne soient pas rejetés par AWS Review.

- « Opportunité de clonage O1234567890 pour un nouveau client — Globex, même charge de travail, date de clôture cible fin du quatrième trimestre »
- « Créez une opportunité comme O9876543210, mais pour l'environnement de production du client plutôt que pour le bac à sable »

Informations sur les pipelines

Obtenez des informations conversationnelles sur votre pipeline de ventes. L'agent analyse la progression des étapes, les délais, les transactions bloquées et les modèles de pertes clôturées pour identifier ce qui compte le plus.

- « Quelles opportunités ont besoin de mon attention cette semaine ? »
- « Combien d'opportunités se termineront le mois prochain ? »
- « Quelles sont les principales raisons pour lesquelles nous avons perdu des opportunités au cours des 6 derniers mois ? »

Résumé de l'opportunité

L'agent synthétise le nom de l'entreprise, le secteur d'activité, le stade, les AWS dépenses mensuelles prévues, la date de clôture cible et d'autres informations clés dans un résumé concis. Il n'est pas nécessaire de scanner les champs de formulaire individuels.

- « Donnez-moi un résumé de l'opportunité O1234567890"
- « Quel est le statut actuel et les principaux détails de l'accord avec Acme Corp ? »

Génération de jeux de ventes

L'agent élabore une stratégie de vente personnalisée en combinant les détails de l'opportunité, le contexte industriel du client et les recommandations de AWS solutions pertinentes dans un plan de vente prêt à l'emploi.

- « Générez un jeu de vente pour saisir l'opportunité O1234567890"
- « Quelle est la meilleure approche pour vendre la migration vers le cloud à ce client du secteur des services financiers ? »
- « Élaborez-moi une stratégie de vente pour le contrat d'analyse de GlobalTech données »

Création d'un profil client

L'agent crée un profil d'entreprise à l'aide d'informations accessibles au public, notamment la classification du secteur, le modèle commercial (B2B/B2C/hybrid), la présence géographique, la taille

de l'entreprise, l'orientation du marché et les récents développements commerciaux. Les profils sont étiquetés « Générés à partir de données accessibles au public et d'informations issues de l'AWS IA ».

- « Créer un profil client pour Acme Corp »
- « Que savons-nous du secteur d'activité et du modèle commercial de ce client ? »
- « Rédigez un aperçu de l'entreprise pour ma prochaine réunion avec GlobalTech »

Recommandation de solution

L'agent compare vos solutions enregistrées aux exigences de l'opportunité, en indiquant le nom de la solution, sa description et en indiquant si elle est déjà associée à l'opportunité.

- « Laquelle de nos solutions correspond le mieux à l'opportunité O1234567890 ? »
- « Notre solution d'analyse de données est-elle déjà associée à cet accord ? »
- « Recommander des solutions à un client qui souhaite migrer ses charges de travail SAP »

Recommandation de financement

L'agent évalue chaque opportunité par rapport aux programmes de AWS financement disponibles en fonction des détails de l'opportunité et des critères d'éligibilité au programme. Lorsqu'une correspondance est trouvée, elle affiche le nom du programme, sa description et un raisonnement détaillé. Vous pouvez ensuite estimer les montants de financement, créer des brouillons de demandes de fonds remplis automatiquement ou en savoir plus sur les programmes par conversation. La disponibilité du budget du SCA (Strategic Collaboration Agreement) est mise en évidence le cas échéant, et toutes les actions respectent les autorisations IAM.

- « Suis-je éligible à des programmes de financement dans le cadre de l'opportunité O6789012345 ? »
- « Estimez le montant du financement pour un POC avec ce client »
- « Créez une demande de prestations MAP pour cette opportunité »

Recommandations pour la prochaine étape

L'agent évalue l'opportunité par rapport aux instructions AWS de progression par étapes, compare les données actuelles aux critères relatifs aux opportunités qualifiées et identifie exactement les

informations qu'il vous reste à collecter. Le résultat est un plan d'action hiérarchisé fondé sur les normes de AWS covente.

- « Que dois-je faire ensuite pour saisir l'opportunité O1234567890 ? »
- « Cette opportunité est-elle prête à être soumise ? Quels sont les champs manquants ? »
- « Quelles sont les conditions requises pour faire passer cette offre de Prospect à Qualified ? »

Progression des opportunités

L'agent accepte les pièces justificatives (transcriptions de réunions, notes d'appels, résumés d'e-mails), extrait les informations pertinentes, les associe aux champs d'opportunité, évalue les exigences relatives aux étapes et met à jour l'opportunité. Si des lacunes persistent, il renvoie une ventilation des exigences satisfaites par rapport aux exigences insatisfaites.

- « Voici mes notes d'appel — mettez à jour l'opportunité O1234567890 avec les détails pertinents »
- « Je joins le compte rendu de la réunion. Progressez sur cette opportunité en fonction de ce dont nous avons discuté. »
- « Consultez le résumé de cet e-mail et dites-moi à quels champs d'opportunité il répond »

Expérience agentic pour l'intégration des partenaires

L'agent automatise l'intégration des partenaires dans Partner Central et fournit une assistance guidée pour la configuration des vendeurs Marketplace et la conformité aux PRM, le tout en langage naturel.

Capacités de l'agent :

- Automatisation du profil des partenaires : scannez votre site Web, renseignez automatiquement votre profil de partenaire et gérez la visibilité, le contact avec les responsables de l'alliance, les domaines de formation et les connexions aux comptes
- Conseils de configuration pour les vendeurs : Step-by-step conseils pour l'enregistrement des vendeurs sur Marketplace, y compris les formulaires fiscaux, les services bancaires, la conformité (KYC/BAV/SU), le catalogue ESC et les rôles liés aux services
- Conseils de conformité aux PRM : évaluez l'état de préparation des PRM, récupérez les codes des produits pour le marquage des revenus et vérifiez les connexions aux comptes des filiales pour une attribution consolidée des revenus

Automatisation du profil des partenaires

L'agent scanne le site Web de votre entreprise pour extraire et remplir automatiquement votre profil de partenaire, puis vous guide pour combler les lacunes restantes. Il peut mettre à jour les champs de profil, modifier la visibilité, gérer le contact principal de votre alliance, lier les domaines de certification des formations et gérer les invitations de connexion aux comptes.

- « Pouvez-vous consulter mon site Web et remplir mon profil ? »
- « Rendez notre profil public afin que les clients AWS puissent nous trouver »
- « Mettez à jour notre contact principal de l'alliance en [nom] à [email] »
- « Connectez le compte de notre filiale EMEA »

Conseils de configuration pour les vendeurs

L'agent passe en revue l'enregistrement des vendeurs Marketplace de bout en bout, détermine le formulaire fiscal approprié en fonction de votre pays et du type d'entité, explique la configuration des opérations bancaires et des versements par région et identifie les exigences de conformité (KYC, vérification du compte bancaire, vérification de l'utilisateur secondaire) qui s'appliquent à vous.

- « Je souhaite commencer à vendre sur Marketplace, par où commencer ? »
- « De quel formulaire fiscal ai-je besoin ? Je suis une entreprise en Allemagne »
- « Ai-je besoin du KYC ? Je vends à des clients en France »
- « Comment configurer les paiements si je suis en dehors des États-Unis ? »

Conseils de conformité PRM

L'agent vérifie si votre compte existe, en vérifiant que votre compte est connecté, qu'une offre existe et que le code de votre produit est disponible pour le marquage. PRM-ready Pour les partenaires disposant de comptes filiales, il vérifie que tous les comptes vendeurs sont correctement liés pour l'attribution des revenus consolidés.

- « Suis-je conforme aux normes PRM ? »
- « Où puis-je trouver le code de mon produit à étiqueter ? »
- « Dites-moi toutes les étapes que je dois effectuer pour être conforme aux normes PRM »

Prise en main

Prêt à configurer le serveur MCP des agents Partner Central ? Consultez le guide de démarrage pour obtenir des instructions de configuration étape par étape.

Démarrage avec le serveur MCP Partner Central Agent

Ce guide explique comment configurer l'accès programmatique au serveur MCP Partner Central Agent à l'aide d'un client MCP personnalisé. Le serveur utilise le protocole HTTPS direct avec authentification SigV4. Aucun proxy ou plug-in IDE n'est requis.

Conditions préalables

Avant de commencer, assurez-vous d'avoir :

- Un compte Partner Central actif (migré vers la AWS console)
- Un AWS compte avec des autorisations IAM pour Partner Central
- AWS CLI installée et configurée avec des informations d'identification
- Accès à la région us-east-1 (Virginie du Nord)
- Connectivité HTTPS à `partnercentral-agents-mcp.us-east-1.api.aws`
- Support du protocole TLS 1.2+ dans votre client HTTP
- Un MCP-compatible client qui prend en charge la signature des demandes JSON-RPC 2.0 et SigV4

Étape 1 : configurer les autorisations IAM

Le serveur MCP Partner Central Agent nécessite des autorisations IAM à deux niveaux : accès au protocole (pour communiquer avec le point de terminaison MCP) et accès aux données (pour effectuer des opérations Partner Central).

Joindre des politiques IAM

Pour associer une politique à votre identité IAM à l'aide de la console AWS de gestion :

1. Ouvrez la console IAM.

2. Dans le volet de navigation de gauche, choisissez Utilisateurs, Groupes d'utilisateurs ou Rôles en fonction de l'identité à laquelle vous souhaitez associer la politique, puis choisissez le nom de l'utilisateur, du groupe ou du rôle spécifique.
3. Sélectionnez l'onglet Autorisations.
4. Choisissez Joindre des politiques (ou Ajouter des autorisations si c'est la première fois).
5. Dans la liste des politiques, recherchez et sélectionnez la stratégie gérée que vous souhaitez associer (par exemple, une politique personnalisée que vous avez créée à partir des blocs JSON ci-dessous).
6. Choisissez Joindre des politiques (ou Suivant puis Ajouter des autorisations) pour confirmer.

Les autorisations prennent effet immédiatement. Vous pouvez associer plusieurs politiques à la même identité.

Recommandé : utilisez la politique gérée

Le moyen le plus simple d'accorder l'accès au protocole MCP consiste à associer la politique `AWSMcpServiceActionsFullAccess` gérée à votre identité IAM. Cette politique inclut toutes les autorisations nécessaires pour interagir avec le serveur MCP.

Pour un contrôle précis, vous pouvez utiliser la clé de `aws:IsMcpServiceAction` condition de vos politiques IAM pour définir les autorisations spécifiques aux actions de service MCP.

Autorisations minimales pour l'accès au protocole MCP

Au minimum, votre identité IAM a besoin de cette action pour interagir avec le serveur MCP :

Action	Description
`partnercentral:UseSession`	Nécessaire pour créer, mettre à jour et récupérer des sessions de conversation

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

```
            ],
            "Resource": "*",
   "Condition": {
                "Bool": {
                    "aws:IsMcpServiceAction": "true"
                }
            }
        }
    ]


}
```

Autorisations d'accès aux données

Pour effectuer réellement les opérations de Partner Central par le biais de l'agent, vous avez besoin d'autorisations supplémentaires en fonction de votre cas d'utilisation.

Gestion des opportunités :

```
{
    "Effect": "Allow",
    "Action": [
        "partnercentral:List*",
        "partnercentral:Get*",
        "partnercentral:CreateOpportunity",
        "partnercentral:UpdateOpportunity",
        "partnercentral:SubmitOpportunity",
        "partnercentral:AssignOpportunity",
        "partnercentral:AssociateOpportunity",
        "partnercentral:DisassociateOpportunity"
    ],
    "Resource": "*"
}
```

Programmes de financement :

```
{
    "Effect": "Allow",
    "Action": [
        "partnercentral:ListBenefitAllocations",
        "partnercentral:ListBenefitApplications",
        "partnercentral:CreateBenefitApplication",
```

```
        "partnercentral:GetBenefitApplication",
        "partnercentral:UpdateBenefitApplication",
        "partnercentral:SubmitBenefitApplication",
        "partnercentral:AmendBenefitApplication",
        "partnercentral:CancelBenefitApplication",
        "partnercentral:RecallBenefitApplication",
        "partnercentral:AssociateBenefitApplicationResource",
        "partnercentral:DisassociateBenefitApplicationResource"
    ],
    "Resource": "*"
}
```

Accès à la Marketplace :

```
{
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:DescribeAgreement",
        "aws-marketplace:SearchAgreements",
        "aws-marketplace:ListEntities"
    ],
    "Resource": "*"
}
```

Intégration à Partner Central :

```
{
    "Effect": "Allow",
    "Action": [
        "partnercentral:ListPartners",
        "partnercentral:GetPartner",
        "partnercentral:GetProfileVisibility",
        "partnercentral:GetAllianceLeadContact",
        "partnercentral:GetProfileUpdateTask",
        "partnercentral:StartProfileUpdateTask",
        "partnercentral:CancelProfileUpdateTask",
        "partnercentral:PutProfileVisibility",
        "partnercentral:PutAllianceLeadContact",
        "partnercentral:SendEmailVerificationCode",
        "partnercentral:AssociateAwsTrainingCertificationEmailDomain",
        "partnercentral:DisassociateAwsTrainingCertificationEmailDomain",
        "partnercentral:GetAccountConnections",
```

```
        "partnercentral:GetConnectionInvitations",
        "partnercentral:CreateConnectionInvitation",
        "partnercentral:CancelConnectionInvitation",
        "partnercentral:RespondConnectionInvitation",
        "partnercentral:CancelConnection",
        "partnercentral:ManageConnectionPreferences"
    ],
    "Resource": "*"
}
```

Pour la configuration des vendeurs sur Marketplace, ajoutez également :

```
{
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:ListEntities",
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:DescribeChangeSet",
        "aws-marketplace:StartChangeSet"
    ],
    "Resource": "*"
}
```

Pour la gestion des rôles liés aux services (revente Authorizations/CPPO) :

```
{
    "Effect": "Allow",
    "Action": [
        "iam:GetRole",
        "iam:CreateServiceLinkedRole"
    ],
    "Resource": "*"
}
```

Politique d'accès complet

Pour le développement et les tests, vous pouvez combiner toutes les autorisations dans une seule politique :

```
aws iam create-policy \
    --policy-name PartnerCentralAgentsFullAccess \
    --policy-document '{
```

```
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "partnercentral:UseSession",
                    "partnercentral:List*",
                    "partnercentral:Get*",
                    "partnercentral:CreateOpportunity",
                    "partnercentral:UpdateOpportunity",
                    "partnercentral:SubmitOpportunity",
                    "partnercentral:AssignOpportunity",
                    "partnercentral:AssociateOpportunity",
                    "partnercentral:DisassociateOpportunity",
                    "partnercentral:CreateResourceSnapshot",
                    "partnercentral:CreateResourceSnapshotJob",
                    "partnercentral:StartResourceSnapshotJob",
                    "partnercentral:CreateEngagement",
                    "partnercentral:CreateEngagementInvitation",
                    "partnercentral:RejectEngagementInvitation",
                    "partnercentral:StartEngagementByAcceptingInvitationTask",
                    "partnercentral:StartEngagementFromOpportunityTask",
                    "partnercentral:CreateBenefitApplication",
                    "partnercentral:UpdateBenefitApplication",
                    "partnercentral:SubmitBenefitApplication",
                    "partnercentral:AmendBenefitApplication",
                    "partnercentral:CancelBenefitApplication",
                    "partnercentral:RecallBenefitApplication",
                    "partnercentral:AssociateBenefitApplicationResource",
                    "partnercentral:DisassociateBenefitApplicationResource",
                    "partnercentral:ListPartners",
                    "partnercentral:StartProfileUpdateTask",
                    "partnercentral:CancelProfileUpdateTask",
                    "partnercentral:PutProfileVisibility",
                    "partnercentral:PutAllianceLeadContact",
                    "partnercentral:SendEmailVerificationCode",
                    "partnercentral:AssociateAwsTrainingCertificationEmailDomain",
                    "partnercentral:DisassociateAwsTrainingCertificationEmailDomain",
                    "partnercentral:CreateConnectionInvitation",
                    "partnercentral:CancelConnectionInvitation",
                    "partnercentral:RespondConnectionInvitation",
                    "partnercentral:CancelConnection",
                    "partnercentral:ManageConnectionPreferences"
                ],
```

```
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "aws-marketplace:DescribeEntity",
                    "aws-marketplace:DescribeAgreement",
                    "aws-marketplace:SearchAgreements",
                    "aws-marketplace:ListEntities",
                    "aws-marketplace:DescribeChangeSet",
                    "aws-marketplace:StartChangeSet"
                ],
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "iam:GetRole",
                    "iam:CreateServiceLinkedRole"
                ],
                "Resource": "*"
            }
        ]
    }'
```

Read-only politique

Pour les environnements de production ou les cas d'utilisation en lecture seule, limitez les autorisations relatives aux opérations de lecture :

```
aws iam create-policy \
    --policy-name PartnerCentralAgentReadOnly \
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "partnercentral:UseSession",
                    "partnercentral:List*",
                    "partnercentral:Get*",
                    "partnercentral:ListPartners",
                    "partnercentral:GetPartner",
```

```
                    "partnercentral:GetProfileVisibility",
                    "partnercentral:GetAllianceLeadContact",
                    "partnercentral:GetProfileUpdateTask",
                    "partnercentral:GetAccountConnections",
                    "partnercentral:GetConnectionInvitations"
                ],
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "aws-marketplace:DescribeEntity",
                    "aws-marketplace:DescribeAgreement",
                    "aws-marketplace:SearchAgreements",
                    "aws-marketplace:ListEntities",
                    "aws-marketplace:DescribeChangeSet"
                ],
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "iam:GetRole"
                ],
                "Resource": "*"
            }
        ]
    }'
```

Étape 2 : Connectez votre client MCP

Le serveur MCP Partner Central Agent utilise le protocole HTTPS direct avec signature des demandes SigV4. Il n'y a pas de couche proxy : votre client MCP envoie des requêtes JSON-RPC 2.0 directement au point de terminaison.

Endpoint

```
https://partnercentral-agents-mcp.us-east-1.api.aws/mcp
```

Authentification

Toutes les demandes doivent être signées avec AWS Signature Version 4 en utilisant :

- Nom du service : `partnercentral-agents-mcp`
- Région: `us-east-1`

Initialisation de la connexion MCP

Envoyez une `initialize` demande pour établir le protocole :

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "initialize",
    "params": {
        "protocolVersion": "2025-03-26",
        "capabilities": {},
        "clientInfo": {
            "name": "my-partner-client",
            "version": "1.0.0"
        }
    }
}
```

Réponse attendue :

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
        "protocolVersion": "2025-03-26",
        "capabilities": {
            "tools": {
                "listChanged": false
            }
        },
        "serverInfo": {
            "name": "PartnerCentralAgentMCPServer",
            "version": "1.0.0"
        }
    }
}
```

Liste des outils disponibles

```
{
    "jsonrpc": "2.0",
    "id": 2,
    "method": "tools/list",
    "params": {}
}
```

Signature de vos appels avec un en-tête MCP

Lorsque vous envoyez des demandes aux agents MCP de Partner Central, nous vous recommandons d'inclure l'en-tête MCP personnalisé en utilisant les méthodes suivantes pour AWS identifier la source de l'application client, surveiller l'utilisation et auditer les performances.AWS utilise cet en-tête pour distinguer le type d'application client effectuant l'appel et pour recueillir des informations sur le taux de réussite des différentes implémentations clientes.

Méthode 1 : champ _meta (programmatic/stateless MCP)

Pour le code qui construit directement des `tools/call` requêtes MCP, fournissez le `_meta` champ sur les demandes.

```
{
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "content": [
                {
                    "type": "text",
                    "text": "List my open opportunities with expected close date in Q1
 2026"
                }
            ],
            "catalog": "AWS"
        },
        "_meta": {
            "integrator": "<Integrator's Company Name / Direct>",
            "sourceProduct": "<Integrator's Application Name>"
        }
    }
```

```
}
```

Méthode 2 : ClientInfo (agents personnalisés basés sur les sessions)

Pour les clients MCP personnalisés établissant des sessions, fournissez les informations d'en-tête MCP dans le `clientInfo` champ :

```
{
    "method": "initialize",
    "params": {
        "protocolVersion": "2024-11-05",
        "clientInfo": {
            "integrator": "<Integrator's Company Name / Direct>",
            "sourceProduct": "<Integrator's Application Name>"
        }
    }
}
```

Champs dans `clientInfo` :

- `integrator`— Nom de l'entreprise ou « Direct » pour les partenaires

  Exemple : `AWS`
- `sourceProduct`— Product/agent nom

  Exemple : `AWS CRM Connector`

Méthode 3 : paramètre d'URL (MCP hébergé uniquement)

Uniquement pour les clients MCP hébergés où l'intégrateur ne peut pas modifier les champs de protocole. Utilisez le paramètre URL :

URL du serveur : `https://mcp.partnercentral.aws?appId=<Integrator's Company Name / Direct>`

Étape 3 : vérifier votre configuration

Envoyez un simple message pour confirmer que tout fonctionne. Utilisez le `Sandbox` catalogue pour tester :

```
{
```

```
    "jsonrpc": "2.0",
    "id": 3,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "content": [
                {
                    "type": "text",
                    "text": "Hello, what can you help me with?"
                }
            ],
            "catalog": "Sandbox"
        }
    }
}
```

Si vous recevez une réponse `"status": "complete"` et une réponse par SMS de l'agent, votre configuration fonctionne correctement. La réponse comprendra également un message `sessionId` que vous pourrez utiliser pour les messages de suivi.

Étape 4 : Exécutez vos premières tâches

Renseignez-vous sur vos opportunités

```
{
    "jsonrpc": "2.0",
    "id": 4,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
            "content": [
                {
                    "type": "text",
                    "text": "List my open opportunities with expected revenue over
 $50K"
                }
            ],
            "catalog": "AWS"
        }
    }
```

```
}
```

Vérifiez l'éligibilité au financement

```
{
    "jsonrpc": "2.0",
    "id": 5,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
            "content": [
                {
                    "type": "text",
                    "text": "Am I eligible for MAP funding for opportunity
 O1234567890?"
                }
            ],
            "catalog": "AWS"
        }
    }
}
```

Récupérer l'historique des sessions

```
{
    "jsonrpc": "2.0",
    "id": 6,
    "method": "tools/call",
    "params": {
        "name": "getSession",
        "arguments": {
            "sessionId": "session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
            "catalog": "AWS"
        }
    }
}
```

Intégration des partenaires

```
{
```

```
    "jsonrpc": "2.0",
    "id": 7,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
            "content": [
                {
                    "type": "text",
                    "text": "Help me onboard to Partner Central"
                }
            ],
            "catalog": "AWS"
        }
    }
}
```

Autres tâches d'intégration à essayer :

- « Guidez-moi tout au long du processus d'entretien fiscal »
- « Pouvez-vous consulter mon site Web et remplir mon profil de partenaire ? »
- « Que dois-je encore faire pour être prêt à vendre sur Marketplace ? »

Considérations sur la sécurité

- Ne transmettez pas AWS d'informations d'identification via les paramètres de l'outil MCP. L'authentification est gérée par la signature des demandes SigV4 au niveau de la couche transport.
- Utilisez le catalogue Sandbox pour les tests et le développement. Le `"Sandbox"` catalogue fournit un environnement isolé qui n'affecte pas les données des partenaires de production.
- Appliquez les politiques IAM du moindre privilège en production. Utilisez la politique de lecture seule pour surveiller et signaler les cas d'utilisation. N'accordez des autorisations d'écriture que lorsque l'utilisateur doit mettre à jour les opportunités ou soumettre des demandes de financement.
- Passez en revue attentivement les opérations d'écriture. Le serveur utilise l'approbation humaine pour toutes les opérations d'écriture. Lorsqu'une action d'écriture est proposée, passez en revue les paramètres avant de l'approuver.
- Les données de session sont transitoires. Les sessions expirent 48 heures après leur création. Ne vous fiez pas aux sessions pour le stockage de données à long terme.

- Les téléchargements de fichiers sont dirigés vers un compartiment S3 éphémère. Les fichiers téléchargés sont stockés temporairement et ne sont pas conservés de façon permanente. Ne chargez pas de fichiers contenant des informations d'identification, des secrets ou d'autres informations sensibles.

Étapes suivantes

- Référence de configuration : référence complète pour le point de terminaison, les actions IAM, la gestion des sessions et les codes d'erreur
- Référence des outils — Documentation détaillée `sendMessage` et `getSession` outils

Référence de configuration

Cette page fournit la référence complète pour la connexion et la configuration du serveur MCP Partner Central Agent.

Endpoint

Propriété	Value
URL	`https://partnercentral-agents-mcp.us-east-1.api.aws/mcp`
Région	`us-east-1` (Virginie du Nord) uniquement
Protocole	JSON-RPC 2.0 sur HTTPS
Streaming	Server-Sent Événements (SSE)
Authentification	AWS Version 4 de la signature
Nom du service SigV4	`partnercentral-agents-mcp`
Exigence TLS	TLS 1.2 ou supérieur

Autorisations IAM

Toutes les actions IAM utilisent le `partnercentral:` préfixe, sauf indication contraire.

Accès au protocole MCP

Le moyen le plus simple d'accorder l'accès au protocole MCP consiste à associer la politique `AWSMcpServiceActionsFullAccess` gérée à votre identité IAM. Cette politique inclut toutes les autorisations nécessaires pour interagir avec le serveur MCP. Pour un contrôle précis, vous pouvez utiliser la clé de `aws:IsMcpServiceAction` condition de vos politiques IAM pour définir les autorisations spécifiques aux actions de service MCP.

Action	Description
`partnercentral:UseSession`	Création, mise à jour et récupération de sessions de conversation

Gestion des opportunités

Action	Description
`partnercentral:List*`	Répertoriez les opportunités, les solutions et les autres ressources de Partner Central
`partnercentral:Get*`	Récupérez les détails des opportunités individuelles et d'autres ressources
`partnercentral:CreateOpportunity`	Créez une opportunité
`partnercentral:UpdateOpportunity`	Modifier les champs d'opportunité (étape, date de clôture, chiffre d'affaires, etc.)
`partnercentral:SubmitOpportunity`	Soumettre une opportunité pour AWS examen
`partnercentral:AssignOpportunity`	Attribuer une opportunité à un représentant du partenaire

Action	Description
`partnercentral:AssociateOpportunity`	Lier une opportunité à une autre ressource (solution, etc.)
`partnercentral:DisassociateOpportunity`	Supprimer un lien entre une opportunité et une autre ressource
`partnercentral:StartEngagementFromOpportunityTask`	Soumettez des opportunités pour démarrer un engagement à partir d'une opportunité

Programmes de financement

Action	Description
`partnercentral:ListBenefitAllocations`	Liste des allocations d'avantages disponibles pour votre compte
`partnercentral:ListBenefitApplications`	Liste des demandes de prestations soumises
`partnercentral:CreateBenefitApplication`	Créez une nouvelle demande de prestations (MAP, POC, WMP)
`partnercentral:GetBenefitApplication`	Récupérez les détails d'une demande de prestations spécifique
`partnercentral:UpdateBenefitApplication`	Mettre à jour une demande de prestations en attente
`partnercentral:AssociateBenefitApplicationResource`	Lier une ressource à une demande de prestations
`partnercentral:DisassociateBenefitApplicationResource`	Supprimer un lien vers une ressource d'une demande de prestations

Accès à la Marketplace

Action	Description
`aws-marketplace:DescribeEntity`	Récupérer les détails d'une entité de marché
`aws-marketplace:SearchAgreements`	Contrats de recherche sur le marché
`aws-marketplace:ListEntities`	Lister les entités du marché

Agent d'intégration

Gestion des comptes partenaires (**`partnercentral`**) :

Action	Description
`partnercentral:ListPartners`	Répertorier les enregistrements de partenaires pour le compte
`partnercentral:GetPartner`	Récupérez les informations d'enregistrement et de profil des partenaires
`partnercentral:GetProfileVisibility`	Récupère la visibilité du profil actuel (PUBLIC/PRIVATE)
`partnercentral:GetAllianceLeadContact`	Récupérez les coordonnées du responsable de l'alliance
`partnercentral:GetProfileUpdateTask`	Vérifier l'état d'une mise à jour de profil asynchrone en attente
`partnercentral:StartProfileUpdateTask`	Soumettre une mise à jour du profil d'un partenaire (asynchrone)
`partnercentral:CancelProfileUpdateTask`	Annuler une tâche de mise à jour de profil en attente
`partnercentral:PutProfileVisibility`	Modifier la visibilité du profil en PUBLIC ou PRIVÉ

Action	Description
`partnercentral:PutAllianceLeadContact`	Mettre à jour le contact principal de l'alliance (nom, titre, e-mail)
`partnercentral:SendEmailVerificationCode`	Envoyer un code de vérification pour les modifications par e-mail domain/contact
`partnercentral:AssociateAwsTrainingCertificationEmailDomain`	Associer un domaine de messagerie pour le suivi des certifications de formation
`partnercentral:DisassociateAwsTrainingCertificationEmailDomain`	Supprimer un domaine d'e-mail de certification de formation associé
`partnercentral:GetAccountConnections`	Répertorier les connexions aux comptes actifs (par exemple, liens subsidiaires)
`partnercentral:GetConnectionInvitations`	Répertorier les invitations de connexion envoyées et reçues en attente
`partnercentral:CreateConnectionInvitation`	Envoyer une invitation de connexion à un autre AWS compte
`partnercentral:CancelConnectionInvitation`	Annuler une invitation de connexion sortante en attente
`partnercentral:RespondConnectionInvitation`	Accepter ou rejeter une invitation de connexion entrante
`partnercentral:CancelConnection`	Annuler une connexion active à un compte
`partnercentral:ManageConnectionPreferences`	Obtenir ou mettre à jour les préférences de partage entre les comptes connectés

Configuration du vendeur sur le Marketplace (**`aws-marketplace`**) :

Action	Description
`aws-marketplace:ListEntities`	Répertoriez les entités Marketplace (par exemple, les entités vendeuses)
`aws-marketplace:DescribeEntity`	Récupérez les informations complètes d'une entité Marketplace
`aws-marketplace:DescribeChangeSet`	Vérifier le statut d'un ensemble de modifications soumis
`aws-marketplace:StartChangeSet`	Soumettre un ensemble de modifications au Marketplace (par exemple, mise à jour du profil du vendeur)

IAM — rôle lié à un service () : **`iam`**

Action	Description
`iam:GetRole`	Vérifiez si le rôle lié au service Marketplace Resale Authorization existe
`iam:CreateServiceLinkedRole`	Créez le `AWSServiceRoleForMarketplaceResaleAuthorization` rôle

Environnements de catalogue

Catalogue	Description
`"AWS"`	Environnement de production. Toutes les opérations ont une incidence sur les données réelles des partenaires.

Catalogue	Description
`"Sandbox"`	Environnement de test isolé. Aucune incidence sur les données de production. À utiliser pour le développement et la validation.

Gestion de session

Propriété	Value
Format de l'identifiant de session	UUID v4 avec `session-` préfixe (par exemple,) `session-550e8400-e29b-41d4-a716-446655440000`
Création de session	Automatique dès le premier `sendMessage` appel sans `sessionId`
Expiration de session	48 heures après la création de la session (expiration absolue, non basée sur l'inactivité)

Chargement d'un fichier

Propriété	Value
Nombre maximum de fichiers par message	3
Limite de taille d'image	3,75 MB
Limite de taille du document	4,5 MO
Extensions autorisées	`doc`, `docx`, `pdf`, `png`, `jpeg`, `xlsx`, `csv`, `txt`
godet S3 (production)	fichiers aws-partner-central-marketplace-ephemeral-writeonly-files
Chemin de téléchargement	s3://{bucket}/{aws-account-id}/

Propriété	Value
Exigence d'URI S3	Doit inclure le paramètre de `versionId` requête

Flux de travail de téléchargement

1. Chargez votre fichier dans le compartiment S3 approprié sous le préfixe d'identifiant de votre AWS compte.
2. Notez l'URI S3, y compris celui `versionId` renvoyé par le téléchargement.
3. Incluez le fichier en tant que bloc de `document` contenu dans votre `sendMessage` appel.

Exemple de format d'URI S3 :

```
s3://aws-partner-central-marketplace-ephemeral-writeonly-files/123456789012/my-document.pdf?versionId=abc123
```

Codes d’erreur

Code	Nom	Description
`-32001`	`AUTHENTICATION_FAILURE`	La signature SigV4 n'est pas valide ou les informations d'identification ont expiré
`-31004`	`TOOL_PERMISSION_DENIED`	L'identité IAM ne dispose pas de l'`partnercentral:` action requise pour cette opération
`-32002`	`ACCESS_DENIED`	Accès général refusé (compte non inscrit, incompatibilité de région, etc.)

Code	Nom	Description
`-32004`	`LIMIT_EXCEEDED`	Limite de débit ou quota dépassé. Réessayez avec un recul exponentiel.
`-30001`	`RESOURCE_NOT_FOUND`	La ressource demandée (session, opportunité, etc.) n'existe pas
`-32600`	`INVALID_REQUEST`	JSON-RPC Demande mal formée ou paramètres non valides
`-32603`	`INTERNAL_ERROR`	Server-side erreur. Réitérez la demande.

Limites de débit

Le serveur applique des limites de débit par compte :

Opération	Taux soutenu	Mode rafale
`sendMessage`	2 demandes par minute	10
Toutes les autres opérations	10 demandes par minute	20

Si vous dépassez ces limites, le serveur renvoie le code d'erreur `-32004` (LIMIT_EXCEDED). Mettez en œuvre un ralentissement exponentiel avec instabilité chez votre client afin de gérer la limitation du débit avec élégance.

Types d'événements de streaming (SSE)

Lorsque `stream` ce paramètre est défini sur `true` dans un `sendMessage` appel, le serveur renvoie Server-Sent des événements avec les types d'événements suivants :

Type d'événement	Description
`stream_start`	Connexion au streaming établie
`assistant-response.start`	L'agent a commencé à générer une réponse
`assistant-response.delta`	Partie de texte incrémentielle de la réponse de l'agent
`assistant-response.completed`	L'agent a fini de générer la réponse
`server-tool-use`	L'agent invoque un outil interne (opération de lecture)
`server-tool-response`	Résultat d'un appel d'outil interne
`tool_approval_request`	L'agent demande l'approbation humaine pour une opération d'écriture
`stream_end`	Fermeture de la connexion au streaming
`done`	Événement final indiquant que le stream SSE est terminé

Étapes suivantes

- Référence des outils — Documentation détaillée `sendMessage` et `getSession` outils

Référence des outils

Le serveur MCP Partner Central Agent propose deux outils MCP : `sendMessage` pour toutes les interactions avec les agents et `getSession` pour récupérer l'état de session. Toutes les opérations de Partner Central (demandes d'opportunités, demandes de financement, analyse de documents) sont gérées en langage naturel via`sendMessage`.

Vue d'ensemble des outils

Outil	Description	Catégorie
`sendMessage`	Envoyez des messages à l'agent AI de Partner Central. Prend en charge le texte, les pièces jointes et les réponses d'approbation personnalisées.	Lire/Écrire
`getSession`	Récupérez l'état de la session, y compris l'historique des conversations, les événement s et les métadonnées.	Read-only

`Envoyer un message`

Outil principal pour toutes les interactions entre les agents de Partner Central AI. Utilisez cet outil pour poser des questions, demander des actions, joindre des documents à des fins d'analyse et répondre aux demandes d'approbation pour les opérations d'écriture.

L'agent gère le contexte de la conversation au cours d'une session, ce qui vous permet de poser des questions de suivi sans répéter le contexte précédent.

Parameters

- `content`(obligatoire) — Tableau de blocs de contenu. Chaque bloc doit inclure un `type` champ qui détermine la structure du bloc. Vous pouvez inclure plusieurs blocs dans un même message (par exemple, texte et pièce jointe).

  Types de blocs de contenu :

Type	Champs	Description
`text`	`type`(obligatoire), `text` (obligatoire)	Texte du message de l'utilisa teur envoyé à l'agent

Type	Champs	Description
`document`	`type`(obligatoire), `filename` (obligatoire), `s3Uri` (obligatoire)	Pièce jointe à analyser par l'agent. Ils `s3Uri` doivent inclure un `versionId` paramètre.
`tool_approval_response`	`type`(obligatoire), `toolUseId` (obligatoire), `decision` (obligatoire), `message` (facultatif)	Réponse à une demande d'approbation émanant d'un interlocuteur

- `catalog`(obligatoire) — Environnement cible pour l'opération.

  Valeurs valides : `"AWS"` (production), `"Sandbox"` (test)

- `sessionId`(facultatif) — UUID v4 identifiant une session existante à poursuivre. Omettre de créer une nouvelle session. Format :`session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`.

  Par défaut : une nouvelle session est créée automatiquement.

- `stream`(facultatif) — Activez le streaming d' Server-Sent événements (SSE) pour une réponse en temps réel.

  Valeurs valides : `true`, `false`

  Valeur par défaut : `false`

Réponse

La réponse inclut :

Champ	Description
`sessionId`	Identifiant de session pour les messages de suivi
`status`	État de la réponse : `"complete"` `"requires_approval"` , ou `"error"`

Champ	Description
`content`	Tableau de blocs de contenu de réponse provenant de l'agent

Exemples

SMS de base (nouvelle session)

Requête :

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "content": [
                {
                    "type": "text",
                    "text": "List my open opportunities with expected close date in Q1
 2026"
                }
            ],
            "catalog": "AWS"
        }
    }
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "id": 1,
    "result": {
        "content": [
            {
                "type": "text",
                "text": "I found 12 open opportunities with expected close dates in Q1
 2026. Here's a summary:\n\n1. **O1234567890** - Acme Corp Cloud Migration - $250,000 -
```

```
 Qualified stage\n2. **O1234567891** - GlobalTech Data Analytics - $180,000 - Prospect
 stage\n..."
            }
        ],
        "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
        "status": "complete"
    }
}
```

Follow-up message (session existante)

Requête :

```
{
    "jsonrpc": "2.0",
    "id": 2,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
            "content": [
                {
                    "type": "text",
                    "text": "Tell me more about O1234567890. Is it ready for
 submission?"
                }
            ],
            "catalog": "AWS"
        }
    }
}
```

Pièce jointe

Téléchargez d'abord un document sur S3, puis référencez-le dans le message :

```
{
    "jsonrpc": "2.0",
    "id": 3,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
```

```
        "arguments": {
            "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
            "content": [
                {
                    "type": "text",
                    "text": "Review this customer proposal and suggest which
 opportunity it aligns with"
                },
                {
                    "type": "document",
                    "filename": "acme-proposal.pdf",
                    "s3Uri": "s3://aws-partner-central-marketplace-ephemeral-writeonly-
files/123456789012/acme-proposal.pdf?versionId=abc123def456"
                }
            ],
            "catalog": "AWS"
        }
    }
}
```

Contraintes relatives au téléchargement de fichiers :

- 3 fichiers maximum par message
- Limite de taille d'image : 3,75 Mo
- Limite de taille du document : 4,5 Mo
- Extensions autorisées : `doc`docx,`pdf`, `png`jpeg,`xlsx`,`csv`, `txt`
- Les fichiers doivent être téléchargés sur `s3://{bucket}/{your-aws-account-id}/`
- L'URI S3 doit inclure le paramètre de `versionId` requête

Human-in-the-loop flux de travail d'approbation

Lorsque l'agent doit effectuer une opération d'écriture (par exemple, mettre à jour une opportunité, soumettre une demande de financement), il renvoie un `"requires_approval"` statut avec les détails de l'action proposée. Vous devez répondre par un bloc de `tool_approval_response` contenu.

Étape 1 — L'agent demande l'approbation :

```
{
    "jsonrpc": "2.0",
```

```
    "id": 4,
    "result": {
        "content": [
            {
                "type": "text",
                "text": "I'd like to update opportunity O1234567890 with the following
 changes:\n- Target close date: 2026-03-31\n- Expected revenue: $300,000\n- Stage:
 Qualified\n\nPlease approve, reject, or override this action."
            },
            {
                "type": "tool_approval_request",
                "toolUseId": "tool-use-98765",
                "toolName": "update_opportunity_enhanced",
                "parameters": {
                    "opportunityId": "O1234567890",
                    "targetCloseDate": "2026-03-31",
                    "expectedRevenue": 300000,
                    "stage": "Qualified"
                }
            }
        ],
        "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
        "status": "requires_approval"
    }
}
```

Étape 2 — Approuver l'action :

```
{
    "jsonrpc": "2.0",
    "id": 5,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
            "content": [
                {
                    "type": "tool_approval_response",
                    "toolUseId": "tool-use-98765",
                    "decision": "approve"
                }
            ],
```

```
            "catalog": "AWS"
        }
    }
}
```

Étape 2 (alternative) — Rejeter l'action :

```
{
    "jsonrpc": "2.0",
    "id": 5,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
            "content": [
                {
                    "type": "tool_approval_response",
                    "toolUseId": "tool-use-98765",
                    "decision": "reject",
                    "message": "The expected revenue should be $250,000, not $300,000"
                }
            ],
            "catalog": "AWS"
        }
    }
}
```

Étape 2 (alternative) — Remplacer par une réponse personnalisée :

```
{
    "jsonrpc": "2.0",
    "id": 5,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
            "content": [
                {
                    "type": "tool_approval_response",
                    "toolUseId": "tool-use-98765",
                    "decision": "override",
```

```
                    "message": "Use expected revenue of $250,000 and keep the stage as
 Prospect instead"
                }
            ],
            "catalog": "AWS"
        }
    }
}
```

Valeurs des décisions d'approbation :

Décision	Comportement
`"approve"`	Exécuter l'outil avec les paramètres proposés
`"reject"`	N'exécutez pas l'outil. Facultatif : `message` explique pourquoi.
`"override"`	Fournissez une réponse personnalisée ou des instructions modifiées via `message`

Streaming avec SSE

Activez le streaming pour recevoir des segments de réponse incrémentiels au fur et à mesure que l'agent traite votre demande :

Requête :

```
{
    "jsonrpc": "2.0",
    "id": 6,
    "method": "tools/call",
    "params": {
        "name": "sendMessage",
        "arguments": {
            "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
            "content": [
                {
                    "type": "text",
                    "text": "Analyze my pipeline and identify opportunities at risk"
                }
```

```
            ],
            "catalog": "AWS",
            "stream": true
        }
    }
}
```

Le serveur répond par un flux d'événements SSE :

```
event: stream_start
data: {"sessionId": "session-550e8400-e29b-41d4-a716-446655440000"}

event: assistant-response.start
data: {}

event: server-tool-use
data: {"toolName": "analyze_pipeline", "parameters": {}}

event: server-tool-response
data: {"toolName": "analyze_pipeline", "result": {"opportunitiesAnalyzed": 47,
 "atRisk": 5}}

event: assistant-response.delta
data: {"text": "I analyzed your pipeline of 47 opportunities and identified "}

event: assistant-response.delta
data: {"text": "5 that are at risk of slipping:\n\n"}

event: assistant-response.delta
data: {"text": "1. **O2345678901** - Close date is past due by 15 days\n"}

event: assistant-response.completed
data: {"status": "complete"}

event: stream_end
data: {}
```

Obtenir une session

Récupérez l'état actuel d'une session de conversation, y compris l'historique complet des conversations, les événements et les métadonnées. Utilisez-le pour inspecter l'état de la session, passer en revue les interactions passées ou reprendre une conversation.

Parameters

- `sessionId`(obligatoire) — UUID de la session à récupérer. Format :`session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`.
- `catalog`(obligatoire) — Environnement auquel appartient la session.

  Valeurs valides : `"AWS"`, `"Sandbox"`

Réponse

Champ	Type	Description
`sessionId`	chaîne	Identifiant de session
`createdAt`	chaîne	Horodatage ISO 8601 de la création de session
`lastActivity`	chaîne	Horodatage ISO 8601 de la dernière activité
`sequenceNumber`	entier	Numéro de séquence de l'événement en cours
`stateType`	chaîne	État de la session en cours
`events`	array	Historique complet des conversations (messages des utilisateurs, réponses des agents, utilisation des outils)
`variables`	objet	Variables de session et métadonnées
`eventCount`	entier	Nombre total d'événements au cours de la session

Exemple

Requête :

```
{
    "jsonrpc": "2.0",
    "id": 7,
    "method": "tools/call",
    "params": {
        "name": "getSession",
        "arguments": {
            "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
            "catalog": "AWS"
        }
    }
}
```

Réponse :

```
{
    "jsonrpc": "2.0",
    "id": 7,
    "result": {
        "content": [
            {
                "type": "text",
                "text": "{\"sessionId\":\"session-550e8400-e29b-41d4-
a716-446655440000\",\"createdAt\":\"2026-01-15T10:30:00Z\",\"lastActivity\":
\"2026-01-15T11:45:00Z\",\"sequenceNumber\":8,\"stateType\":\"END_TURN\",\"eventCount
\":8,\"events\":[...],\"variables\":{}}"
            }
        ]
    }
}
```

Gestion des erreurs

Toutes les erreurs suivent le format d'erreur JSON-RPC 2.0 :

```
{
    "jsonrpc": "2.0",
    "id": 1,
```

```
    "error": {
        "code": -32001,
        "message": "Authentication failed. Verify your SigV4 credentials and ensure
 they have not expired."
    }
}
```

Consultez the section called “Codes d’erreur” la liste complète des codes d'erreur et leur signification.

Stratégie de nouvelle tentative recommandée

- Pour `-32004` (LIMIT_EXCEDED) : réessayez avec un recul exponentiel commençant à 1 seconde
- Pour `-32603` (INTERNAL_ERROR) : Réessayez jusqu'à 3 fois avec un recul exponentiel
- Pour `-32001` (AUTHENTICATION_FAILURE) : actualisez les informations d'identification et réessayez
- Pour toutes les autres erreurs : ne réessayez pas automatiquement, examinez le message d'erreur et corrigez la demande

À propos de AWS API Partner Central

Les sections suivantes fournissent des informations générales sur les API.

Rubriques

- Utilisation de AWS API Partner Central
- Pris en charge AWS Régions
- Permissions
- Quotas pour AWS API Partner Central
- Journalisation pour le AWS API Partner Central
- Notifications pour le AWS API Partner Central
- Tester dans un bac à sable

Utilisation de AWS API Partner Central

Les sections suivantes fournissent des informations sur l'utilisation des API AWS Partner Central.

Rubriques

- Utilisation de AWS API du compte Partner Central
- Utilisation de AWS API de vente centralisée pour les partenaires
- Utilisation de AWS API Partner Central Benefits
- Utilisation de AWS API Partner Central Channel
- Utilisation de l'API de mesure des recettes

Utilisation de AWS API du compte Partner Central

Gestion du compte partenaire

Les comptes partenaires sont enregistrés et liés à un AWS compte existant pour offrir une IAM-based expérience complète. Cette approche élimine le besoin d'informations d'identification au niveau de l'utilisateur, garantissant ainsi que toutes les inscriptions et opérations sont gérées via l'entité IAM au sein du AWS compte.AWS Partner Central permet une intégration fluide avec les AWS services, et ce modèle assure la gestion des entités partenaires, prend en charge plusieurs catalogues et établit un système d'autorisation AWS au niveau du compte.

Les partenaires peuvent utiliser les API de compte partenaire disponibles pour enregistrer, gérer et mettre à jour leurs comptes :

API d'enregistrement des partenaires

Les partenaires peuvent enregistrer leur compte en utilisant leur AWS compte Partner Engagement. À l'aide de l'API Create, les partenaires fourniront les informations requises sur le responsable de l'alliance, accepteront les conditions de l'APN et fourniront un nom légal unique.

API de profil de partenaire

Les partenaires gèrent les profils contenant les détails de l'entreprise (nom, description, site Web, logo) avec un contrôle de public/private visibilité, une validation asynchrone et un suivi du statut permettant une mise à jour à la fois.

API de gestion de domaine

Les partenaires peuvent enregistrer et vérifier les domaines commerciaux par le biais de la validation par e-mail afin d'établir l'identité de l'organisation et d'associer la formation et les certifications des employés.

API de connexion aux partenaires

Les partenaires connectent leurs AWS comptes à d'autres AWS comptes à des fins diverses, par exemple pour collaborer à la covente avec d'autres partenaires, partager les données des AWS Marketplace Offers entre les comptes, autoriser les distributors/channel partenaires à distribute/re vendre leurs produits, etc.

API de vérification des partenaires

La vérification du partenaire est une condition préalable obligatoire à l'enregistrement d'un compte partenaire. Avant que les partenaires puissent créer un compte partenaire, ils doivent effectuer les processus de vérification commerciale et de vérification d'identité. Ces vérifications confirment que l'entreprise est légalement enregistrée et confirment l'identité de la personne qui enregistre le AWS compte.

Rubriques

- Travailler avec l'enregistrement des partenaires
- Travailler avec le profil du partenaire
- Utilisation de la gestion de domaines

- Travailler avec des connexions de comptes partenaires
- Travailler avec Partner Verification

Travailler avec l'enregistrement des partenaires

Enregistrement des partenaires

Les partenaires peuvent enregistrer leur compte en utilisant leur AWS compte Partner Engagement. À l'aide de l'API Create, les partenaires fourniront les informations requises sur le responsable de l'alliance, accepteront les conditions de l'APN et fourniront un nom légal unique.

Pendant l'enregistrement

1. Création synchrone d'une entité partenaire : les clients lancent le processus d'enregistrement des partenaires en appelant l'API Create, qui effectue la validation des entrées et crée l'entité partenaire dans le cadre de la même demande.
2. Exigences relatives aux informations relatives au responsable de l'alliance : lors de l'enregistrement du partenaire, le client doit fournir les coordonnées du responsable de l'alliance pour les communications relatives au réseau de AWS partenaires (APN).
3. Application des conditions générales — Les clients doivent accepter les conditions générales de l'APN lors de leur inscription. L'acceptation est requise pour terminer l'enregistrement et accéder à toutes les fonctionnalités. Les conditions s'appliquent à tous les avantages et fonctionnalités du programme APN.
4. Tag-On-Create Support — Dans le cadre de l'expérience d'autorisation AWS cohérente (CAE) et de Tag-Based l'autorisation, les clients peuvent ajouter des balises au moment de la création de l'entité partenaire.
5. Name-Based Validation légale — L'enregistrement renforcera l'unicité en fonction du nom légal du partenaire, garantissant ainsi l'absence de doublons d'enregistrements sous la même entité.

Après l'enregistrement

1. Support pour la gestion des balises — Une fois enregistrés, les partenaires peuvent étiqueter, détaguer et répertorier les balises d'une entité partenaire existante.
2. Récupérer l'entité partenaire : après la création d'une entité partenaire, les clients peuvent utiliser l'API List pour récupérer l'identifiant du partenaire, puis utiliser l'API Get pour récupérer les détails d'une entité partenaire spécifique.

Récapitulatif API

1. CreatePartner API : les clients lancent le processus d'enregistrement des partenaires en appelant l' CreatePartner API, qui effectue la validation des entrées et crée l'entité partenaire dans le cadre de la même demande.
2. ListPartners API : après enregistrement, les clients peuvent utiliser l'API List pour récupérer la liste des entités partenaires.
3. GetPartner API : Une fois enregistrés, les clients peuvent utiliser l'API Get pour récupérer les détails d'une entité partenaire spécifique.
4. PutAllianceLeadContact API : met à jour les principales informations de contact du responsable de l'alliance. Cette opération permet aux partenaires de transférer la désignation du principal responsable de l'alliance ou de modifier les coordonnées tout en préservant la continuité organisationnelle.
5. GetAllianceLeadContact API : met à jour les principales informations de contact du responsable de l'alliance. Cette opération permet aux partenaires de transférer la désignation du principal responsable de l'alliance ou de modifier les coordonnées tout en préservant la continuité organisationnelle.

Travailler avec le profil du partenaire

Gestion des profils

Les partenaires gèrent les profils contenant les détails de l'entreprise (nom, description, site Web, logo) avec un contrôle de public/private visibilité, une validation asynchrone et un suivi du statut permettant une mise à jour à la fois.

1. Informations de profil — Les profils des partenaires doivent inclure les informations essentielles de l'entreprise telles que le nom d'affichage, la description, l'URL du site Web, le logo IndustrySegments et PrimarySolutionType.
2. Support multilingue — Les profils des partenaires sont disponibles dans plusieurs langues. Chaque version localisée du profil contient un attribut pour les paramètres régionaux.
3. Contrôle de la visibilité du profil — Les partenaires peuvent configurer la visibilité de leur profil (public ou privé).
4. Stockage du profil — Les informations du profil seront stockées dans l'objet du compte partenaire.
5. Gestion de l'état du profil : les mises à jour des profils sont validées de manière asynchrone avant leur publication. Les partenaires peuvent vérifier l'état de la dernière mise à jour du profil

(IN_PROGRESS, SUCCEED, FAILED et CANCELLED). Seuls les profils approuvés sont visibles publiquement.

6. Contrôle de la simultanéité des demandes : une seule demande de mise à jour de profil est autorisée à la fois. Si une mise à jour de profil est déjà en cours, les clients doivent explicitement annuler la mise à jour en cours via l' CancelPartnerProfileUpdateTask API avant d'en lancer une nouvelle.
7. Annulation des tâches de mise à jour : les partenaires peuvent annuler une mise à jour de profil en cours en appelant l' CancelPartnerProfileUpdateTask API. L'annulation n'est autorisée que lorsque la mise à jour est au statut IN_PROGRESS et doit être terminée avant de commencer une nouvelle mise à jour.

Récapitulatif API

1. StartPutProfileTask API : Cette API accepte les mises à jour du profil des partenaires et effectue une validation synchrone des entrées de base.
2. GetProfileUpdateTask API : Cette API permet aux partenaires de récupérer l'état actuel de mise à jour du profil. Il est destiné à être utilisé une fois que le client a entamé une mise à jour de profil via StartPutPartnerProfile l'API, permettant aux partenaires de vérifier si leur demande est IN_PROGRESS, FAILED, SUCCEDED ou ANNULÉE.
3. CancelProfileTask API : permet à un partenaire d'annuler explicitement une tâche de mise à jour de profil dont le statut est actuellement IN_PROGRESS. Cela inclut à la fois des phases de vérification automatique et de révision manuelle. L'annulation d'une demande permet au partenaire de lancer une nouvelle mise à jour de profil sans attendre la fin du processus de vérification en cours.
4. PutProfileVisibility API : en appelant cette API, les partenaires peuvent contrôler la visibilité d'un profil quel que soit le lieu. Cela permet aux partenaires de rendre les profils publics ou privés sans en modifier le contenu.
5. GetProfileVisibility API : en appelant l' UpdatePartnerProfileVisibility API, les partenaires peuvent contrôler la visibilité d'un profil, quel que soit le lieu. Cela permet aux partenaires de rendre les profils publics ou privés sans en modifier le contenu.

Utilisation de la gestion de domaines

Gestion de domaine

Les partenaires peuvent enregistrer et vérifier les domaines commerciaux par le biais de la validation par e-mail afin d'établir l'identité de l'organisation et d'associer la formation et les certifications des employés.

Récapitulatif API

1. SendEmailVerificationCode API : lance le processus de vérification des e-mails en envoyant un code de vérification à l'adresse e-mail spécifiée. Utilisé à la fois pour la création de nouveaux contacts et pour les mises à jour d'adresses e-mail.
2. AssociateAwsTrainingCertificationEmailDomain API : associe un domaine de messagerie à la formation et à la certification AWS pour le compte partenaire, permettant ainsi la vérification automatique des certifications des employés.
3. DisassociateAwsTrainingCertificationEmailDomain API : supprime l'association entre un domaine de messagerie et la formation et la certification AWS pour le compte partenaire.

Travailler avec des connexions de comptes partenaires

Les partenaires peuvent gérer les connexions entre leurs propres comptes ou les connexions avec les AWS comptes d'autres partenaires à l'aide de l'API AWS Partner Connection. Le processus comporte deux phases :

1. Phase d'invitation à la connexion : lancement et réponse aux demandes de connexion
2. Phase de connexion active : gestion des connexions établies et de leurs activités commerciales associées

Création et envoi d'une invitation de connexion

Étape 1 : créer une invitation de connexion

La première étape de l'établissement d'une connexion partenaire consiste à créer et à envoyer une invitation de connexion à l'aide de l'`CreateConnectionInvitation`action. Lorsque vous créez une invitation, vous devez spécifier :

- Catalogue : le AWS catalogue (AWS ou Sandbox) dans lequel la connexion sera gérée

- ConnectionType: le type de relation que vous souhaitez établir. Sélectionnez l'une des méthodes suivantes :
  - **`OPPORTUNITY_COLLABORATION`**: utilisez ce type lorsque vous souhaitez envoyer une demande de connexion au compte d'un autre partenaire afin de lui envoyer des invitations à un engagement pour une opportunité de collaboration
  - **`SUBSIDIARY`**: utilisez ce type lorsque vous souhaitez connecter plusieurs comptes vendeurs AWS Marketplace que vous possédez à votre compte « principal » que vous avez associé à l'APN ou sur lequel vous êtes enregistré en tant que partenaire
- ReceiverIdentifier: ID de profil partenaire public du destinataire (pour le type de connexion Opportunity Collaboration) ou ID de AWS compte (pour le type de connexion filiale)
- Email : Votre adresse e-mail de contact pour la communication
- Nom : Votre nom en tant qu'expéditeur
- Message : description expliquant le but de la demande de connexion

Une fois créée, l'invitation passe à l'état En attente, en attente d'une action de la part du destinataire.

> **Important**
>
> En envoyant une invitation de connexion, vous consentez à révéler votre identifiant de AWS compte au destinataire s'il accepte l'invitation. Cette divulgation est nécessaire pour établir la connexion au niveau du compte.

Étape 2 : Surveillez vos invitations envoyées

Vous pouvez suivre le statut des invitations que vous avez envoyées à l'aide de l'`ListConnectionInvitations`action dont le `ParticipantType` paramètre est défini sur`SENDER`. Cela vous permet de :

- Afficher toutes les invitations sortantes
- Filtrer par statut (`PENDING`,`ACCEPTED`,`REJECTED`,`EXPIRED`)
- Filtrer par type de connexion
- Vérifiez les invitations spécifiques adressées à d'autres partenaires

Étape 3 : Annuler une invitation (facultatif)

Si vous devez retirer une invitation en attente avant qu'elle ne soit acceptée, vous pouvez utiliser cette `CancelConnectionInvitation` action. Remarque :

- Vous ne pouvez annuler les invitations que si l'état est En attente
- Une fois qu'une invitation est acceptée et qu'une connexion est établie, elle ne peut pas être annulée
- Pour mettre fin à une connexion établie, vous devez plutôt utiliser l'`CancelConnection`action

Réception et réponse aux invitations de connexion

Étape 1 : Répertorier les invitations entrantes

Pour consulter les invitations de connexion que vous avez reçues, utilisez l'`ListConnectionInvitations`action dont le ParticipantType paramètre est défini sur`RECEIVER`. Vous pouvez filtrer par :

- Type de connexion
- État (`PENDING`,`ACCEPTED`,`REJECTED`)
- Identifiant de l'expéditeur

Étape 2 : Vérifiez les détails de l'invitation

Utilisez cette `GetConnectionInvitation` action pour afficher les détails complets d'une invitation spécifique, notamment :

- Nom et adresse e-mail de contact de l'expéditeur
- Message d'invitation expliquant l'objectif
- Type de connexion demandé
- Date d’expiration
- Informations de profil public de l'expéditeur

Étape 3 : Accepter ou rejeter l'invitation

Après avoir examiné les détails de l'invitation, deux options s'offrent à vous :

Option A : Accepter l'invitation

Utilisez l'`AcceptConnectionInvitation`action pour établir la connexion. Lorsque vous acceptez :

- Une nouvelle connexion est créée à l'état actif
- Le statut de l'invitation passe à Acceptée
- L'identifiant de votre AWS compte est révélé à l'expéditeur
- Vous pouvez commencer à mener des activités commerciales activées par le type de connexion

> **Important**
>
> Consentement important : En acceptant une invitation de connexion, vous consentez à révéler votre identifiant de AWS compte à l'expéditeur. Cette divulgation est nécessaire pour établir la connexion au niveau du compte.

Option B : Refuser l'invitation

Utilisez cette `RejectConnectionInvitation` action si vous ne souhaitez pas établir la connexion. Vous pouvez éventuellement fournir un motif de rejet. En cas de rejet :

- Le statut de l'invitation passe à Rejeté
- Aucune connexion n'est établie
- L'invitation n'apparaîtra plus dans votre liste d'invitations en attente

Expiration automatique

Si aucune action n'est entreprise pendant la période d'expiration, le système fait automatiquement passer l'invitation à l'état Expiré. Les invitations expirées ne peuvent être ni acceptées ni rejetées.

Gestion des connexions actives

Afficher vos connexions

Une fois la connexion établie, les deux parties peuvent la consulter et la gérer à l'aide des actions suivantes :

`ListConnections`: Récupérez la liste de vos connexions avec des options de filtrage :

- Filtrer par type de connexion et par état (par exemple,`OPPORTUNITY_COLLABORATION:ACTIVE`)
- Filtrer en fonction de l'identifiant de l'autre partie
- Afficher les informations récapitulatives pour chaque connexion

`GetConnection`: Récupérez les détails complets d'une connexion spécifique, notamment :

- Tous les types de connexion associés à cette relation de partenariat
- État de chaque type de connexion (active ou interrompue)
- Informations de contact figurant sur l'invitation originale
- AWS les identifiants de compte et les identifiants de profil public des deux parties
- Horodatages de création et de fin pour chaque type de connexion

Ajouter des types de connexion à une connexion existante

Si vous avez déjà une connexion active avec un partenaire et que vous souhaitez établir un nouveau type de relation, vous pouvez envoyer une nouvelle invitation de connexion avec un autre type de connexion. Lors de l'acceptation :

- Le nouveau type de connexion est ajouté à la connexion existante
- Les deux types de connexion restent gérables indépendamment
- Le même identifiant de connexion est conservé pour la relation

> **Note**
>
> Il ne peut y avoir qu'une seule connexion active d'un type donné entre deux comptes dans un catalogue donné à la fois.

Types de connexion de terminaison

L'une ou l'autre des parties peut mettre fin à un type de connexion spécifique à l'aide de `CancelConnection` cette action. En cas de résiliation :

- Spécifiez l'ID de connexion et le type de connexion à terminer
- Indiquez le motif de la résiliation

- L'état du type de connexion passe à `CANCELED`
- Les autres types de connexion au sein de la même connexion ne sont pas affectés

Fin complète de la connexion

Lorsque tous les types de connexion d'une connexion sont interrompus, la connexion elle-même passe à l'état Terminé. Une connexion complètement interrompue :

- Ne peut pas être réactivé, mais vous pouvez envoyer une nouvelle demande de connexion pour continuer à utiliser ce compte
- Reste dans le système pour la tenue des dossiers historiques
- Peut être consulté à l'aide de l'`GetConnection`API
- Nécessite une nouvelle invitation pour rétablir la relation

Surveillance des événements de connexion

Les partenaires doivent surveiller les événements liés à la connexion à l'aide d'Amazon EventBridge afin de rester informés des points suivants :

- Nouvelles invitations de connexion entrantes
- Changements de statut des invitations envoyées (`ACCEPTED`,`REJECTED`,`EXPIRED`)
- Terminaisons de connexion initiées par l'autre partie

À la réception d'un événement, utilisez l'action Get API appropriée pour obtenir les derniers détails :

- `GetConnectionInvitation`pour les mises à jour des invitations
- `GetConnection`pour les mises à jour de connexion

Évènements possibles

- Invitation de connexion créée : avertit le compte du destinataire des invitations de connexion entrantes.
- Invitation de connexion annulée : avertit le compte du destinataire lorsqu'un expéditeur annule une invitation de connexion entrante.
- Invitation de connexion expirée : avertit les comptes expéditeur et destinataire lorsqu'une invitation de connexion expire sans action.

- Invitation de connexion rejetée : avertit le compte de l'expéditeur lorsque son invitation de connexion est rejetée par le destinataire.
- Invitation de connexion acceptée : avertit le compte de l'expéditeur lorsque son invitation de connexion est acceptée et qu'une connexion est établie.
- Connexion annulée : avertit les deux parties connectées lorsque toutes les connexions sont complètement interrompues.

Comprendre les états de connexion

États d'invitation à la connexion

State	Description	Peut passer à
`PENDING`	État initial lors de la création ; en attente d'une action du récepteur	`ACCEPTED`, `REJECTED`, `EXPIRED`, `CANCELED`
`ACCEPTED`	Récepteur accepté ; connexion créée	Aucun (état du terminal)
`REJECTED`	Destinataire refusé ; contient un motif de rejet facultatif	Aucun (état du terminal)
`EXPIRED`	System-expired en raison de l'absence d'action pendant la période d'expiration	Aucun (état du terminal)
`CANCELED`	L'expéditeur a retiré l'invitation avant son acceptation	Aucun (état du terminal)

États du type de connexion

State	Description
`ACCEPTED`	Le type de connexion est opérationnel ; les activités commerciales associées sont activées

State	Description
CANCELED	Type de connexion terminé par l'une ou l'autre des parties

Bonnes pratiques

1. Communication claire : insérez des messages détaillés et clairs dans vos invitations de connexion expliquant l'objectif et la collaboration attendue.
2. Réponses rapides : surveillez les invitations de connexion et répondez-y rapidement pour éviter leur expiration automatique.
3. Révision régulière : passez régulièrement en revue vos connexions actives pour vous assurer qu'elles répondent toujours aux besoins actuels de votre entreprise.
4. Résiliation appropriée : Lorsque vous mettez fin à une relation commerciale, utilisez l' CancelConnection action avec une raison claire pour maintenir une communication professionnelle.
5. Surveillance des événements : configurez des EventBridge règles pour recevoir automatiquement des notifications en cas de modification de l'état de la connexion, afin de rester informé des mises à jour importantes.
6. Gestion des types de connexion : déterminez quels types de connexion sont nécessaires avant d'établir des connexions, car chaque type offre des fonctionnalités commerciales différentes.
7. Sensibilisation à la sécurité : N'oubliez pas que l'établissement d'une connexion révèle les identifiants des AWS comptes entre les parties. Connectez-vous uniquement à des partenaires de confiance.

Les types de connexion et leurs objectifs

Type de connexion	Objectif	Cas d’utilisation
OPPORTUNITY_COLLABORATION	Permet la collaboration sur les opportunités de covente avec d'autres partenaires	Si vous souhaitez qu'un autre partenaire ait accès à votre opportunité ou vice versa, utilisez ce type de connexion. Cette connexion vous permet d'envoyer des invitations à

Type de connexion	Objectif	Cas d'utilisation
		participer à des opportunités à des partenaires connectés.
`SUBSIDIARY`	Établit des connexions entre vos multiples comptes vendeur AWS Marketplace et votre compte partenaire principal	À utiliser lorsque vous possédez plusieurs comptes vendeur AWS Marketplace que vous souhaitez connecter à votre compte principal sur lequel vous êtes enregistré en tant que partenaire et vendeur.

Travailler avec Partner Verification

Gérer la vérification des partenaires

La vérification du partenaire est une condition préalable obligatoire à l'enregistrement d'un compte partenaire. Avant que les partenaires puissent créer un compte partenaire, ils doivent effectuer les processus de vérification commerciale et de vérification d'identité. Ces vérifications confirment que l'entreprise est légalement enregistrée et confirment l'identité de la personne qui enregistre le AWS compte.

Les partenaires peuvent utiliser les API de vérification disponibles pour lancer et surveiller les processus de vérification :

Pendant la vérification

1. Initiation à la vérification commerciale — Les partenaires initient la vérification commerciale en appelant l'`StartVerification`API avec les informations d'enregistrement de l'entreprise, notamment le nom légal, le numéro d'enregistrement, le code du pays et la juridiction de constitution.
2. Initiation à la vérification d'identité — Les partenaires initient la vérification d'identité en appelant l'`StartVerification`API avec les informations du titulaire. L'API renvoie une URL de complétion sécurisée dans laquelle le déclarant effectue le flux de travail de vérification d'identité à l'aide d'une pièce d'identité émise par le gouvernement.

3. Surveillance de l'état de vérification — Les partenaires utilisent l'`GetVerification`API pour récupérer le statut actuel et les détails des processus de vérification. L'API renvoie le type de vérification, le statut, les horodatages et les détails spécifiques à la vérification.
4. Fin de la vérification — La vérification commerciale et la vérification d'identité doivent toutes deux atteindre le `SUCCEEDED` statut requis pour que les partenaires puissent procéder à la création d'un compte partenaire. Les vérifications échouées ou expirées doivent être résolues avant l'enregistrement du compte.

Après vérification

1. Création d'un compte partenaire — Une fois la vérification réussie, les partenaires créent leur compte partenaire à l'aide de l'`CreatePartner`API décrite dans la documentation du compte partenaire.
2. Récupération des dossiers de vérification — Les partenaires peuvent récupérer les informations de vérification à tout moment à l'aide de l'`GetVerification`API avec l'identifiant de vérification renvoyé lors du lancement.

Récapitulatif API

1. **`StartVerification`**API : les partenaires lancent les processus de vérification en appelant l'`StartVerification`API. Pour la vérification des entreprises, l'API valide les informations d'enregistrement des entreprises. Pour la vérification de l'identité, l'API génère une URL de fin limitée dans le temps permettant au déclarant de terminer la vérification d'identité.
2. **`GetVerification`**API : après avoir lancé la vérification, les partenaires utilisent l'`GetVerification`API pour récupérer le statut et les détails de la vérification. L'API renvoie l'état actuel, les horodatages et les informations spécifiques à la vérification.

Utilisation de AWS API de vente centralisée pour les partenaires

Cette référence d'API AWS Partner Central est conçue pour aider les AWS partenaires à intégrer les systèmes de gestion de la relation client (CRM) à Partner Central. L'API automatise les interactions avec Partner Central, ce qui contribue à garantir des engagements efficaces dans le cadre d'activités commerciales conjointes.

L'API fournit des fonctionnalités AWS d'API standard. Vous pouvez y accéder en utilisant les actions d'API ou en utilisant un AWS SDK adapté à votre langage de programmation ou à votre plateforme. Pour plus d'informations, consultez Getting Started with AWS and Tools to build on AWS.

Fonctionnalités proposées par AWS API Partner Central

1. Gestion des opportunités : Facilite la gestion des opportunités de covente en AWS utilisant des actions d'API telles que `CreateOpportunity`,`UpdateOpportunity`,`ListOpportunities`,`GetOpportunity`, et`AssignOpportunity`.
2. AWS gestion des références : Facilite la réception de références partagées en AWS utilisant des actions telles que`ListEngagementInvitations`, `GetEngagementInvitation`,`StartEngagementByAcceptingInvitation`, et. `RejectEngagementInvitation`
3. Association d'entités : associez des entités connexes telles que AWS des produits, des solutions partenaires, AWS Marketplace des solutions, des AWS Marketplace produits et des offres AWS Marketplace privées à des opportunités en utilisant les actions `AssociateOpportunity` et`DisassociateOpportunity`.
4. Afficher les détails des AWS opportunités : utilisez cette `GetAWSOpportunitySummary` action pour récupérer des résumés en temps réel des AWS opportunités liées à vos opportunités.
5. Répertorier les solutions : fournit des API de liste pour répertorier les solutions proposées par les partenaires`ListSolutions`.
6. Abonnement aux événements : les partenaires peuvent s'abonner à des mises à jour en temps réel sur les opportunités en écoutant des événements tels que l'opportunité créée, l'opportunité mise à jour, l'invitation d'engagement acceptée, l'invitation d'engagement rejetée et l'invitation d'engagement créée avec Amazon EventBridge.

Régions et terminaux pris en charge

L'API AWS Partner Central est disponible dans la région USA Est (Virginie du Nord).

Région	Endpoint
us-east-1	partnercentral-selling.us-east-1.api.aws

Les partenaires peuvent tester et valider les intégrations d'API dans un environnement sandbox sécurisé. Cela vous permet de tester les actions de l'API sans affecter les données en temps réel. Pour de plus amples informations, veuillez consulter Tester dans un bac à sable pour AWS API de vente centralisée pour les partenaires.

Vendre des entités d'API

AWS Les entités Partner Central représentent les principaux composants commerciaux utilisés dans les engagements de covente entre partenaires et AWS. Ces entités encapsulent les informations relatives aux opportunités, aux solutions, aux produits et aux offres, permettant une collaboration fluide et une gestion des activités de vente conjointes. Les sections suivantes décrivent les entités principales de l'API de vente Partner Central.

Opportunity

Une opportunité est une vente ou une transaction potentielle qu'une entreprise identifie et poursuit activement. Il s'agit d'un prospect ou d'un prospect qualifié ayant un besoin spécifique auquel les produits ou services de l'entreprise peuvent répondre. Les opportunités sont généralement suivies dans un pipeline de vente ou un système CRM et constituent la base des revenus futurs. Une gestion efficace des opportunités implique de développer des prospects tout au long du processus de vente, de la qualification initiale à la clôture. Pour plus d'informations, voir Utilisation de vos opportunités et types de données

Solutions partenaires

Représente une solution partenaire (appelée offre sur AWS Partner Central), qui est un produit logiciel ou un cabinet de conseil créé et fourni par AWS des partenaires. Les solutions destinées aux partenaires aident les clients à relever des défis commerciaux spécifiques ou à atteindre des objectifs spécifiques à l'aide de AWS services. Pour plus d'informations, voir Qu'est-ce qu'une solution ?

AWS product

Représente un AWS service ou un produit spécifique.AWS propose une large gamme de produits et services conçus pour fournir des solutions d'infrastructure évolutives, fiables et rentables. Les partenaires peuvent obtenir la dernière liste de AWS produits sur la page d'importation en bloc de Partner Central (Démarrer l'importation >AWS Produits et solutions). Pour plus d'informations, vous pouvez en savoir plus sur AWS les produits ou consulter la liste de tous les AWS produits.

AWS Marketplace offre privée

AWS Marketplace L'offre privée est une fonctionnalité qui permet aux AWS Marketplace vendeurs de proposer des prix et des conditions spécifiques à des AWS clients individuels. Par le biais d'offres privées, les vendeurs peuvent négocier des prix personnalisés, des calendriers de paiement et des conditions de licence utilisateur final.AWS les clients peuvent obtenir des solutions logicielles qui répondent à leurs besoins spécifiques, tout en bénéficiant éventuellement de conditions ou de prix plus avantageux par rapport aux offres standard. Le processus d'offre privée implique que le vendeur crée une offre unique avec des conditions personnalisées, qui est ensuite partagée en privé avec le AWS client désigné pour examen et acceptation. Pour plus d'informations, consultez la section Offres privées dans AWS Marketplace.

Invitation à des

L'invitation à l'engagement fait référence à une demande officielle adressée aux partenaires AWS pour collaborer sur une recommandation spécifique. Cela permet AWS au partenaire de travailler ensemble pour faire avancer l'opportunité. L'invitation peut être acceptée ou rejetée par le partenaire.

Rubriques

- Tirer parti de vos opportunités
- Travailler avec les opportunités de AWS
- Travailler avec les mises à jour des opportunités
- Travailler avec des opportunités multipartenaires
- Associer, dissocier et attribuer des opportunités
- Travailler avec vos prospects
- Utilisation des informations relatives au dimensionnement des transactions
- Bonnes pratiques

Tirer parti de vos opportunités

Qu'est-ce qu'une opportunité ?

Au cours des premières étapes du processus de vente, un représentant commercial évalue si une personne intéressée (appelée prospect) a le potentiel de devenir client. Cette phase d'évaluation et de validation est appelée Qualification. Une fois qu'un prospect est jugé qualifié et qu'il est considéré comme ayant une probabilité plus élevée de se convertir en client, il est alors classé comme opportunité.

Tirez parti de vos opportunités

Les partenaires peuvent gérer les opportunités créées dans leurs systèmes CRM et les synchroniser avec AWS Partner Central à l'aide de l'API de vente AWS Partner Central. Cela permet aux partenaires de suivre et de gérer les opportunités du lancement à la clôture.

Création d'une opportunité

La première étape de la gestion des opportunités consiste à créer une opportunité à l'aide de l'`CreateOpportunity`action. Cela crée une opportunité avec le `Lifecycle.ReviewStatus` set to`Pending Submission`. Cependant, l'opportunité n'est pas encore soumise AWSà validation.

Une fois l'opportunité créée, vous devez associer au moins une solution,AWS Marketplace une solution ou un AWS Marketplace produit partenaire à l'opportunité à l'aide de l'`AssociateOpportunity`action. Cette action définit clairement ce que l'opportunité tente de vendre. Vous pouvez consulter la liste complète des solutions disponibles dans votre compte à l'aide de l'`ListSolutions`API, et vous pouvez associer entre une et dix solutions.

Facultativement, vous pouvez également associer des AWS produits pertinents à l'opportunité à l'aide de l'`AssociateOpportunity`action. Cette étape aide les équipes AWS commerciales à comprendre quels AWS produits devraient être vendus en fonction de l'opportunité.

Une fois que la solution ou le produit requis est associé et que les AWS produits sont éventuellement liés, vous pouvez démarrer un engagement sur l'opportunité en utilisant l'`StartEngagementFromOpportunityTask`action. C'est à ce moment que vous soumettez l'opportunité de démarrer un engagement.

Processus de révision

Après avoir démarré l'engagement sur l'opportunité à l'aide de l'`StartEngagementFromOpportunityTask`action, l'opportunité entre dans la phase de AWS validation, et elle `Lifecycle.ReviewStatus` est définie sur`Submitted`. Aucune modification ne peut être apportée à l'opportunité tant que le processus de révision n'est pas terminé.

Au cours de cette phase de validation,AWS veille à ce que les détails de l'opportunité soient exacts et complets. Pendant que la validation est en cours, le `Lifecycle.ReviewStatus` est défini sur`In-Review`.

Si des modifications ou des informations supplémentaires sont requises de la part du partenaire,AWS définit la `Lifecycle.ReviewStatus` valeur et toute mise à jour requise est communiquée via le `Lifecycle.ReviewComments` champ. `Action Required`

Une fois que l'opportunité a été validée, `Lifecycle.ReviewStatus` elle est `Approved` modifiée, ce qui la prépare aux activités de covente.

Les partenaires doivent surveiller l'`Opportunity Updated`événement à l'aide d'Amazon EventBridge. Cela les informera de tout changement de statut ou de tout commentaire provenant de AWS. Dès réception de l'événement, les partenaires peuvent utiliser l'action de l'`GetOpportunity`API pour récupérer les derniers détails de l'opportunité et vérifier le `Lifecycle.ReviewStatus` champ.

Résolution des problèmes de validation

Si le paramètre `Lifecycle.ReviewStatus` est défini sur`Action Required`, les partenaires doivent résoudre les problèmes mis en évidence par AWS. Pour résoudre ces problèmes, les partenaires peuvent mettre à jour l'opportunité à l'aide de l'action `UpdateOpportunity` API.

Pendant l'`Action Required`état, seuls certains champs sont modifiables. Ces champs comprennent :

1. `Customer.Account.Address.City`
2. `Customer.Account.Address.Country`
3. `Customer.Account.Address.PostalCode`
4. `Customer.Account.Address.StateOrRegion`
5. `Customer.Account.Address.StreetAddress`
6. `Customer.Account.WebsiteUrl`
7. `LifeCycle.TargetCloseDate`
8. `Project.ExpectedCustomerSpend.Amount`
9. `Project.ExpectedCustomerSpend.Currency`
10. `Project.CustomerBusinessProblem`
11. `PartnerOpportunityIdentifier`

Après avoir apporté les modifications nécessaires, l'opportunité entre à nouveau dans la phase de validation, et le processus se répète jusqu'à ce que l'opportunité `Lifecycle.ReviewStatus` soit définie sur ou. `Approved` `Disqualified`

Post-Approval mises à jour

Une fois que l'opportunité se `Approved` présente, les partenaires peuvent continuer à l'actualiser selon les besoins en utilisant l'`UpdateOpportunity`action, facilitant ainsi des activités de covente fluides.

Les partenaires doivent continuer à suivre les `Opportunity Updated` événements via Amazon EventBridge afin de rester informés de toute modification. Pour plus d'informations sur le suivi des AWS mises à jour, consultez la section « Utilisation des mises à jour relatives aux opportunités ». Les partenaires peuvent également mettre à jour certains champs en fonction des règles de validation commerciales.

Travailler avec les opportunités de AWS

1. Recevoir le AWS Opportunity

Les opportunités sont partagées avec les partenaires lorsqu'un responsable AWS commercial associe un partenaire au système CRM AWS d'une opportunité. Elles sont appelées AWS opportunités, distinctes des opportunités créées pour le compte du partenaire.

Lorsqu'un partenaire est associé à une AWS opportunité,AWS crée une invitation d'engagement contenant un sous-ensemble de données provenant de l'AWS opportunité. Les partenaires recevront un événement créé par invitation à l'engagement.

2. Révision de l'invitation à l'engagement

L'invitation d'engagement contient des informations essentielles telles que `Project.TitleProject.CustomerUseCase`,`Lifecycle.Stage`,`Project.CustomerBusinessProb` et quelques champs supplémentaires. Les partenaires peuvent utiliser ces données pour décider s'ils souhaitent saisir l'opportunité.

Toutefois, les champs suivants de l'AWS opportunité ne sont pas inclus dans l'invitation d'engagement :

```
Customer.Account.Address.StreetAddress
Customer.Account.Address.City
Customer.Account.Address.PostalCode
Customer.Contact
Customer.Account.AWSAccountId
Project.OtherSolutionDescription
```

```
RelatedEntityIdentifiers
```

3. Gestion de l'invitation de fiançailles

Dès réception d'un événement créé par invitation à l'engagement, les partenaires peuvent utiliser l'`GetEngagementInvitation`action pour récupérer les détails de l'AWS opportunité.

Si le partenaire décide de ne pas saisir l'opportunité, il peut rejeter l'invitation en utilisant l'`RejectEngagementInvitation`action, ainsi que les mesures requises`RejectionReason`. Une fois rejetée, l'accès à l'opportunité est perdu.

Pour accepter l'invitation et continuer, les partenaires doivent utiliser l'`StartEngagementByAcceptingInvitationTask`action. Il s'agit d'une action asynchrone qui exécute les tâches suivantes de manière séquentielle :

1. Accepte l'invitation de fiançailles.
2. Crée une nouvelle opportunité dans le compte du partenaire à l'aide des données de l'AWS opportunité.
3. Inclut les informations supplémentaires requises pour identifier le client sur le compte du partenaire.

Une fois terminé, un événement Opportunity Created est déclenché avec l'ID d'opportunité correspondant. Les partenaires peuvent ensuite utiliser cet identifiant dans l'`GetOpportunity`action pour récupérer tous les détails de l'opportunité.

Si le partenaire n'utilise pas les événements, il peut appeler `StartEngagementByAcceptingInvitationTask` à nouveau avec la même charge utile pour vérifier le dernier statut.

4. Gérer le AWS Opportunité Post-Acceptance

Une fois que l'opportunité est enregistrée dans le compte du partenaire, elle peut être gérée et mise à jour comme toute autre opportunité dans le système. Les partenaires peuvent utiliser des actions telles que `UpdateOpportunity` pour apporter les modifications nécessaires.

Pour plus de détails sur le suivi des AWS mises à jour et la gestion des mises à jour des opportunités, consultez la Travailler avec les mises à jour des opportunités section.

Travailler avec les mises à jour des opportunités

Opportunités d'actualisation

Les partenaires peuvent utiliser cette `UpdateOpportunity` action pour mettre à jour les opportunités, avec des règles spécifiques régissant les champs mis à jour et le moment de leur mise à jour :

1. Les mises à jour ne peuvent pas être effectuées si `Lifecycle.ReviewStatus` c'est le cas `Submitted` ou`In-Review`.
2. Avant la soumission, les partenaires peuvent effectuer des mises à jour, mais AWS ne les traiteront que si l'`StartEngagementFromOpportunityTask`action est invoquée.
3. Lorsque l'opportunité se présente `Submitted` ou que le `In-review` statut est atteint, toutes les mises à jour sont bloquées.
4. Si le `Action Required` statut de l'opportunité est en cours,AWS ouvre les champs de sélection pour les mises à jour. Ces champs comprennent :
   - `Customer.Account.Address.City`
   - `Customer.Account.Address.Country`
   - `Customer.Account.Address.PostalCode`
   - `Customer.Account.Address.StateOrRegion`
   - `Customer.Account.Address.StreetAddress`
   - `Customer.Account.WebsiteUrl`
   - `LifeCycle.TargetCloseDate`
   - `Project.ExpectedCustomerSpend.Amount`
   - `Project.ExpectedCustomerSpend.CurrencyCode`
   - `Project.ExpectedCustomerSpend.EstimationURL`
   - `Project.ExpectedCustomerSpend.Frequency`
   - `Project.ExpectedCustomerSpend.TargetCompany`
   - `Project.CustomerBusinessProblem`
   - `PartnerOpportunityIdentifier`
5. Après le processus de révision (c'est-à-dire le moment `Lifecycle.ReviewStatus` défini sur`Approved`), les champs suivants ne peuvent pas être mis à jour :
   - `Customer.Account.Address.Country`

- `Customer.Account.Address.PostalCode`
- `Customer.Account.Industry`
- `Customer.Account.WebsiteUrl`
- `Project.CustomerBusinessProblem`
- `PartnerOpportunityIdentifier`
- `Project.Title`

6. Pour tous les autres champs, des mises à jour peuvent être effectuées à l'aide de `UpdateOpportunity` cette action. Cependant, des restrictions supplémentaires peuvent s'appliquer en fonction des règles commerciales relatives au programme ou au type d'opportunité spécifique. Pour plus de détails, reportez-vous à la section Validations au niveau du champ.
7. Pour toutes les mises à jour effectuées via l'interface utilisateur et l'API, l'événement Opportunity Updated est généré.

Recevoir des mises à jour de AWS sur les opportunités

AWS met généralement à jour AWS les opportunités, et chaque fois qu'une mise à jour est effectuée, un événement Opportunity Updated est généré.

Pour récupérer les dernières mises à jour depuis AWS, les partenaires doivent invoquer deux actions distinctes

1. `GetOpportunity`pour récupérer les détails de l'opportunité du partenaire.
2. `GetAWSOpportunitySummary`pour récupérer des résumés en temps réel des données relatives aux AWS opportunités.

La plupart des mises à jour régulières de AWS seront disponibles via la `GetAWSOpportunitySummary` réponse. Cependant, il AWS peut parfois mettre à jour les attributs directement dans l'opportunité du partenaire.

Pour utiliser les mises à jour de AWS

1. Invoquez l'`GetAWSOpportunitySummary`action pour récupérer les derniers détails de l'AWS opportunité.
2. Si les modifications doivent être reflétées dans l'opportunité du partenaire, utilisez l'`UpdateOpportunity`action pour copier les données pertinentes sur l'opportunité du partenaire.

Les partenaires peuvent choisir d'automatiser ce processus en tant que mécanisme de mise à jour directe ou de mettre en œuvre un processus de révision manuel pour valider et mettre à jour les données.

Travailler avec des opportunités multipartenaires

AWS Partner s peuvent travailler ensemble sur des opportunités en AWS tant que participant actif.

Rubriques

- Engagements et instantanés
- Création d'une politique personnalisée pour ResourceSnapshotJobRole
- Inviter les partenaires à saisir une opportunité
- Récupération des détails de l'invitation à un engagement
- Répondre à une invitation de fiançailles
- Examen et mise à jour des opportunités multipartenaires
- Utiliser des instantanés pour recevoir les mises à jour des partenaires
- Afficher les membres de l'engagement
- Surveillance de l'état des tâches liées aux captures d'écran

Engagements et instantanés

Un engagement est une ressource détenue AWS pour la collaboration entre plusieurs AWS Partner comptes et AWS sur une opportunité client. Les engagements facilitent le partage sécurisé des informations entre les partenaires et la collaboration tout en préservant la propriété et le contrôle individuels des opportunités. Les engagements englobent les opportunités qui proviennent à la fois de partenaires AWS et de partenaires, en AWS tant que participant actif. Les partenaires peuvent inviter AWS ou d'autres partenaires à participer à un engagement en utilisant`EngagementInvitations`. Lorsque les partenaires invités acceptent, ils reçoivent leur propre opportunité dans le cadre du même engagement.

Les membres de l'engagement partagent leurs progrès par le biais de instantanés, c'est-à-dire de copies instantanées et immuables de champs spécifiques provenant d'une ressource sous-jacente telle qu'une opportunité. Des instantanés sont créés dans le cadre de l'engagement et partagés avec les membres. Lorsque la ressource sous-jacente change, le propriétaire peut créer une nouvelle révision instantanée pour refléter les mises à jour.AWS fournit des tâches instantanées, c'est-à-

dire des tâches appartenant au client qui créent automatiquement de nouvelles révisions lorsque la ressource change. Une fois partagés, les instantanés restent accessibles en permanence aux membres de l'engagement.

Création d'une politique personnalisée pour ResourceSnapshotJobRole

La collaboration sur des opportunités impliquant plusieurs partenaires nécessite de partager des instantanés de vos opportunités avec d'autres partenaires dans le cadre d'un engagement. Pour conserver l'accès aux informations les plus récentes sur les opportunités, vous devez créer un rôle personnalisé appelé`ResourceSnapshotJobRole`. Ce rôle permet au système de créer des instantanés de vos opportunités en votre nom et de récupérer des instantanés d'autres partenaires participant au même engagement. Sans ce rôle, les mises à jour que vous apportez à votre opportunité ne seront pas visibles par les autres partenaires participant à l'engagement, qui continueront de voir des instantanés périmés de vos données d'opportunité.

> **Note**
>
> Vous pouvez utiliser un autre nom que`ResourceSnapshotJobRole`. Si vous utilisez un autre nom, remplacez toutes les instances de `ResourceSnapshotJobRole` dans les politiques suivantes par le nom de votre politique.

Pour créer ce rôle, accédez à votre console IAM et créez-le avec la `ResourceSnapshotJobRole` politique de confiance personnalisée suivante :

JSON

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

Après avoir créé ce rôle, vous devez joindre la politique AWSPartnerCentralSellingResourceSnapshotJobExecutionRolePolicyAWS gérée ou joindre la politique d'autorisation suivante :

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "partnercentral:CreateResourceSnapshot"
            ],
            "Resource": [
                "arn:aws:partnercentral:*::catalog/AWS/engagement/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "partnercentral:GetOpportunity"
            ],
            "Resource": [
                "arn:aws:partnercentral:*:111122223333:catalog/AWS/opportunity/*"
            ]
        }
    ]
}
```

Remplacez `{account}` par votre ID de compte AWS.

Après avoir créé le `ResourceSnapshotJobRole` rôle et attaché les autorisations, vous devez définir le rôle `ResourceSnapshotJobRole` pour votre organisation. Utilisez l'PutSellingSystemSettingsaction pour définir le rôle que vous avez créé `ResourceSnapshotJobRole` pour votre entreprise (identifié par votre AWS compte) :

```
aws-cli/2.13.5 Python/3.11.4 Linux/4.14.255-314-253.539.amzn2.x86_64
Host: partner-central.aws.amazon.com
Content-Type: application/json
```

```
{
  "ResourceSnapshotJobRoleIdentifier": "arn:aws:iam::{account}:role/
ResourceSnapshotJobRole"
}
```

`{account}`Remplacez-le par votre identifiant de AWS compte et `ResourceSnapshotJobRole` par le nom du rôle que vous avez créé. Ce rôle sera implicitement assumé par les jobs de capture des ressources afin d'accéder à vos opportunités et de créer des instantanés à partager avec d'autres partenaires dans le cadre de l'engagement.

Inviter les partenaires à saisir une opportunité

Les partenaires peuvent collaborer sur des opportunités émanant des deux partenaires et AWS en lançant une invitation à un engagement. Vous pouvez inviter un maximum de neuf partenaires à une opportunité.

Inviter des partenaires à participer à une opportunité

1. Suivez les étapes décrites dans la section Trouver des partenaires et entrer en contact avec eux dans le guide de vente AWS Partner Central. Vous devez entrer en contact avec des partenaires avant de pouvoir les inviter à participer à des opportunités. Cette connexion accorde un accès mutuel aux identifiants de compte de chacun, garantissant ainsi que les invitations parviennent au compte partenaire prévu.
2. Utilisez l'StartEngagementFromOpportunityTaskaction pour créer un engagement et y associer une opportunité. Cette action asynchrone exécute les tâches suivantes de manière séquentielle :
    1. Crée un engagement.
    2. Associe l'opportunité à l'engagement.
    3. Invoque l'CreateEngagementInvitationaction pour.AWS
    4. Soumet l'opportunité àAWS.
    5. Démarre le `ResourceSnapshotJob` pour créer des instantanés.
3. Invitez d'autres partenaires à se joindre à nous.
    a. Pour obtenir l'identifiant d'engagement associé à votre opportunité, utilisez l'ListEngagementFromOpportunityTasksaction, filtrée en fonction du `TaskIdentifier` résultat renvoyé à l'étape précédente.

b. Utilisez l'identifiant d'engagement et l'identifiant de compte du partenaire destinataire pour envoyer une invitation`CreateEngagementInvitation`.

Une invitation réussie envoie un événement créé dans le cadre d'une invitation d'engagement au partenaire destinataire.

Récupération des détails de l'invitation à un engagement

Lorsque vous recevez un événement créé par invitation d'engagement, utilisez l'GetEngagementInvitationaction pour récupérer les détails de l'invitation. La réponse inclut des informations essentielles sur l'opportunité client associée à l'engagement.

Vérifiez les détails de l'invitation, en particulier les `InvitationMessage` `Project.CustomerBusinessProblem` champs `Project.TitleProject.CustomerUseCase`,, et. Ces informations fournissent le contexte de l'opportunité client et des attentes du partenaire invitant en matière de collaboration.

Utilisez ces informations pour déterminer si vous souhaitez saisir l'opportunité en acceptant ou en refusant l'invitation d'engagement.

Répondre à une invitation de fiançailles

Lorsqu'un partenaire vous envoie une invitation d'engagement, il crée une invitation d'engagement créée par un événement qui vous informe de l'invitation. Vous avez la possibilité d'accepter ou de rejeter l'invitation. Cette décision détermine votre participation à l'opportunité multipartenaire.

Pour rejeter une invitation à un engagement :

Utilisez cette RejectEngagementInvitationaction pour rejeter l'invitation. Vous devez fournir un `RejectionReason` paramètre expliquant votre décision. Une fois rejetée, vous perdez l'accès aux détails de l'invitation, et un événement de rejet d'une invitation d'engagement indique au partenaire expéditeur que vous avez rejeté son invitation.

Pour accepter une invitation à un engagement :

Pour accepter l'invitation et passer à l'opportunité multipartenaire, utilisez l'StartEngagementByAcceptingInvitationTaskaction. Cette action asynchrone exécute les tâches suivantes de manière séquentielle :

1. Accepte l'invitation de fiançailles.

2. Crée une nouvelle opportunité dans votre compte partenaire en utilisant les données de l'opportunité du partenaire expéditeur. Vous recevrez un événement créé par une opportunité.
3. Inclut les informations supplémentaires requises pour identifier le client sur votre compte.
4. Publie un événement d'acceptation d'une invitation à un engagement qui indique au partenaire que vous avez accepté l'invitation et que vous avez été ajouté à l'engagement.

Invitation d'engagement expirée

Si vous ne répondez pas à une invitation d'engagement dans les quinze jours, celle-ci expire et vous ne pouvez pas participer à l'opportunité multipartenaire. Un événement d'expiration d'une invitation d'engagement indique au partenaire expéditeur que l'invitation a expiré.

> **Note**
>
> Si vous souhaitez toujours collaborer, le partenaire expéditeur doit lancer une nouvelle invitation d'engagement.

Examen et mise à jour des opportunités multipartenaires

Lorsqu'un partenaire initie un engagement sur une opportunité, le statut de révision de l'opportunité nouvellement créée sur le compte du partenaire destinataire est déterminé par le statut de l'opportunité du partenaire émetteur.

Opportunité avec statut d'évaluation soumise :

Si l'opportunité du partenaire d'envoi est `Lifecycle.ReviewStatus` définie sur`Submitted`, le processus suivant se déroule :

1. La nouvelle opportunité créée sur le compte du partenaire destinataire sera `Lifecycle.ReviewStatus` définie sur`Submitted`.
2. Le `Submitted` statut reste en vigueur jusqu'à ce que le partenaire expéditeur en ait l'opportunité`Approved`.
3. Une fois que l'opportunité du partenaire d'envoi est validée et qu'elle `Lifecycle.ReviewStatus` est définie sur`Approved`, les opportunités des autres partenaires hériteront automatiquement du `Approved` statut.

Ce processus garantit la cohérence des détails des opportunités entre les opportunités multipartenaires, éliminant ainsi le besoin de plusieurs évaluations.

Une fois terminé, un événement créé par une opportunité est déclenché avec l'ID d'opportunité correspondant. Les partenaires peuvent utiliser cet identifiant avec l'`GetOpportunity`action pour récupérer tous les détails de l'opportunité.

Opportunité avec statut de révision approuvé :

Si le partenaire expéditeur initie un engagement sur une opportunité `Lifecycle.ReviewStatus` définie sur`Approved`, le processus suivant se déroule :

1. La nouvelle opportunité créée sur le compte du partenaire destinataire sera `Lifecycle.ReviewStatus` définie sur`Approved`.
2. Un événement créé par une opportunité est déclenché avec l'ID d'opportunité correspondant.
3. Les partenaires peuvent utiliser l'GetOpportunityaction associée à l'identifiant de l'opportunité pour récupérer tous les détails de l'opportunité.

Cependant, l'opportunité approuvée peut ne pas contenir certaines informations spécifiques au partenaire qui n'ont pas été copiées de l'opportunité du partenaire expéditeur. Le partenaire destinataire doit mettre à jour ces informations à l'aide de l'UpdateOpportunityaction lorsque l'étape d'opportunité est mise à jour `Qualified` ou à une étape ultérieure :

- `Project.CustomerUseCase`
- `Project.DeliveryModels`
- `Solution`

- `PrimaryNeedsFromAws`
- `LifeCycle.TargetCloseDate`
- `Project.ExpectedCustomerSpend.Amount`
- `Project.ExpectedCustomerSpend.Currency`
- `Marketing.source`
- `Marketing.AwsFundingUsed`

Une fois l'opportunité créée dans le compte du partenaire destinataire, elle peut être gérée et mise à jour comme toute autre opportunité dans le système du partenaire. Les partenaires peuvent utiliser

l'UpdateOpportunityaction pour apporter des modifications ou fournir plus d'informations sur leur participation à l'opportunité.

Utiliser des instantanés pour recevoir les mises à jour des partenaires

Dans le cadre d'un engagement, les partenaires maintiennent et mettent à jour leurs opportunités de manière indépendante. Lorsqu'un utilisateur révise les ressources d'un engagement, par exemple en ajoutant une opportunité, un événement créé par un instantané des ressources d'engagement est publié.

Pour rester informé des modifications et accéder aux informations les plus récentes concernant les opportunités d'autres partenaires, vous pouvez utiliser les actions suivantes :

- ListEngagementResourceAssociations— Utilisez cette action pour récupérer l'identifiant d'engagement associé à l'opportunité.
- ListResourceSnapshots— Utilisez cette action pour récupérer une liste complète de toutes les opportunités associées à l'engagement, en fournissant une vue d'ensemble des instantanés disponibles pour tous les partenaires.
- GetResourceSnapshot— Utilisez cette action pour obtenir des résumés en temps réel d'instantanés d'opportunités spécifiques, ce qui vous permet de consulter les informations les plus récentes sans accéder directement à l'opportunité d'un autre partenaire.

Si vous identifiez des modifications ou des mises à jour pertinentes liées à l'opportunité d'un autre partenaire qui devraient être reflétées dans les vôtres, utilisez l'UpdateOpportunityaction. Cette action facilite l'intégration sélective de données pertinentes dans votre opportunité, garantissant ainsi l'alignement et la cohérence de l'engagement.

Afficher les membres de l'engagement

Un engagement sur une opportunité multipartenaire peut avoir jusqu'à dix partenaires. Chaque fois qu'un nouveau partenaire accepte l'invitation d'engagement, un événement d'ajout d'un membre à l'engagement est publié, informant tous les membres actuels du changement.

Pour voir tous les membres collaborant dans le cadre d'un engagement, procédez comme suit :

1. Utilisez l'ListEngagementResourceAssociationsaction pour récupérer l'ID d'engagement associé à l'opportunité.
2. Fournissez l'identifiant indiqué à l'étape 1 de l'ListEngagementMembersaction pour récupérer les coordonnées des partenaires des membres de l'engagement.

> **Note**
>
> Seuls les membres d'un engagement peuvent invoquer l'`ListEngagementMembers`action.

Surveillance de l'état des tâches liées aux captures d'écran

Lorsque vous travaillez avec des opportunités multipartenaires, vous devez créer un rôle qui vous permet de créer des instantanés de vos opportunités pour d'autres partenaires et de récupérer des instantanés d'opportunités d'autres partenaires dans le cadre d'un engagement. Sans ce rôle, les mises à jour que vous apportez à votre opportunité ne seront pas accessibles aux autres partenaires participant à l'engagement, qui continueront de voir des instantanés périmés de votre opportunité.

Pour suivre l'état des jobs de capture des ressources associés à une opportunité multipartenaire dans le cadre d'un engagement, procédez comme suit :

1. Utilisez l'ListEngagementResourceAssociationsaction pour récupérer l'ID d'engagement associé à l'opportunité.
2. Fournissez l'identifiant de l'étape 1 à l'ListResourceSnapshotJobsaction pour générer une liste de toutes les tâches instantanées détenues par l'appelant dans le cadre de l'engagement. Récupérez l'identifiant de la tâche dans la réponse.
3. Fournissez l'ID de tâche à l'GetResourceSnapshotJobaction pour suivre l'état de la tâche et vérifier si elle est en cours d'exécution.

L'`ResourceSnapshotJob`opération publie des métriques sur Amazon CloudWatch pour ses opérations asynchrones. CloudWatch traite les données en indicateurs lisibles en temps quasi réel pour vous aider à surveiller les performances et l'état du service.

Les mesures suivantes sont disponibles :

- Défauts — Compte les problèmes internes lors de l'exécution du travail. Utilisez cette métrique pour surveiller la stabilité du service et définir des alarmes pour les seuils de défaillance.
- Erreurs : suit les problèmes liés aux clients pendant le traitement des tâches. Cette métrique permet d'identifier les problèmes liés aux entrées ou aux configurations des clients.
- Invocations : représente le nombre total d'appels de tâches, y compris les tentatives réussies et infructueuses. Utilisez-le pour suivre les tendances d'utilisation du service au fil du temps.

> **Note**
>
> Les métriques ne sont communiquées CloudWatch qu'en cas d'activité dans le `ResourceSnapshotJob` service. S'il n'existe aucune tâche ou aucune donnée pour une métrique spécifique, cette métrique n'est pas signalée.

Pour garantir le bon déroulement de l'`ResourceSnapshotJob`opération :

- Utilisez les métriques pour vérifier que le service fonctionne comme prévu.
- Créez des CloudWatch alarmes pour surveiller les métriques et déclencher des actions (telles que l'envoi de notifications) lorsque les métriques dépassent les plages acceptables.
- Configurez une surveillance proactive pour identifier et résoudre les problèmes.

Pour plus d'informations sur l'utilisation CloudWatch des métriques, consultez le guide de CloudWatch l'utilisateur Amazon.

Associer, dissocier et attribuer des opportunités

Les opportunités peuvent être associées ou dissociées aux solutions,AWS produits,AWS Marketplace solutions, produits, offres et AWS Marketplace ensembles d'AWS Marketplace offres des partenaires à n'importe quel stade du cycle de vie des opportunités.AWS Marketplace

Associer des opportunités à d'autres entités

Les entités associées sont extraites du `GetOpportunity` procédé au sein de l'`RelatedEntityIdentifiers`objet. Ils `RelatedEntityIdentifiers` peuvent être mis à jour en utilisant`AssociateOpportunity`. Notez que ce champ ne peut pas être mis à jour à l'aide de la `CreateOpportunity` méthode `UpdateOpportunity` or.

Solutions

Avant de commencer un engagement à l'aide de l'`StartEngagementFromOpportunityTask`action, il est obligatoire d'associer au moins une et jusqu'à dix solutions partenaires.AWS Une AWS recommandation peut contenir ou non une solution partenaire.

Pour consulter vos solutions existantes, utilisez l'`ListSolutions`action.

Les partenaires peuvent créer, mettre à jour et gérer leurs solutions dans la `Build` section AWS Partner Central.

> **Important**
>
> Pour passer de l'étape d'opportunité à Engagé ou Lancé, vous devez associer une liste de solutions ou de AWS Marketplace produits valide à l'opportunité.

> **Note**
>
> Les anciennes solutions partenaires (`S-XXXX`identifiants) utilisant le type d'`Solution`entité sont toujours prises en charge. Cependant, nous vous recommandons d'associer le type d'`AwsMarketplaceSolution`entité aux opportunités, car le type d'`Solution`entité sera obsolète à l'avenir.

AWS produits

Jusqu'à 20 AWS produits peuvent être associés à une opportunité. Pour consulter la liste des AWS produits disponibles, utilisez la liste des AWS produits hébergés sur GitHub. L'association avec AWS les produits se fait exclusivement à l'aide de l'`AssociateOpportunity`action. Pour remplacer ou supprimer un produit, utilisez l'`DisassociateOpportunity`action.

Solutions

Les opportunités peuvent être associées à des solutions depuis votre compte vendeur. Utilisez l'`AssociateOpportunity`action avec `RelatedEntityType` set to`AwsMarketplaceSolutions`.

L'entité doit avoir le statut public ou limité. Le service rejette les entités provisoires ou inactives.

Pour associer une solution, un ARN est requis. L'exemple suivant montre le format ARN d'une solution :

```
arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/soln-
ab1c2de3fgh4i
```

Vous pouvez également associer les solutions d'un AWS Marketplace compte subsidiaire connecté à votre compte principal via Partner Account Connection. Entrez l'ARN complet, y compris l'ID du compte de la filiale.

L'`ListSolutions`action renvoie `AwsMarketplaceSolutionArn` un résumé de chaque solution et prend en charge le filtrage par ARN.

AWS Marketplace Produits

Des opportunités peuvent être associées àAWS Marketplace des produits depuis votre compte vendeur. Utilisez l'`AssociateOpportunity`action avec `RelatedEntityType` set to`AwsMarketplaceProducts`.

L'entité doit avoir le statut public ou limité. Le service rejette les entités provisoires ou inactives.

Pour associer un produit, un ARN est requis. L'ARN inclut le type de produit (par exemple`AmiProduct`,`SaaSProduct`,`ContainerProduct`). L'exemple suivant montre le format ARN d'un produit AMI :

```
arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/prod-
ab1c2de3fgh4i
```

Vous pouvez également associer des produits d'un AWS Marketplace compte subsidiaire connecté à votre compte principal par le biais de Partner Account Connection. Entrez l'ARN complet, y compris l'ID du compte de la filiale.

AWS Marketplace offres et ensembles d'offres

AWS Marketplace offres

Les opportunités peuvent être liées à une offre AWS Marketplace privée. Pour consulter les offres disponibles, utilisez l'API ListEntities from the AWS Marketplace Catalog. Actuellement, vous ne pouvez associer que des offres provenant du compte AWS Marketplace vendeur associé àAWS Partner Central.

Pour associer une offre privée, un ARN est requis. Échantillon :

```
arn:aws:aws-marketplace:us-east-1:123123123123:AWSMarketplace/Offer/offer-
dtn3example1tg
```

AWS Marketplace ensembles d'offres

Les opportunités peuvent également être associées à des ensembles AWS Marketplace d'offres. Les ensembles d'offres permettent de regrouper plusieurs offres de marché connexes pour un emballage

de solutions complet. Pour consulter les ensembles d'offres disponibles, utilisez l'API ListEntities from the AWS Marketplace Catalog.

Pour associer un ensemble d'offres, un ARN est requis. Échantillon :

```
arn:aws:aws-marketplace:us-east-1:123123123123:AWSMarketplace/OfferSet/offerset-
dtn3example1tg
```

Important : une opportunité peut être associée à une offre ou à un ensemble d'offres, mais pas aux deux. Un seul de ces types d'entités peut être associé à une opportunité à la fois.

Dissocier les opportunités des autres entités

Utilisez cette `DisassociateOpportunity` action pour dissocier l'opportunité des types d'entités suivants :

- Solutions
- AWS Produits
- AWS Marketplace Solutions
- AWS Marketplace Produits
- AWS Marketplace Offres
- AWS Marketplace Sets d'offres

En fonction de l'état de l'opportunité, différentes règles de validation s'appliquent lorsque vous dissociez ces objets associés.

Attribuer des opportunités

`AssignOpportunity`À utiliser pour changer le propriétaire de l'opportunité. Vous pouvez définir n'importe quel utilisateur de Partner Central comme propriétaire de l'opportunité.

Travailler avec vos prospects

Qu'est-ce qu'un lead ?

Dans le cadre des ventes interentreprises (B2B), un prospect représente un client potentiel qui a exprimé son intérêt pour les produits ou services d'une entreprise mais qui n'a pas encore été pleinement qualifié d'opportunité de vente. Les prospects constituent la première étape du pipeline

des ventes et doivent être entretenus et qualifiés avant de pouvoir être convertis en opportunités de covente active avec elles.AWS

Les prospects partagés par AWS les partenaires sont basés sur des signaux d'engagement des clients tels que la participation aux webinaires, la participation à des campagnes ou les demandes de solutions proposées par les partenaires. Ces prospects incluent des informations contextuelles précieuses telles que des informations sur l'entreprise cliente, les problèmes commerciaux et les détails de l'engagement pour aider les partenaires à hiérarchiser et à qualifier les opportunités potentielles.

Travailler avec des invitations à des prospects

Les partenaires reçoivent des invitations à des prospects AWS lorsque des clients potentiels expriment leur intérêt pour les solutions ou les services des partenaires. Le processus d'invitation de prospects permet aux partenaires d'évaluer les prospects avant de s'engager, garantissant ainsi une allocation efficace des ressources et des taux de conversion plus élevés.

Recevoir des invitations à des prospects

AWS crée des invitations à des prospects et les partage avec des partenaires via l'API de vente. Lorsqu'une nouvelle invitation à un prospect est disponible,AWS envoie une invitation à l'engagement avec le contexte du prospect aux partenaires éligibles.

Les partenaires doivent surveiller l'`Engagement Invitation Created`événement à l'aide d'Amazon EventBridge. Cette notification d'événement inclut le `payloadType` champ défini sur`LeadInvitation`, permettant aux partenaires de distinguer les invitations à des prospects des invitations à des opportunités. Dès réception de l'événement, les partenaires peuvent récupérer les détails de l'invitation pour évaluer le prospect.

Inviter des prospects

Les partenaires peuvent consulter toutes leurs invitations à des prospects à l'aide de l'action `ListEngagementInvitations` API. Cette action permet de récupérer les invitations dont le principal appelant est un expéditeur ou un destinataire.

Pour filtrer spécifiquement les invitations à des prospects, les partenaires doivent utiliser le `payloadType` filtre avec la valeur`LeadInvitation`. Ce filtre garantit que seules les invitations à des prospects sont renvoyées, à l'exclusion des invitations à des opportunités dans les résultats.

Les invitations à des prospects peuvent se présenter dans les états suivants :

- En attente : En attente de l'acceptation ou du rejet du partenaire
- Accepté : le partenaire a accepté l'invitation et a eu accès à tous les détails du prospect
- Rejeté : le partenaire a décliné l'invitation
- Expiré : l'invitation a dépassé sa date d'expiration sans action

Évaluation des invitations à des prospects

Avant d'accepter une invitation à un prospect, les partenaires doivent récupérer les informations détaillées de l'invitation à l'aide de l'action `GetEngagementInvitation` API. Cela fournit un contexte essentiel pour prendre des décisions d'acceptation ou de rejet éclairées.

La charge utile de l'invitation à un prospect inclut :

Informations sur le client (pré-acceptation disponible) :

- Nom de l'entreprise, secteur d'activité et pays
- URL du site Web
- Segment de marché (entreprise, grande, moyenne, petite, micro)
- AWS niveau de maturité (Évaluation, Single-Account, Multi-Account)

Interactions avec les clients :

- Type de source et informations sur la campagne
- Mesures prises par le client (remplissage du formulaire, participation au webinaire, etc.)
- Description du problème commercial fournie par le client
- Catégorie de cas d'utilisation
- Titre du contact (fournit le contexte de l'autorité pour la qualification BANT)

> **Important**
>
> Les coordonnées du client (nom, e-mail, numéro de téléphone) ne sont pas visibles au moment de l'invitation. Les coordonnées ne sont disponibles qu'après avoir accepté l'invitation, ce qui garantit la confidentialité des clients et prévient les comportements de lead farming.

Accepter les invitations à des prospects

Lorsqu'un partenaire décide de rechercher un prospect, il doit accepter l'invitation à l'aide de l'action `AcceptEngagementInvitation` API. Cette action ajoute le partenaire à l'engagement et donne accès à tous les détails du prospect, y compris les coordonnées du client.

Après acceptation :

- Le partenaire est ajouté en tant que membre de l'engagement
- Les coordonnées des clients deviennent visibles dans le contexte du responsable de l'engagement
- Le lead est classé comme partenaire qualifié (PQL) dans les systèmes AWS
- Le prospect apparaît dans la liste des prospects actifs du partenaire
- Les partenaires peuvent commencer à développer le prospect et à établir un contact avec le client

Les partenaires peuvent accepter plusieurs invitations de prospects de manière programmatique en appelant `AcceptEngagementInvitation` pour chaque invitation, ce qui permet un traitement groupé efficace de prospects de haute qualité.

Rejet des invitations à des prospects

Si un prospect ne correspond pas aux capacités, aux capacités ou à l'orientation commerciale du partenaire, les partenaires peuvent refuser l'invitation à l'aide de l'action `RejectEngagementInvitation` API. Lorsqu'ils rejettent une invitation de prospects, les partenaires doivent fournir un motif de rejet afin d'AWS améliorer le routage et la mise en correspondance des prospects. Les raisons de rejet les plus courantes sont les suivantes :

- Absence de couverture géographique
- Expertise technique insuffisante pour le cas d'utilisation
- Contraintes de capacité actuelles
- Inadéquation entre les clients et le secteur
- Désalignement budgétaire

Après le rejet :

- Le prospect est retiré de la file d'attente des invitations du partenaire
- Les coordonnées du client ne sont jamais partagées avec le partenaire

- Le prospect est classé comme un prospect rejeté par le partenaire (PRL) dans les systèmes AWS
- L'invitation reste visible dans la liste des invitations rejetées du partenaire à titre de référence

Les prospects rejetés peuvent être réaffectés à d'autres partenaires, à condition que les exigences en matière de consentement du client soient respectées.

Gestion des prospects acceptés

Après avoir accepté une invitation à un prospect, les partenaires peuvent accéder à des informations complètes sur le prospect, entretenir des relations avec les clients et suivre le prospect tout au long des étapes de qualification.

Afficher les détails du prospect

Les partenaires accèdent aux détails complets des prospects à l'aide de `GetEngagement` l'action API. L'engagement contient un contexte Lead contenant des informations complètes sur le client et ses interactions avec lui AWS.

Le contexte Lead inclut :

Profil client complet :

- Toutes les informations sur l'entreprise figurant dans l'invitation originale
- Coordonnées complètes pour toutes les interactions avec les clients (nom, e-mail, téléphone, titre de l'entreprise)
- AWS niveau de maturité et segment de marché

Historique des interactions :

- Plusieurs points de contact avec les clients regroupés en un seul engagement
- Chaque interaction inclut des informations sur la source, l'action du client, le problème commercial et les coordonnées
- Les interactions sont répertoriées par ordre chronologique afin de fournir une vue complète du parcours client

État de qualification :

- État actuel de la qualification du prospect (statut non qualifié, qualifié, disqualifié ou personnalisé)

- Les partenaires peuvent mettre à jour ce statut au fur et à mesure de leur processus de qualification

Cette vue complète permet aux partenaires de comprendre l'historique complet de l'engagement du client sur plusieurs AWS campagnes et points de contact, plutôt que de traiter chaque interaction comme un prospect distinct.

Répertorier les prospects actifs

Les partenaires peuvent récupérer tous leurs prospects actifs à l'aide de l'action `ListEngagements` API avec le `contextType` filtre défini sur`Lead`. Cela renvoie les engagements dont le partenaire est membre et l'engagement contient un contexte Lead.

La réponse de la liste inclut des informations clés telles que :

- Numéro d'engagement et titre
- Nom de l'entreprise du client
- Score d'engagement actuel
- Statut de qualification
- Nombre d'interactions
- Horodatages de création et de modification

Les partenaires peuvent utiliser cette liste pour créer des tableaux de bord, suivre l'état du pipeline de prospects et identifier les prospects nécessitant une attention particulière.

Enrichir les prospects grâce à des informations de prospection

Les partenaires peuvent enrichir les prospects acceptés grâce àAWS des informations leur permettant de mieux les hiérarchiser et de les qualifier avant d'investir dans la sensibilisation. L'enrichissement est effectué de manière asynchrone par le biais de tâches de prospection, qui renforcent l'engagement d'un prospect grâce à des signaux AWS dérivés pour étayer les décisions de qualification.

Démarrage d'une tâche de prospection

Les partenaires initient l'enrichissement à l'aide de `StartProspectingFromEngagementTask` l'action API. Cette action accepte jusqu'à 100 identifiants d'engagement dans une seule demande,

ce qui permet aux partenaires d'enrichir les prospects par lots. La tâche s'exécute de manière asynchrone et renvoie immédiatement un `TaskId` et `TaskArn` que les partenaires utilisent pour suivre la progression. L'initiale `TaskStatus` est l'une des valeurs `PENDING` suivantes : `IN_PROGRESSCOMPLETED`,, ou`FAILED`.

Les partenaires peuvent éventuellement fournir un `TaskName` pour étiqueter la tâche et un `ClientToken` pour garantir l'impuissance des nouvelles tentatives.

Sondage pour obtenir les résultats

La prospection étant asynchrone, les partenaires interrogent pour vérifier si elle est terminée à l'aide de l'action `GetProspectingFromEngagementTask` API, le résultat étant `TaskId` renvoyé au début. La réponse indique le statut global de la tâche et un résultat par engagement pour chaque identifiant soumis.

Chaque engagement est traité indépendamment, de sorte que les engagements individuels peuvent réussir ou échouer sans affecter les autres participants à la même tâche. Pour chaque engagement, le résultat inclut :

- Identifiant d'engagement : engagement qui a été traité.
- État : `PENDINGIN_PROGRESS`,`COMPLETED`, ou`FAILED`.
- Identifiant du contexte de prospection : renseigné lorsque l'engagement est traité avec succès. Cet identifiant fait référence au contexte de prospection enrichi ajouté à l'engagement et peut être utilisé lors d'opérations ultérieures.
- Code de raison et message : renseignés uniquement lorsqu'un engagement échoue, ils fournissent un code d'échec énuméré et une description lisible par l'homme avec des suggestions d'étapes de restauration.

Surveillance de plusieurs tâches

Pour suivre l'enrichissement de nombreux prospects, les partenaires utilisent l'action `ListProspectingFromEngagementTasks` API. Cette action répertorie toutes les tâches de prospection initiées par le compte de l'appelant et prend en charge les filtres facultatifs par identifiant de tâche, nom de tâche ou plage horaire de début, avec un tri configurable. La réponse est paginée ; utilisez la `NextToken` valeur de chaque réponse pour récupérer les pages suivantes.

Lorsque l'enrichissement est terminé avec succès, les AWS informations sont ajoutées à l'engagement en tant que contexte de prospection. Les partenaires peuvent récupérer les

informations enrichies `GetEngagement` et les utiliser pour définir le statut de qualification du prospect et décider des prospects à passer à la conversion.

Mise à jour des informations sur les

Au fur et à mesure que les partenaires gèrent les prospects et collectent des informations supplémentaires, ils peuvent mettre à jour les informations relatives aux prospects à l'aide de l'action `UpdateEngagementContext` API. Cette action permet aux partenaires de modifier le contexte du lead dans le cadre de l'engagement.

Les partenaires peuvent mettre à jour :

État de qualification : les partenaires doivent mettre à jour le statut de qualification au fur et à mesure de leur processus de qualification interne :

- Non qualifié : statut par défaut après acceptation d'un prospect ; indique que le prospect doit être évalué
- Qualifié : le prospect répond aux critères de qualification et présente un fort potentiel de conversion en opportunité
- Disqualifié : le prospect ne répond pas aux critères ou ne correspond pas aux solutions du partenaire
- États personnalisés : les partenaires peuvent définir leurs propres états de qualification en fonction de leurs processus internes

Métadonnées des prospects : les partenaires peuvent ajouter ou mettre à jour des informations supplémentaires relatives à leurs processus de qualification et de formation.

La mise à jour du statut de qualification aide les partenaires à suivre la progression des prospects, à générer des rapports de pipeline précis et à identifier les prospects prêts à être convertis en opportunités.

Contrôle AWS Mises à jour

AWS peut mettre à jour les informations relatives aux prospects en fonction de nouveaux signaux d'engagement client ou d'un score d'engagement affiné. Lorsque l'engagement est AWS mis à jour, les partenaires reçoivent un `Engagement Updated` événement via Amazon EventBridge.

Ces mises à jour peuvent inclure :

- Scores d'engagement affinés en fonction de l'activité des nouveaux clients
- Interactions supplémentaires avec les clients à partir de nouvelles campagnes ou de nouveaux points de contact
- Informations client mises à jour

Les partenaires doivent surveiller ces événements et appeler `GetEngagement` pour obtenir les dernières informations sur les prospects. Cela garantit que les partenaires disposent des informations les plus récentes pour prioriser et gérer les prospects.

L'`Engagement Updated`événement inclut le `contextTypes` terrain, ce qui permet aux partenaires de filtrer spécifiquement les prospects (engagements avec le contexte des prospects).

Convertir un prospect en opportunité

Lorsqu'un prospect a été qualifié et démontre une intention d'achat sérieuse, les partenaires peuvent en faire une opportunité de collaboration officielle en matière de covente avec AWS lui. Cette conversion marque le passage du lead nurturing (principalement piloté par les partenaires) à la gestion des opportunités (en collaboration avec).AWS

Approche recommandée : API pratique

Les partenaires peuvent rationaliser le processus de création et de liaison des opportunités en utilisant l'action `StartOpportunityFromEngagementTask` pratique de l'API. Cette API de tâches orchestre automatiquement plusieurs actions :

1. Crée une ébauche d'opportunité avec des informations préliminaires provenant du contexte principal
2. Lie l'opportunité à l'engagement de la source via un aperçu des ressources
3. Ajoute le CustomerProject contexte à l'engagement

Cette méthode pratique réduit le nombre d'appels d'API requis et garantit un suivi approprié de l'attribution. Les partenaires utilisant cette approche doivent tout de même :

- Complétez les détails de l'opportunité en utilisant `UpdateOpportunity`
- Solutions pour partenaires associés utilisant `AssociateOpportunity`
- Soumettez l'opportunité en utilisant une `StartEngagementFromOpportunityTask` fois que vous êtes prêt

L'API pratique est particulièrement utile pour les partenaires dotés de flux de travail automatisés menant à des opportunités qui souhaitent minimiser la complexité de l'intégration.

Approche alternative : Step-by-step API

Création d'un projet d'opportunité

La première étape de la conversion d'un prospect consiste à créer un brouillon d'opportunité à l'aide de l'action `CreateOpportunity` API. Cela crée une opportunité avec le `Lifecycle.ReviewStatus` set to`Pending Submission`. À ce stade, l'opportunité n'est pas encore soumise AWSà validation.

Les partenaires devraient compléter l'opportunité avec les informations recueillies lors du lead nurturing, notamment :

- Détails du compte client
- Informations sur le projet et problème commercial
- Dépenses attendues des clients
- Date de clôture cible
- Toute autre information pertinente collectée lors de la qualification

Le projet d'opportunité permet aux partenaires de préparer des informations complètes avant de les soumettre AWS, garantissant ainsi des taux d'approbation plus élevés et une validation plus rapide.

Lier l'opportunité à l'engagement des leaders

Après avoir créé le brouillon d'opportunité, les partenaires doivent le lier à l'engagement du prospect source à l'aide de l'action `CreateResourceSnapshot` API. Cette étape est essentielle pour :

- Suivi de l'attribution : établit la chaîne de provenance, de l'invitation à la conclusion de l'opportunité, afin de calculer avec précision le retour sur investissement pour les campagnes marketing et les sources de prospects.
- Mesures de conversion : Permet AWSà ses partenaires de mesurer les taux de conversion des prospects en opportunités et d'identifier les sources de prospects efficaces.
- Préservation du contexte : conserve l'historique complet du parcours client, y compris toutes les interactions originales et les signaux d'engagement.

Lors de la création de l'instantané des ressources, les partenaires spécifient :

- L'identifiant d'engagement (source : engagement du lead)
- L'identifiant de l'opportunité
- Type de ressource (Opportunité)

Cette action ajoute automatiquement un CustomerProject contexte à l'engagement, signalant que le prospect a été converti en opportunité active. L'engagement contient désormais à la fois le contexte principal d'origine (préservation de l'historique) et le nouveau CustomerProject contexte (indiquant une opportunité active).

Compléter les détails de l'opportunité

Une fois l'opportunité créée et liée, les partenaires doivent remplir les conditions supplémentaires relatives à l'opportunité avant de la soumettre. Consultez l'action de `AssociateOpportunity` l'API pour plus de détails.

Mise à jour des détails du projet : les partenaires peuvent utiliser l'action `UpdateOpportunity` API pour affiner les informations relatives au projet, les problèmes commerciaux des clients, les dépenses prévues et d'autres informations pertinentes recueillies lors de la qualification des prospects.

Soumission de l'opportunité

Une fois que toutes les informations requises sont complètes, les partenaires soumettent l'opportunité de AWS validation à l'aide de l'API `StartEngagementFromOpportunityTask` pratique. Cela fait passer l'opportunité du statut de brouillon au processus de AWS révision.

Exigence de validation : L'engagement doit avoir un CustomerProject contexte avant d'être soumis. Ce contexte est automatiquement créé lors de l'utilisation `CreateResourceSnapshot` pour lier l'opportunité à l'engagement. Si ce contexte est absent, la soumission échouera.

Après la soumission :

- Les `Lifecycle.ReviewStatus` modifications apportées par l'opportunité à `Submitted`
- Le prospect est classé dans la catégorie Partner Sales Qualified Lead (PSQL) dans les systèmes AWS
- AWS commence la validation pour s'assurer que les détails de l'opportunité sont exacts et complets
- Aucune modification ne peut être apportée à l'opportunité tant que le processus d'évaluation n'est pas terminé

L'opportunité suit ensuite le flux de validation standard décrit dans la documentation « Gérer vos opportunités », en passant par des états tels que Action requise In-Review, Approuvée ou Disqualifiée.

Utilisation des informations relatives au dimensionnement des transactions

Qu'est-ce que Deal Sizing ?

Le dimensionnement des offres est une fonctionnalité de la section Opportunités d'engagement client APN (ACE) de AWS Partner Central qui utilise l'IA pour fournir des estimations automatisées du montant des transactions et des recommandations de AWS service lorsque les partenaires créent ou mettent à jour des opportunités.

Estimations MRR prévisionnelles de l'IA

Les informations relatives au dimensionnement des transactions fournissent la médiane des fourchettes de revenus récurrents mensuels (MRR) prévues en fonction des AWS opportunités historiques, des cas d'utilisation et des descriptions commerciales des partenaires. Les partenaires doivent revoir et mettre à jour le MRR total au fur et à mesure que les partenaires recueillent plus d'informations sur la transaction et l'évolution du cycle de vente.

AI-Recommended AWS Produits

AI-recommended les produits sont identifiés en fonction de la description des problèmes commerciaux des clients et des détails des opportunités. L'IA recommande des AWS produits pertinents conformes aux exigences techniques et aux cas d'utilisation. Les partenaires doivent examiner ces suggestions et personnaliser la sélection de produits en fonction des besoins spécifiques du client.

Informations améliorées grâce aux URL du calculateur de prix

Les partenaires peuvent également importer AWS les URL du calculateur de prix pour renseigner automatiquement les sélections de services et recevoir des informations améliorées, notamment des indicateurs d'éligibilité au Migration Acceleration Program (MAP), des recommandations d'optimisation et une analyse des économies potentielles.

Comprendre les Source-Separated données

Deal Sizing fournit des informations provenant de deux sources distinctes pour vous donner une visibilité complète sur la valeur des opportunités :

- AWS Source : AI-recommended produits basés sur des modèles issus d'opportunités historiques similaires
- Source partenaire : Vos propres estimations importées depuis le calculateur de AWS prix

Gestion de la variation entre les sources

Lorsque les estimations fournies par les partenaires diffèrent considérablement des AWS recommandations, les partenaires doivent :

1. Passez en revue les détails de l'opportunité pour vous assurer que toutes les informations pertinentes sont saisies
2. Vérifiez que les configurations d'URL du calculateur de prix correspondent aux exigences réelles
3. Considérez AWS les recommandations comme des points de données basés sur des opportunités historiques similaires
4. Prenez des décisions éclairées en fonction du contexte et des exigences spécifiques du client

Accès aux informations sur le dimensionnement des transactions

Les partenaires accèdent aux données de dimensionnement des transactions par le biais de l'action `GetAwsOpportunitySummary` API, qui renvoie un `AwsOpportunitySummary` objet étendu contenant des prévisions relatives au dimensionnement des transactions, des recommandations de produits et des informations d'optimisation.

Quand les données de dimensionnement des transactions deviennent disponibles

Les informations sur le dimensionnement des transactions sont disponibles une fois qu'une opportunité est créée avec suffisamment de détails sur le client et le projet. Le système analyse les attributs de l'opportunité et génère :

1. MRR prévu par l'IA : calculé automatiquement en fonction des caractéristiques des opportunités
2. AWS recommandations de produits : services adaptés au problème commercial et aux exigences techniques du client
3. Informations améliorées (lorsque l'URL du calculateur de prix est fournie) : éligibilité au MAP, recommandations d'optimisation et analyse des économies de coûts

> **Note**
>
> Les informations sur le dimensionnement des offres ne sont pas disponibles pour toutes les opportunités. L'éligibilité dépend de facteurs tels que l'historique du partenaire et l'exhaustivité des informations fournies sur les opportunités.

Fournir des informations sur le dimensionnement des transactions

Les partenaires fournissent des informations pour le dimensionnement des transactions par le biais des actions `CreateOpportunity` et de `UpdateOpportunity` l'API. Le système utilise les attributs d'opportunité pour générer des recommandations.

Fournir la valeur totale du contrat (TCV)

Les partenaires qui estiment la valeur totale du contrat plutôt que le chiffre d'affaires récurrent mensuel peuvent indiquer directement la valeur de leur transaction. AWS convertit automatiquement le TCV en AWS MRR à l'aide de modèles de AI-powered conversion.

Pour le dimensionnement des offres dans TCV, indiquez `ExpectedContractDuration` as `Months` (les valeurs valides vont de 1 à 144 mois) et `TargetCompany` as `Self` et `Frequency` as. `None`

```
{
  "Project": {
    "ExpectedContractDuration": { "Term": "Months", "Value": "36" },
    "ExpectedCustomerSpend": [
      {
        "Amount": "1200000.00",
        "CurrencyCode": "USD",
        "Frequency": "None",
        "TargetCompany": "Self"
      }
    ]
  }
}
```

AWS calcule l'estimation du MRR d'AWS et renvoie les deux entrées sur `GetOpportunity` :

```
{
  "Project": {
```

```
    "ExpectedContractDuration": { "Term": "Months", "Value": "36" },
    "ExpectedCustomerSpend": [
      {
        "Amount": "5600.00",
        "CurrencyCode": "USD",
        "Frequency": "Monthly",
        "TargetCompany": "AWS"
      },
      {
        "Amount": "1200000.00",
        "CurrencyCode": "USD",
        "Frequency": "None",
        "TargetCompany": "Self"
      }
    ]
  }
}
```

Dans l'exemple ci-dessus, le `ExpectedCustomerSpend` premier tableau contient l'estimation AWS MRR (`TargetCompany: "AWS"`,`Frequency: "Monthly"`) et le second tableau représente la valeur totale du contrat du partenaire (`TargetCompany: "Self"`,`Frequency: "None"`).

Règles de validation du TCV

Toutes les erreurs de validation TCV renvoient un code d'erreur `INVALID_VALUE` avec`Reason: "BUSINESS_VALIDATION_FAILED"`. Le facteur distinctif est le `FieldName` et `Message` :

Condition	Champ	Message
Entrée automatique sans `ExpectedContractDuration`	`project.expectedContractDuration`	ExpectedContractDuration est obligatoire lorsqu'une entrée automatique est présente
`ExpectedContractDuration` sans saisie automatique (création uniquement)	`project.expectedCustomerSpend`	Une entrée automatique est requise lorsqu'elle ExpectedContractDuration est présente
Plusieurs entrées automatiques	`project.expectedCustomerSpend`	Une seule entrée automatique est autorisée

Condition	Champ	Message
Non valide TargetCompany lorsque la durée est présente	`project.expectedCustomerSpend.targetCompany`	TargetCompany doit être « AWS » ou « Self »
Incompatibilité de devise entre AWS et Self	`project.expectedCustomerSpend.currencyCode`	CurrencyCode doit correspondre entre les entrées
Entrée autonome + EstimationUrl ensemble	`project.expectedCustomerSpend.estimationUrl`	Entrée automatique et EstimationUrl ne peut pas coexister

> **Note**
>
> Activé`UpdateOpportunity`, s'il `ExpectedContractDuration` est présent sans saisie automatique mais que la valeur de l'offre partenaire existe dans le stockage, le système reconstruit automatiquement l'entrée automatique ; aucune erreur n'est générée.

Basculer entre le TCV et le MRR manuel

Pour passer du mode TCV au mode MRR manuel, configurez `ExpectedContractDuration` explicitement `null` et fournissez uniquement une `AWS` entrée :

```
{
  "Project": {
    "ExpectedContractDuration": null,
    "ExpectedCustomerSpend": [
      {
        "Amount": "8000.00",
        "CurrencyCode": "USD",
        "Frequency": "Monthly",
        "TargetCompany": "AWS"
      }
    ]
  }
}
```

> **Note**
>
> L'omission dans `ExpectedContractDuration` la charge utile (et non l'envoi du champ) signifie « aucun changement » : les données TCV existantes sont préservées. L'envoi explicite `null` signifie « effacer le mode TCV ».

URL du TCV et du calculateur de prix

Si un partenaire fournit à la fois une `Self` entrée (mode TCV) et un`EstimationUrl`, l'API renvoie une erreur de validation. Entrée automatique et EstimationUrl ne peut pas coexister dans la même demande.

Fournir des estimations MRR manuelles

Les partenaires peuvent saisir manuellement les estimations MRR à l'aide du `Project.ExpectedCustomerSpend[].Amount` champ :

> **Note**
>
> Le `CurrencyCode` champ accepte `USD` et`EUR`. Si la partition AWS est `aws-eusc` (AWS European Sovereign Cloud), le code de devise doit êtreEUR.

```
{
  "Project": {
    "ExpectedCustomerSpend": [
      {
        "Amount": "25000.00",
        "CurrencyCode": "USD",
        "Frequency": "Monthly",
        "TargetCompany": "AWS"
      }
    ]
  }
}
```

Ce champ est accessible à tous les partenaires et peut être utilisé pour accepter les prévisions de l'IA en définissant la même valeur ou pour remplacer les prévisions par des Partner-provided valeurs.

Fournir des URL de calculateur de prix

Les partenaires peuvent éventuellement fournir des URL de calculateur de AWS prix via le `Project.ExpectedCustomerSpend[].EstimationUrl` terrain pour importer automatiquement les configurations de service et recevoir des informations améliorées, notamment des indicateurs d'éligibilité au MAP, des recommandations d'optimisation et une analyse des économies de coûts.

```
{
  "Project": {
    "ExpectedCustomerSpend": [
      {
        "Amount": null,
        "CurrencyCode": "USD",
        "EstimationUrl": "https://calculator.aws/#/estimate?id=abc123",
        "Frequency": "Monthly",
        "TargetCompany": "AWS"
      }
    ]
  }
}
```

Lorsqu'une URL valide du calculateur de prix est fournie, le système calcule automatiquement le montant du MRR à partir de la configuration du calculateur et remplit à la fois le champ Montant et les données de dépenses détaillées au niveau du produit dans la source partenaire de. `AwsProductsSpendInsightsBySource` Les partenaires doivent définir le paramètre Amount sur null lorsqu'ils fournissent un EstimationUrl . Le système le calculera automatiquement.

Les URL dont le format n'est pas valide et les URL qui ne peuvent pas être résolues seront rejetées avec un. `ValidationException`

Comment quantifier et EstimationUrl travailler ensemble

Vous disposez d'une certaine flexibilité dans la manière dont vous fournissez des estimations de revenus récurrents mensuels (MRR) pour les opportunités. L'API gère intelligemment différentes combinaisons de montants manuels et d'URL du calculateur de AWS prix afin de garantir que vos opportunités puissent toujours être créées avec succès.

Utiliser uniquement un montant manuel

Lorsque vous fournissez uniquement le champ Montant, l'API enregistre votre estimation manuelle et la définit EstimationUrl sur null. Cette approche est utile lorsque vous avez une estimation du MRR spécifique à partir de conversations avec les clients ou de vos propres calculs.

Exemple de demande :

```
{
    "Project": {
        "ExpectedCustomerSpend": [{
            "Amount": "25000.00",
            "CurrencyCode": "USD",
            "Frequency": "Monthly",
            "TargetCompany": "AWS"
        }]
    }
}
```

En utilisant uniquement un AWS URL du calculateur de prix

Lorsque vous ne fournissez que le EstimationUrl champ :

- URL valide — L'API calcule le montant MRR à partir de la configuration de votre calculateur et enregistre à la fois l'URL et le montant calculé.
- URL non valide — L'API renvoie un ValidationException contenant des informations expliquant pourquoi l'URL n'est pas valide.

Exemple de demande :

```
{
    "Project": {
        "ExpectedCustomerSpend": [{
            "EstimationUrl": "https://calculator.aws/#/estimate?id=abc123",
            "CurrencyCode": "USD",
            "Frequency": "Monthly",
            "TargetCompany": "AWS"
        }]
    }
}
```

Exemple de réponse d'erreur (URL non valide) :

```
{
    "ErrorList": [{
        "Code": "INVALID_VALUE",
        "FieldName": "project.expectedCustomerSpend.estimationUrl",
```

```
        "Message": "Invalid Estimation URL: https://calculator.aws/#/estimate?
id=invalid"
    }],
    "Message": "The provided Estimation URL is invalid",
    "Reason": "INVALID_VALUE"
}
```

Fournir à la fois le montant et EstimationUrl

Lorsque vous fournissez les deux champs, l'API veille à ce que votre opportunité soit créée avec succès :

1. Si les deux valeurs sont inchangées, aucune mise à jour n'est effectuée sur ces champs
2. L'URL n'est pas valide — L'API enregistre le montant que vous avez fourni et le définit EstimationUrl sur null, ce qui permet de poursuivre la création de votre opportunité.
3. L'URL est valide et le montant calculé correspond au montant que vous avez fourni. Les deux valeurs sont enregistrées.
4. L'URL est valide mais le montant calculé ne correspond pas au montant que vous avez fourni — L'API enregistre le montant que vous avez fourni et le définit EstimationUrl sur null sans renvoyer d'erreur.

> **Note**
>
> Lorsque les deux valeurs ne correspondent pas aux valeurs existantes, l'API tente d'abord de calculer le montant à partir de l'URL. Le montant que vous avez fourni manuellement a toujours la priorité en cas de non-concordance ou de problème de validation.

Principal avantage : les URL non valides ne bloquent jamais la création d'opportunités

Cette approche garantit que les URL non valides ou inaccessibles du calculateur de AWS prix ne vous empêchent jamais de créer ou de mettre à jour des opportunités. Le montant que vous avez fourni manuellement a toujours la priorité en cas de conflit ou de problème de validation, ce qui vous permet de contrôler totalement les données relatives à vos opportunités.

Comprendre le modèle de AwsOpportunitySummary données étendu

Cette section décrit la structure de réponse renvoyée par l'action d'`GetAwsOpportunitySummary`API.

> **Note**
>
> AI-forecasted Le MRR est renvoyé `Project.ExpectedCustomerSpend[].Amount`

L'action de l'`GetAwsOpportunitySummary`API fournit des informations sur le dimensionnement des transactions via la nouvelle `Insights.AwsProductsSpendInsightsBySource` structure qui fournit une analyse complète des dépenses avec attribution de source.

Séparation des sources :AWS par rapport aux données des partenaires

Le dimensionnement des transactions sépare les informations par source afin de garantir la transparence et d'éviter le double comptage :

- AWS Source : Recommandations relatives aux produits d'IA provenant de AWS
- Source du partenaire : données sur les dépenses et détails des produits importés à partir des URL du calculateur de prix fournies par le partenaire

Cette séparation permet aux partenaires de :

1. Comparez leurs estimations aux AWS recommandations
2. Valider leurs hypothèses de dimensionnement
3. Évitez de gonfler les totaux en combinant accidentellement les deux sources
4. Maintenir la visibilité sur la provenance des données

Aperçu de la structure

```
{
  "Insights": {
    "AwsProductsSpendInsightsBySource": {
      "<AWS | Partner>": {
        "CurrencyCode": "string",
        "Frequency": "string",
        "TotalAmount": "string",
        "TotalOptimizedAmount": "string",
        "TotalPotentialSavingsAmount": "string",
        "TotalAmountByCategory": { },
        "AwsProducts": [ ]
```

```
      }
    }
  },
  "Project": {
    "ExpectedCustomerSpend": [ ]
  },
  "RelatedEntityIds": {
    "AwsProducts": [ ]
  }
}
```

Référence de champ

AwsProductsSpendInsightsBySource Carte

Source-separated des informations sur les dépenses qui fournissent une analyse indépendante pour les AWS recommandations et les estimations des partenaires.

Le `Insights.AwsProductsSpendInsightsBySource` champ est une carte contenant jusqu'à deux clés :

Clé	Type	Description
AWS	AwsProductsSpendInsights	AI-generated des informations, y compris les produits recommandés par AWS
Partenaire	AwsProductsSpendInsights	Partner-sourced informations dérivées des URL du calculateur de prix, y compris les coûts de service détaillés et les optimisations

> **Note**
>
> Seules les sources pour lesquelles des données sont disponibles sont incluses dans la réponse. Les nouvelles opportunités commencent généralement lorsque seule la AWS source est renseignée.

AwsProductInsights Objet

Analyse complète des dépenses pour une seule source (AWS ou partenaire), y compris les montants totaux, les économies d'optimisation, la ventilation par catégorie de programmes et des informations détaillées sur le niveau des produits.

Chaque objet source contient les champs suivants :

Champ	Type	Description
CurrencyCode	chaîne	Code de devise ISO 4217. Les valeurs prises en charge sont « USD » et « EUR ».
Frequency (Fréquence)	chaîne	Indique la fréquence à laquelle le client est censé dépenser le montant prévu. Seule la valeur Mensuel est autorisée pour le champ Fréquence, qui représente les dépenses mensuelles récurrentes.
TotalAmount	chaîne	Estimation totale des dépenses pour cette source avant les optimisations
TotalOptimizedAmount	chaîne	Estimation totale des dépenses après application des optimisations recommandées
TotalPotentialSavingsAmount	chaîne	Économies quantifiées réalisables grâce à la mise en œuvre d'optimisations
TotalAmountByCategory	objet	Montants des dépenses liés aux AWS programmes et aux voies de modernisation

Champ	Type	Description
AwsProducts	array	Product-level détails, y compris les coûts et les recommandations d'optimisation

Limite de l'état actuel : pour la AWS source TotalAmount, TotalOptimizedAmount, et TotalPotentialSavingsAmount peut être omis lorsque les recommandations de dépenses par produit ne sont pas disponibles. Les partenaires doivent faire référence `Project.ExpectedCustomerSpend[].Amount` à l'estimation totale du AWS MRR dans ces cas.

TotalAmountByCategory Objet

Les cartes dépensent des montants pour AWS les programmes et les initiatives stratégiques :

Clé de catégorie	Description
Éligible au MAP	Dépenses consacrées aux services éligibles au financement du Migration Acceleration Program
Éligibilité au CAT	Services de soutien à la répartition et au suivi des coûts
Pathway - Passez au Cloud Native	Services alignés sur la modernisation native du cloud
Pathway - Passez aux Managed Services	Services soutenant l'adoption des services gérés
Parcours - Passer à l'open source	Alternatives aux services open source
Voie - Passage aux conteneurs	Container-based options de service
Pathway - Passez au mode sans serveur	Services informatiques sans serveur
Parcours - Passage aux bases de données	Services de modernisation de bases de données

Clé de catégorie	Description
Parcours - Passez à l'analytique	Services d'analyse et de traitement des données

> **Note**
>
> Les montants totaux par catégorie peuvent être supérieurs TotalAmount car les produits individuels peuvent appartenir à plusieurs catégories.

AwsProducts Array

Liste des AWS services avec indicateurs d'éligibilité au programme (MAP, voies de modernisation), estimations de coûts et recommandations d'optimisation.

Chaque objet de produit contient :

Champ	Type	Obligatoire	Description
ProductCode	chaîne	Oui	AWS Identifiant du produit Partner Central utilisé pour l'association d'opportunités
ServiceCode	chaîne	Non	Code de service du calculateur de prix (liens vers l'URL d'origine du calculateur)
Catégories	array	Oui	Liste des catégories de programmes et de parcours auxquelles ce produit est éligible

Champ	Type	Obligatoire	Description
Amount	chaîne	Non	Coût de service de référence avant les optimisations (peut être nul pour AWS les recommandations provenant de sources)
OptimizedAmount	chaîne	Non	Coût du service après application des optimisations (peut être nul pour AWS les recommandations provenant de sources)
PotentialSavingsAmount	chaîne	Non	Service-specific réduction des coûts grâce à des optimisations (peut être nulle pour les AWS recommandations provenant de sources)
Optimisations	array	Non	Liste des recommandations d'optimisation spécifiques pour ce produit

Gestion nulle : pour les produits AWS d'origine, le montant et PotentialSavingsAmount les champs peuvent être nuls lorsque les recommandations de dépenses par produit ne sont pas disponibles. OptimizedAmount Le système fournit `Project.ExpectedCustomerSpend[0].Amount` plutôt un MRR total.

Objet d'optimisation

Chaque recommandation d'optimisation contient :

Champ	Type	Description
Description	chaîne	Human-readable explication de la stratégie d'optimisation
SavingsAmount	chaîne	Économies de coûts quantifiées réalisables grâce à la mise en œuvre de cette optimisation

Exemples

Exemple 1 :AWS recommandations uniquement (état actuel)

Cet exemple montre la réponse typique lorsque seules les recommandations de l'IA sont disponibles et que les montants de dépenses par produit ne peuvent pas encore être recommandés.

```
{
  "Catalog": "AWS",
  "Insights": {
    "AwsProductsSpendInsightsBySource": {
      "AWS": {
        "CurrencyCode": "USD",
        "Frequency": "Monthly",
        "AwsProducts": [
          {
            "ProductCode": "AmazonEC2",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging"],
            "Amount": null,
            "OptimizedAmount": null,
            "PotentialSavingsAmount": null,
            "Optimizations": []
          },
          {
            "ProductCode": "AWSLambda",
            "Categories": ["Cost Allocation Tagging", "Pathway - Move to Cloud
 Native"],
            "Amount": null,
            "OptimizedAmount": null,
            "PotentialSavingsAmount": null,
            "Optimizations": []
```

```
          },
          {
            "ProductCode": "AmazonRDS",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging", "Pathway - Move
 to Managed Services"],
            "Amount": null,
            "OptimizedAmount": null,
            "PotentialSavingsAmount": null,
            "Optimizations": []
          }
        ]
      }
    }
  },
  "Project": {
    "ExpectedCustomerSpend": [
      {
        "Amount": "28000.00",
        "CurrencyCode": "USD",
        "EstimationUrl": null,
        "Frequency": "Monthly",
        "TargetCompany": "AWS"
      }
    ]
  },
  "RelatedEntityIds": {
    "AwsProducts": [
      "AmazonEC2",
      "AWSLambda",
      "AmazonRDS"
    ]
  }
}
```


Points clés : la AWS source indique les produits recommandés avec des catégories, mais aucune quantité par produit. Estimation totale du MRR disponible via`Project.ExpectedCustomerSpend[].Amount`.

Exemple 2 : Importation du calculateur de prix pour les partenaires

Cet exemple montre des informations améliorées lorsqu'un partenaire a fourni une URL de calculateur de prix valide.

```
{
  "Catalog": "AWS",
  "Insights": {
    "AwsProductsSpendInsightsBySource": {
      "Partner": {
        "CurrencyCode": "USD",
        "Frequency": "Monthly",
        "TotalAmount": "32500.00",
        "TotalOptimizedAmount": "30000.00",
        "TotalPotentialSavingsAmount": "2500.00",
        "TotalAmountByCategory": {
          "MAP Eligible": "22500.00",
          "Cost Allocation Tagging": "32500.00",
          "Pathway - Move to Cloud Native": "5000.00",
          "Pathway - Move to Managed Services": "7500.00"
        },
        "AwsProducts": [
          {
            "ServiceCode": "ec2",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging"],
            "Amount": "15000.00",
            "OptimizedAmount": "13500.00",
            "PotentialSavingsAmount": "1500.00",
            "Optimizations": [
              {
                "Description": "Convert to 3-year reserved instances",
                "SavingsAmount": "1000.00"
              },
              {
                "Description": "Migrate to Graviton-based instances",
                "SavingsAmount": "500.00"
              }
            ]
          },
          {
            "ServiceCode": "lambda",
            "Categories": ["Cost Allocation Tagging", "Pathway - Move to Cloud
 Native"],
            "Amount": "5000.00",
            "OptimizedAmount": "5000.00",
            "PotentialSavingsAmount": "0.00",
            "Optimizations": []
          },
```

```
          {
            "ServiceCode": "AmazonRDS",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging", "Pathway - Move
 to Managed Services"],
            "Amount": "7500.00",
            "OptimizedAmount": "6500.00",
            "PotentialSavingsAmount": "1000.00",
            "Optimizations": [
              {
                "Description": "Optimize Multi-AZ for dev/test environments",
                "SavingsAmount": "1000.00"
              }
            ]
          },
          {
            "ServiceCode": "AmazonS3",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging"],
            "Amount": "5000.00",
            "OptimizedAmount": "5000.00",
            "PotentialSavingsAmount": "0.00",
            "Optimizations": []
          }
        ]
      }
    }
  },
  "Project": {
    "ExpectedCustomerSpend": [
      {
        "Amount": "32500.00",
        "CurrencyCode": "USD",
        "EstimationUrl": "https://calculator.aws/#/estimate?id=abc123",
        "Frequency": "Monthly",
        "TargetCompany": "AWS"
      }
    ]
  },
  "RelatedEntityIds": {
    "AwsProducts": []
  }
}
```

Points clés : La source du partenaire contient le détail complet des dépenses avec les montants par produit. Les recommandations d'optimisation fournissent des stratégies de réduction des coûts réalisables. La répartition par catégorie indique l'éligibilité au MAP (22 500$ sur un total de 32 500$). Les produits incluent ServiceCode un lien vers la configuration du calculateur de prix.

Exemple 3 : Présence des deux sources (scénario de comparaison)

Cet exemple montre comment les AWS recommandations et les estimations des partenaires apparaissent ensemble, ce qui permet de les comparer.


```
{
  "Catalog": "AWS",
  "Insights": {
    "AwsProductsSpendInsightsBySource": {
      "AWS": {
        "CurrencyCode": "USD",
        "Frequency": "Monthly",
        "AwsProducts": [
          {
            "ProductCode": "AmazonEC2",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging"],
            "Amount": null,
            "OptimizedAmount": null,
            "PotentialSavingsAmount": null,
            "Optimizations": []
          },
          {
            "ProductCode": "AWSLambda",
            "Categories": ["Cost Allocation Tagging", "Pathway - Move to Cloud
 Native"],
            "Amount": null,
            "OptimizedAmount": null,
            "PotentialSavingsAmount": null,
            "Optimizations": []
          },
          {
            "ProductCode": "EBS",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging"],
            "Amount": null,
            "OptimizedAmount": null,
            "PotentialSavingsAmount": null,
            "Optimizations": []
          },
```

```
          {
            "ProductCode": "RDS",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging", "Pathway - Move
to Managed Services"],
            "Amount": null,
            "OptimizedAmount": null,
            "PotentialSavingsAmount": null,
            "Optimizations": []
          }
        ]
      },
      "Partner": {
        "CurrencyCode": "USD",
        "Frequency": "Monthly",
        "TotalAmount": "32500.00",
        "TotalOptimizedAmount": "30000.00",
        "TotalPotentialSavingsAmount": "2500.00",
        "TotalAmountByCategory": {
          "MAP Eligible": "22500.00",
          "Cost Allocation Tagging": "32500.00",
          "Pathway - Move to Cloud Native": "5000.00",
          "Pathway - Move to Managed Services": "7500.00"
        },
        "AwsProducts": [
          {
            "ServiceCode": "ec2",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging"],
            "Amount": "15000.00",
            "OptimizedAmount": "13500.00",
            "PotentialSavingsAmount": "1500.00",
            "Optimizations": [
              {
                "Description": "Convert to 3-year reserved instances",
                "SavingsAmount": "1000.00"
              }
            ]
          },
          {
            "ServiceCode": "lambda",
            "Categories": ["Cost Allocation Tagging", "Pathway - Move to Cloud
Native"],
            "Amount": "5000.00",
            "OptimizedAmount": "5000.00",
            "PotentialSavingsAmount": "0.00",
```

```
            "Optimizations": []
          },
          {
            "ServiceCode": "amazonRDS",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging", "Pathway - Move
to Managed Services"],
            "Amount": "7500.00",
            "OptimizedAmount": "6500.00",
            "PotentialSavingsAmount": "1000.00",
            "Optimizations": [
              {
                "Description": "Optimize Multi-AZ for dev/test environments",
                "SavingsAmount": "1000.00"
              }
            ]
          },
          {
            "ServiceCode": "amazonS3",
            "Categories": ["MAP Eligible", "Cost Allocation Tagging"],
            "Amount": "5000.00",
            "OptimizedAmount": "5000.00",
            "PotentialSavingsAmount": "0.00",
            "Optimizations": []
          }
        ]
      }
    }
  },
  "Project": {
    "ExpectedCustomerSpend": [
      {
        "Amount": "28000.00",
        "CurrencyCode": "USD",
        "EstimationUrl": "https://calculator.aws/#/estimate?id=abc123",
        "Frequency": "Monthly",
        "TargetCompany": "AWS"
      }
    ]
  },
  "RelatedEntityIds": {
    "AwsProducts": [
      "AmazonEC2",
      "AWSLambda",
      "EBS",
```

```
      "RDS"
    ]
  }
}
```

Points clés : Les deux sources sont présentes :AWS recommandations (28 000$ au total) et calculateur pour les partenaires (32,5 000$ au total).AWS les recommandations indiquent 4 produits ; le calculateur des partenaires indique 4 produits (3 se chevauchant). Les partenaires peuvent comparer AWS les recommandations à leurs estimations détaillées.AWS total disponible via`Project.ExpectedCustomerSpend[].Amount`.

Bonnes pratiques

Gestion des valeurs nulles

1. EstimationUrl Champ : Attendez-vous EstimationUrl à une valeur nulle pour la plupart des partenaires. Il s'agit d'un comportement standard qui ne doit pas être considéré comme une erreur. Afficher les options de saisie manuelle du MRR comme chemin principal.
2. Catégories manquantes : si ce n' TotalAmountByCategory est pas le cas pour la AWS source, cela signifie que les recommandations par produit ne sont pas disponibles. Les ventilations par catégorie sont disponibles pour les Partner-sourced données lorsque les URL du calculateur de prix sont fournies.

Recommandations d'optimisation

1. Afficher des informations exploitables : mettez en évidence les descriptions des optimisations et les montants des économies réalisées pour aider les partenaires à comprendre les opportunités de réduction des coûts.
2. Économies cumulées : PotentialSavingsAmount additionnez les produits pour montrer le potentiel d'optimisation total.
3. Hiérarchiser par impact : triez les optimisations par ordre d'impact SavingsAmount pour aider les partenaires à se concentrer sur les recommandations ayant le plus d'impact.

Surveillance des mises à jour

1. Sondage périodique : Sondage `GetAwsOpportunitySummary` périodique (recommandé : toutes les 5 minutes) pendant les flux de création d'opportunités.

2. Gérer la génération asynchrone : les informations relatives au dimensionnement des transactions peuvent ne pas être immédiatement disponibles après la création de l'opportunité. Implémentez les états de chargement appropriés et une logique de nouvelle tentative.

Modèles d'intégration des API

1. Tirez parti des intégrations existantes : les partenaires qui appellent déjà `GetAwsOpportunitySummary` n'ont qu'à utiliser des champs supplémentaires dans la réponse, aucun nouvel appel d'API n'est requis.
2. Dégradation progressive : concevez des interfaces utilisateur pour gérer les scénarios dans lesquels les données de dimensionnement des transactions ne sont pas encore disponibles ou incomplètes.

Qualité et précision des données

1. Validez les URL du calculateur de prix : assurez-vous que les URL sont valides avant de les soumettre afin d'éviter les réponses EstimationUrl nulles.
2. Fournissez des informations détaillées sur les opportunités : des informations plus complètes sur les opportunités (détails du client, description du projet, exigences techniques) fournissent des recommandations d'IA plus précises.
3. Révision de l'écart : lorsque les estimations des partenaires diffèrent considérablement AWS des recommandations, passez en revue les détails des opportunités pour vous assurer que tout le contexte pertinent est saisi.

Bonnes pratiques

Réagir aux événements

Lorsque vous gérez des événements depuis l'API AWS Partner Central, assurez-vous que votre logique de traitement est idempotente pour gérer les événements dupliqués. Au lieu de passer des GetOpportunityappels immédiats pour chaque événement, pensez à regrouper ou à récupérer les détails de manière sélective en fonction des besoins de votre application. Pour des opérations ininterrompues, méfiez-vous des quotas.

Mettre en œuvre un verrouillage optimiste

Le verrouillage optimisé empêche le remplacement involontaire des données lors de mises à jour simultanées. Voici un scénario typique :

1. Le partenaire récupère une opportunité dans son système CRM.
2. L'utilisateur A met à jour l'opportunité sur AWS Partner Central.
3. L'utilisateur B met à jour la même opportunité en même temps grâce à l'intégration CRM.
4. Si les données changent, le système CRM tente de les télécharger mais renvoie un`ConflictException`.
5. L'utilisateur examine l'erreur et résout manuellement les données contradictoires.

Pour éviter ce scénario, toutes les UpdateOpportunitydemandes doivent inclure le `LastModifiedDate` paramètre, que vous pouvez obtenir à partir des informations précédentes CreateOpportunityUpdateOpportunity, et GetOpportunityles actions. La mise à jour ne réussit que si elle `LastModifiedDate` correspond à notre système. Si ce n'est pas le cas, vous devez récupérer la dernière version GetOpportunityet `LastModifiedDate` réessayer la mise à jour.

Synchronisation des données entre CRM et AWS Partner Central

Il est essentiel de synchroniser votre système avec les dernières données de Partner Central. Voici deux stratégies pour garantir que votre système reflète les données les plus récentes :

Utilisation d'événements (recommandé)

1. Chargez les données à l'aide de ListOpportunities.
2. Abonnez-vous à des événements d'opportunité.
3. Répondez aux nouvelles opportunités ou aux changements.
   - Lorsque vous recevez `Opportunity Created` ou que vous recevez `Engagement Invitation Accepted` des événements, utilisez `GetOpportunity` pour récupérer les dernières données. `Opportunity Updated`
   - Lorsque vous recevez `Engagement Invitation Rejected` des événements, supprimez les opportunités correspondantes.

Utilisation ListOpportunities du sondage

1. Chargez les données à l'aide de ListOpportunities.

2. Choisissez une fréquence d'interrogation, en veillant à ce qu'elle ne soit pas trop fréquente pour ne pas épuiser votre quota de lecture quotidien.
3. Identifiez les données les plus récentes `LastModifiedDate` de vos données stockées, en vous assurant qu'elles proviennent de AWS.
4. Utilisez l'horodatage dans le `AfterLastModifiedDate` filtre lorsque vous appelez. ListOpportunities

```
{
   "FilterList": [
       {
           "Name": "AfterLastModifiedDate",
           "ValueList": [ "2023-05-01T20:37:46Z" ] // Replace with actual timestamp
 of your last synced data
        }
   ]
}
```

5. AWS renverra les opportunités créées ou mises à jour après la valeur indiquée sur l'horodatage.
6. Effectuez une itération sur toutes les pages renvoyées en utilisant `NextToken` et mettez à jour les données de votre système à l'aide GetOpportunityde.

```
{
   "NextToken": "AAMA-EFRSN...PZa942D",
   "FilterList": [
       {
           "Name": "AfterLastModifiedDate",
           "ValueList": [ "2023-05-01T20:37:46Z" ] // Replace with actual timestamp
 of your last synced data
        }
   ]
}
```

Utilisation de AWS API Partner Central Benefits

L'API AWS Partner Central Benefits rationalise la manière dont les partenaires découvrent, demandent et gèrent les avantages dans le cadre de tous les programmes de AWS partenariat. En centralisant la gestion des avantages dans une interface unique et intuitive, le service crée une méritocratie transparente qui récompense les partenaires tout en simplifiant les opérations.

L'API fournit des fonctionnalités AWS d'API standard. Vous pouvez y accéder en utilisant les actions d'API ou en utilisant un AWS SDK adapté à votre langage de programmation ou à votre plateforme. Pour plus d'informations, consultez Getting Started with AWS and Tools to build on AWS.

Qu'est-ce qu'un avantage ?

Dans le réseau de AWS partenaires (APN), un avantage représente un avantage ou une capacité dont un partenaire peut tirer parti AWS pour soutenir la croissance de son entreprise et la réussite de ses clients. Les avantages sont conçus pour récompenser les partenaires en fonction de leurs qualifications et de leurs contributions, tout en apportant une valeur tangible qui aide les partenaires à développer leurs AWS pratiques.

Les avantages peuvent prendre de nombreuses formes, notamment :

- Incitatifs financiers - Crédits, décaissements en espèces et programmes de financement tels que Migration Acceleration Program (MAP) ou Marketing Development Funds (MDF)
- Avantages de l'accès - Accès en avant-première à de nouveaux services, outils ou ressources spécialisées
- Reconnaissance : badges numériques, certifications, désignations de compétences et statut au niveau du partenaire
- Ressources techniques - Assistance en matière d'architecture de solution, support technique et ProServe engagements
- Support marketing : Co-marketing opportunités, statut de partenaire vedette et publications sur les réussites

Découvrir les avantages disponibles

Les partenaires commencent leur parcours en découvrant les avantages dont ils disposent et en comprenant les conditions d'éligibilité. L'API Benefits fournit des fonctionnalités de découverte complètes via les API Benefit Discovery.

Liste des avantages disponibles

Les partenaires peuvent consulter tous les avantages disponibles à l'aide de l'action `ListBenefits` API. Cette action permet de récupérer un catalogue d'avantages avec des fonctionnalités de filtrage optionnelles pour aider les partenaires à trouver rapidement des opportunités pertinentes.

Pour découvrir efficacement les avantages, les partenaires peuvent filtrer par :

- Programme - Filtrez par programmes AWS partenaires spécifiques tels que MAP (Migration Acceleration Program), MDF (Marketing Development Funds), ISV Workload Migration, Innovation Sandbox, Proof of Concept ou Partner Initiative Funding
- Type d'expédition : filtrez selon la manière dont les avantages sont fournis : `CREDITS` `CASHDISCOUNT`,`ACCESS`,,`RECOGNITION`, ou `RESOURCE`
- État - Filtrer par statut de prestation : `ACTIVE` ou `DEPRECATED`

L'`ListBenefits`API renvoie des résumés des avantages contenant des informations essentielles telles que le numéro de l'avantage, le nom, la description, les programmes associés et les types d'expédition. Cette vue récapitulative permet aux partenaires d'analyser rapidement les avantages disponibles et d'identifier ceux qui correspondent le mieux à leurs objectifs commerciaux.

Exemple d'approche de filtrage :

Les partenaires axés sur les incitations financières peuvent filtrer par type d'exécution `CREDITS` et `CASH` pour voir les opportunités de financement. Les partenaires qui souhaitent bénéficier d'une assistance technique peuvent filtrer par type d'exécution `RESOURCE` ou `ACCESS` pour trouver des avantages liés au support technique et à l'accès aux outils.

Afficher les informations détaillées sur les avantages

Une fois qu'un partenaire a identifié un avantage qui l'intéresse, il peut récupérer tous les détails à l'aide de l'action de l'`GetBenefit`API. Cela fournit des informations complètes, notamment :

- Critères d'éligibilité - Exigences détaillées que les partenaires doivent remplir pour bénéficier de l'avantage
- Schéma de demande de prestations - Les informations spécifiques et la documentation que les partenaires doivent fournir lors de leur demande
- Détails de la subvention - Ce que les partenaires recevront une fois la subvention approuvée
- Programmes associés - Quels sont les programmes AWS partenaires pris en charge par la prestation

Le schéma de demande de prestations est particulièrement important car il définit la structure et les champs obligatoires pour les demandes de prestations. Les partenaires doivent examiner attentivement ce schéma avant de créer une demande de prestations afin de s'assurer qu'ils collectent toutes les informations nécessaires dès le départ.

> **Important**
>
> La détermination de l'éligibilité pour chaque avantage est gérée par le client ou par la couche d'interface utilisateur. Pour les avantages liés au financement, l'éligibilité est généralement basée sur les performances du partenaire et sur sa participation au programme.

Rubriques

- Utilisation des demandes de prestations
- Travailler avec les allocations d'avantages

Utilisation des demandes de prestations

Une demande d'avantages modélise la demande d'un partenaire pour un avantage spécifique. Il saisit toutes les informations nécessaires pour évaluer et traiter la demande en fonction des conditions spécifiques aux prestations. Le cycle de vie des demandes d'avantages sociaux passe par plusieurs étapes, de la création du brouillon à l'approbation finale ou au rejet.

Création de demandes de prestations

Les partenaires lancent le processus de demande d'avantages en créant une demande d'avantages à l'aide de l'action `CreateBenefitApplication` API. Au moment de sa création, l'application entre dans le `PENDING_SUBMISSION` statut, ce qui permet aux partenaires de préparer des informations complètes avant de les soumettre pour examen.

Lors de la création d'une demande de prestations, les partenaires doivent fournir :

- Identifiant du bénéfice : ID ou ARN du bénéfice demandé
- Types d'exécution - Comment le partenaire s'attend à bénéficier de l'avantage (`CREDITSCASH`,`DISCOUNT`,`ACCESS`,`RECOGNITION`, ou`RESOURCE`)
- Détails de la demande de prestations : document JSON contenant des informations spécifiques aux avantages, telles que définies par le schéma de demande de prestations
- Jeton client - Un jeton d'idempuissance unique pour éviter les soumissions dupliquées

Les partenaires peuvent éventuellement fournir :

- Nom et description : Human-readable identificateurs de l'application

- Contacts avec les partenaires - Coordonnées de la personne chargée de gérer cette demande d'avantages (maximum 1 contact)
- Pièces jointes - Documents justificatifs tels que les plans de projet, les SOW, les preuves des coûts ou les enquêtes de satisfaction client (maximum 10 fichiers)
- Ressources associées - Liens vers des opportunités connexes ou des allocations d'avantages existantes (maximum 10 ressources)
- Balises : Key-value paires pour l'organisation et le suivi des ressources (200 balises maximum)

L'`CreateBenefitApplication`API effectue une validation logicielle en vérifiant uniquement les types et modèles de champs de base. La validation complète de la logique métier a lieu lors de la soumission, ce qui permet aux partenaires de sauvegarder les demandes incomplètes et de les compléter ultérieurement.

Meilleure pratique : les partenaires doivent utiliser le schéma de demande d'avantages de `GetBenefit` pour comprendre exactement quelles informations sont requises avant de créer une demande. Cela réduit le risque d'échec de soumission en raison de données manquantes ou non valides.

Mise à jour des demandes de prestations

Les partenaires peuvent modifier les ébauches de demandes d'avantages à l'aide de `UpdateBenefitApplication` l'action API. Cela permet aux partenaires d'affiner les informations, d'ajouter de la documentation ou de corriger les erreurs avant de les soumettre.

Lors de la mise à jour d'une demande de prestations, les partenaires doivent fournir :

- Identifiant : ID ou ARN de la demande de prestations à mettre à jour
- Révision - Le numéro de révision actuel pour un verrouillage optimiste
- Détails de la demande de prestations - Le document JSON complet et mis à jour

Les partenaires doivent soumettre l'objet complet de la demande de prestations, même si seuls des champs spécifiques sont modifiés. La meilleure pratique consiste à récupérer d'abord les derniers détails de l'application en utilisant`GetBenefitApplication`, à modifier les champs nécessaires, puis à soumettre la charge utile complète mise à jour à`UpdateBenefitApplication`.

Verrouillage optimiste : le champ de révision garantit que les mises à jour ne sont appliquées que si l'application n'a pas changé depuis sa dernière récupération. Si la révision ne correspond pas à

la valeur actuelle de la base de données, la mise à jour est rejetée avec une erreur de conflit. Cela empêche les partenaires de remplacer accidentellement les modifications apportées par d'autres processus ou systèmes.

Les mises à jour ne peuvent être effectuées que lorsque l'application est en `PENDING_SUBMISSION` état. Une fois soumises, les candidatures entrent dans le processus de révision et ne peuvent plus être mises à jour via cette API.

Associer des ressources à des applications d'avantages

Les partenaires peuvent lier les demandes de prestations aux ressources connexes à l'aide de l'action `AssociateBenefitApplicationResource` API. Cela crée des liens précieux entre les avantages et le contexte commercial dans lequel ils sont utilisés.

Les types de ressources pris en charge incluent :

- OPPORTUNITÉ - Associez la demande d'avantages à une opportunité client spécifique dans. Cela est particulièrement utile pour les avantages spécifiques à une opportunité, tels que le financement MAP ou les crédits POC liés aux engagements des clients.
- BENEFIT_ALLOCATION - Lien vers les allocations d'avantages existantes, permettant aux partenaires d'enchaîner les avantages ou de démontrer comment les avantages antérieurs sont exploités

L'association de ressources ne peut avoir lieu qu'avant la soumission. Une fois qu'une demande de prestations est soumise (le statut change de`IN_REVIEW`), aucune autre association n'est autorisée. Les partenaires peuvent associer jusqu'à 10 ressources à chaque demande d'avantages.

Pour dissocier une ressource, les partenaires utilisent l'action `DisassociateBenefitApplicationResource` API. Tout comme l'association, la dissociation ne peut se produire qu'en cas de `PENDING_SUBMISSION` statut.

> **Important**
>
> Les partenaires doivent disposer des autorisations appropriées pour accéder à la demande d'avantages et lire la ressource spécifique associée. L'API valide ces autorisations lors de l'opération d'association.

Soumission de demandes de prestations

Lorsqu'une demande d'avantages est complète et prête à être AWS examinée, les partenaires la soumettent à l'aide de l'action `SubmitBenefitApplication` API. La soumission déclenche une transition d'état de `PENDING_SUBMISSION` à `IN_REVIEW` et lance le flux de travail d'approbation spécifique aux avantages.

Lors de la soumission, l'API effectue une validation complète, notamment :

- Validation des champs obligatoires : garantit la présence de tous les champs obligatoires dans les détails de la demande de prestations
- Validation du format des champs : vérifie que les valeurs des champs correspondent aux modèles et aux types de données attendus
- Validation des règles métier : applique une logique métier et des contraintes spécifiques aux avantages
- Validation des ressources : confirme que toutes les ressources associées sont valides et accessibles
- Validation des fichiers : vérifie que le traitement de toutes les pièces jointes a été effectué avec succès

Si la validation échoue, l'API renvoie un ValidationException code d'erreur détaillé et des messages indiquant les champs à corriger. Les partenaires doivent résoudre ces problèmes en utilisant `UpdateBenefitApplication` avant de tenter à nouveau de soumettre une demande.

Exigence de traitement des fichiers : les demandes de prestations ne peuvent pas être soumises si les pièces jointes sont toujours en cours `PENDING` d'élaboration. Les partenaires doivent attendre la fin du traitement du fichier (changement de statut`SUCCEEDED`) avant de le soumettre. Si le traitement des fichiers échoue (statut`FAILED`), les partenaires doivent télécharger les versions corrigées.

Une fois la soumission réussie :

- Le statut de la demande passe à `IN_REVIEW`
- L'équipe du bénéficiaire est informée de la nouvelle soumission
- Les partenaires ne peuvent plus modifier les détails des applications via les API de mise à jour standard
- L'application entre dans un flux de travail d'approbation défini qui peut inclure les étapes d'approbation commerciale, d'approbation technique et d'approbation financière

Les partenaires peuvent suivre l'avancement des soumissions en récupérant la demande et en surveillant le champ Étape, qui indique l'étape d'approbation actuelle (par exemple, « Approbation commerciale », « Approbation technique », « Approbation financière »).

Gestion des candidatures soumises

Une fois soumises, les partenaires disposent d'options limitées pour gérer les demandes de prestations, mais l'API fournit des actions spécifiques pour des scénarios courants.

Rappel des demandes de prestations

Si un partenaire découvre une erreur ou doit apporter des modifications après la soumission, il peut rappeler l'application à l'aide de l'action `RecallBenefitApplication` API. Recall fait revenir l'application à son `PENDING_SUBMISSION` statut, autorisant les mises à jour et la soumission à nouveau.

Lors du rappel d'une candidature, les partenaires doivent fournir :

- Identifiant : ID ou ARN de la demande de prestations à rappeler
- Motif - Une explication facultative du rappel (maximum de 1 000 caractères)

Le champ Motif permet une meilleure traçabilité et aide àAWS comprendre les problèmes courants à l'origine des rappels, ce qui oriente les améliorations futures du processus de demande de prestations.

Après le rappel, les partenaires peuvent l'utiliser `UpdateBenefitApplication` pour apporter les modifications nécessaires, puis les soumettre à nouveau `SubmitBenefitApplication` lorsqu'ils sont prêts.

> **Important**
>
> Le rappel n'est généralement disponible qu'au cours des premières étapes de révision. Les demandes qui sont passées aux étapes d'approbation ultérieures ou qui ont été approuvées peuvent ne pas être éligibles au rappel.

Modification des demandes de prestations

Pour les corrections mineures après la soumission, les partenaires peuvent utiliser l'action `AmendBenefitApplication` API pour mettre à jour des champs spécifiques sans rappeler

l'intégralité de l'application. Cela est particulièrement utile lorsque AWS les réviseurs demandent des clarifications ou des corrections au cours du processus de révision.

Les modifications utilisent une Patch-style approche JSON dans laquelle les partenaires spécifient :

- Path : expression JSONPath identifiant le champ à mettre à jour (par exemple,) `$.CreditDisbursementDetails.AwsAccountIdForCredits`
- Valeur : nouvelle valeur du champ
- Opération - L'opération à effectuer (actuellement seule `REPLACE` est prise en charge)

Les partenaires peuvent soumettre jusqu'à 10 modifications en un seul appel d'API. Chaque modification doit inclure un numéro de révision mis à jour pour un verrouillage optimal.

Les modifications doivent être utilisées pour des corrections mineures. En cas de modifications importantes, les partenaires doivent utiliser cette option `RecallBenefitApplication` pour remettre la demande à l'état de brouillon pour des mises à jour complètes.

Annulation des demandes de prestations

Si un partenaire n'a plus besoin d'un avantage ou souhaite retirer sa candidature, il peut l'annuler à l'aide de l'action `CancelBenefitApplication` API. L'annulation fait passer la demande au `CANCELED` statut, mettant définitivement fin au processus de candidature.

Lors de l'annulation d'une candidature, les partenaires doivent fournir :

- Identifiant : ID ou ARN de la demande de prestations à annuler
- Motif - Une explication facultative de l'annulation (maximum de 1 000 caractères)

Une fois annulées, les applications ne peuvent pas être réactivées. Si le partenaire décide par la suite qu'il a besoin de cette prestation, il doit créer une nouvelle demande de prestation.

Les raisons courantes d'annulation sont les suivantes :

- Une opportunité client a été perdue ou retardée
- Les contraintes de capacité des partenaires ont changé
- Les priorités commerciales ont changé
- L'avantage n'est plus nécessaire aux fins prévues

Afficher les détails de la demande de prestations

Les partenaires peuvent récupérer des informations complètes sur une demande d'avantages à l'aide de l'action `GetBenefitApplication` API. Cela fournit une vue complète de l'application, notamment :

Métadonnées de l'application :

- Identifiant unique (ID) et nom de ressource Amazon (ARN)
- Identifiant d'avantage associé
- Nom et description de l'application
- État actuel (`PENDING_SUBMISSIONIN_REVIEW`,`ACTION_REQUIRED`,`APPROVED`,`REJECTED`, ou`CANCELED`)
- Étape de traitement en cours

Informations sur le statut :

- Motif du statut : Human-readable explication du statut actuel
- Codes de motif du statut - Codes structurés indiquant des problèmes ou des exigences spécifiques (par exemple, « Liste de contrôle incomplète », « Plan de projet manquant », « Design Win manquant »)

Contenu de l'application :

- Document JSON complet sur les détails de la demande de prestations
- Informations de contact du partenaire
- Pièces jointes avec statut de traitement
- Ressources associées (opportunités ou allocations)
- Balises de ressources

Informations d'audit :

- Horodatage de création
- Horodatage de la dernière modification
- Numéro de révision actuel

Les codes de motif du statut sont particulièrement utiles lorsqu'une demande est en cours `ACTION_REQUIRED` d'examen. Ces codes fournissent des conseils spécifiques et exploitables sur les points que les partenaires doivent aborder pour que la demande soit traitée.

Liste des demandes de prestations

Les partenaires peuvent consulter toutes leurs demandes d'avantages à l'aide de l'action `ListBenefitApplications` API. Cela renvoie une liste paginée de résumés d'applications avec de puissantes fonctionnalités de filtrage.

Les partenaires peuvent filtrer les candidatures par :

- Programme - Afficher les applications pour des programmes spécifiques (MAP, MDF, Sandbox, POC, etc.)
- Type d'expédition - Filtrez par mode de livraison (`CREDITSCASH`,`ACCESS`,, etc.)
- Identifiant des avantages : consultez les demandes relatives à un avantage spécifique
- État - Filtrer par statut de l'application (`PENDING_SUBMISSIONIN_REVIEW`,`APPROVED`,`REJECTED`,`CANCELED`)
- Étape - Filtrer par étape d'approbation (approbation commerciale, approbation technique, approbation financière)
- ARN de ressources associées - Trouvez des applications liées à des opportunités ou à des allocations spécifiques

La réponse à la liste comprend des informations récapitulatives essentielles telles que :

- ID, ARN et nom de l'application
- Numéro de prestation associé
- Programmes et types d'expédition
- État actuel et stade
- Horodatages de création et de modification
- Ressources associées
- Numéro de révision actuel
- Champs sélectionnés dans les détails de la demande de prestations (tels que définis par le titulaire des avantages)

Les partenaires peuvent configurer le tri des résultats à l'aide du paramètre Sort, avec des options permettant de trier par date de création, statut, programme ou identifiant d'avantage par ordre croissant ou décroissant.

Création de tableaux de bord : L'`ListBenefitApplications`API est conçue pour faciliter la création de tableaux de bord pour les partenaires. En filtrant et en triant les applications, les partenaires peuvent créer des vues telles que :

- Demandes nécessitant une action (`ACTION_REQUIRED`statut)
- Candidatures récemment soumises (triées par date de création, `IN_REVIEW` statut)
- Demandes approuvées en attente d'attribution (`APPROVED`statut)
- Candidatures par programme (filtrées par programmes spécifiques)

Travailler avec les allocations d'avantages

Une allocation d'avantages représente un avantage qui a été accordé à un partenaire. Lorsqu'une demande d'avantages est approuvée, elle AWS crée une ou plusieurs allocations d'avantages qui fournissent aux partenaires les crédits, le financement, l'accès ou toute autre forme de réalisation des avantages réels.

Comprendre la répartition des avantages

Les allocations d'avantages peuvent représenter différents types de réalisation :

- Versements financiers (en espèces) - Paiements financiers directs aux partenaires avec suivi des décaissements
- AWS Crédits (CRÉDITS) - Codes de crédit applicables aux AWS comptes avec suivi de l'utilisation
- Allocations de consommables : montants Pre-approved de financement ou portefeuilles avec suivi de l'utilisation
- Subventions d'accès (ACCESS) - Accès aux systèmes, aux outils ou aux ressources
- Reconnaissance (RECONNAISSANCE) - Badges numériques, certifications ou désignations de statut
- Ressources (RESSOURCE) - Support technique, services consultatifs ou ProServe engagements

Le cycle de vie de chaque allocation d'avantages est suivi par le biais de son champ État :

- ACTIVE - L'allocation est disponible pour utilisation

- INACTIF - L'allocation est temporairement indisponible
- REMPLI - L'allocation a été entièrement utilisée ou livrée

Les allocations d'avantages incluent également des informations temporelles définissant leur date d'entrée en vigueur (StartsAt) et leur date d'expiration (ExpiresAt), permettant aux partenaires de comprendre les délais d'utilisation des avantages.

Liste des allocations d'avantages

Les partenaires peuvent consulter toutes leurs allocations d'avantages à l'aide de l'action `ListBenefitAllocations` API. Cela renvoie une liste paginée de résumés d'allocation avec des fonctionnalités de filtrage complètes.

Les partenaires peuvent filtrer les allocations par :

- Identifiant d'avantage - Afficher les allocations pour un avantage spécifique
- Identifiant de l'application Benefit - Afficher les allocations résultant d'une application spécifique
- Type d'expédition - Filtrez par type d'allocation (`CREDITSCASH`,`ACCESS`,, etc.)
- État - Filtrer par statut d'allocation (`ACTIVE`,`INACTIVE`,`FULFILLED`)

La réponse de la liste inclut des résumés d'allocation contenant :

- ID, ARN et nom d'allocation
- Numéro de prestation et numéro de demande de prestations associés (le cas échéant)
- Type d'expédition
- État actuel
- Horodatage de création
- Date de début

Cette vue de liste permet aux partenaires de créer des tableaux de bord de suivi des allocations et d'identifier :

- Allocations actives disponibles pour une utilisation immédiate
- Allocations approchant de l'expiration
- Allocations remplies pour le suivi historique

- Valeur d'allocation totale par programme ou type d'exécution

Afficher les informations d'allocation détaillées

Les partenaires peuvent récupérer les détails complets de l'allocation à l'aide de `GetBenefitAllocation` l'action API. Cela fournit des informations complètes sur l'allocation, y compris des détails spécifiques à l'expédition qui varient en fonction du type d'allocation.

Informations de base sur les allocations :

- Identifiant unique (ID) et nom de ressource Amazon (ARN)
- Nom et statut de l'allocation
- Motif du statut (explication du statut actuel)
- Numéro de prestation et numéro de demande de prestations associés
- Type d'expédition
- Horodatages de création, de mise à jour, de début et d'expiration

Détails de l'expédition (Type-Specific) :

Le FulfillmentDetail champ contient différentes structures d'information en fonction du type d'expédition :

Détails des décaissements en espèces (type en espèces)

Pour les paiements directs en espèces, les détails d'expédition incluent :

- Montant décaissé - Le montant total du financement versé, y compris le montant, la valeur et le code de devise
- Détails de l'émission (le cas échéant) - Informations sur les bons de commande ou les événements d'émission :
  - Numéro d'émission : identifiant unique pour le versement
  - Montant de l'émission - Montant en devise locale
  - Émis à - Horodatage du décaissement

Cette structure prend en charge à la fois les décaissements uniques et les scénarios de financement préapprouvés dans lesquels plusieurs émissions peuvent avoir lieu contre une seule allocation.

Détails du crédit (type de CRÉDITS)

Pour les AWS crédits, les détails d'expédition incluent :

- Montant alloué - Montant total du crédit alloué
- Montant émis - Montant total du crédit émis (peut être inférieur au montant alloué pour une émission progressive)
- Codes de crédit - Ensemble de détails relatifs aux codes de crédit individuels :
  - AWS Numéro de compte - Compte sur lequel le crédit est appliqué
  - Valeur - Valeur monétaire du code de crédit
  - AWS Code de crédit - La chaîne de code de crédit réelle
  - État - État du code de crédit (`ACTIVE`,`INACTIVE`,`FULFILLED`)
  - Émis à - Quand le code de crédit a été émis
  - Expire à - Quand le code de crédit expire

Ce suivi détaillé du crédit permet aux partenaires de surveiller l'utilisation du crédit sur plusieurs comptes et de comprendre quels crédits sont disponibles, partiellement utilisés ou entièrement utilisés.

Détails de l'allocation des consommables () Pre-Approved Amounts/Wallets

Pour les pools de financement préapprouvés, les détails d'exécution incluent :

- Montant alloué - Montant total de l'allocation
- Montant restant - Le montant non utilisé est toujours disponible
- Quantité utilisée - Quantité déjà consommée
- Détails de l'émission (le cas échéant) - Informations sur le bon de commande pour l'allocation

Les allocations consommables permettent aux partenaires de prélever des fonds selon les besoins pour les activités éligibles, avec un suivi en temps réel du solde restant.

Détails d'accès (type d'accès)

Pour les autorisations d'accès, les détails d'expédition incluent :

- Description - Explication détaillée de l'accès accordé et de la façon de l'utiliser

Les allocations d'accès peuvent donner accès à des services de prévisualisation, à des outils spécialisés ou à des ressources restreintes. Le champ de description fournit des instructions sur la manière dont les partenaires peuvent activer et utiliser l'accès accordé.

Appliquer les allocations aux nouvelles demandes de prestations

Le ApplicableBenefitIds champ relatif à l'allocation des avantages identifie les autres avantages qui peuvent tirer parti de cette allocation. Cela permet le chaînage des avantages grâce auquel les partenaires peuvent utiliser les allocations existantes pour demander des avantages supplémentaires.

Par exemple, un partenaire peut :

1. Recevez une allocation de financement du MAP préapprouvée
2. Créez des applications d'avantages pour les projets de migration de clients individuels
3. Associez ces demandes à l'allocation préapprouvée
4. Prélever des fonds sur l'allocation au fur et à mesure que les projets sont approuvés

Cette approche rationalise le processus de demande d'avantages pour les partenaires disposant d'un financement préapprouvé, réduisant ainsi le cycle d'examen des demandes de projets individuelles.

Utilisation de AWS API Partner Central Channel

AWS Les API Partner Central Channel vous permettent de gérer de manière programmatique votre activité de AWS revente autorisée. Ces API permettent aux fournisseurs de AWS solutions,AWS aux distributeurs et aux vendeurs AWS de distribution de :

- Signaler les comptes AWS de gestion utilisés pour la revente de programmes vers Partner Central
- Envoyer des invitations àAWS des comptes pour accepter des associations partenaires
- Créez les relations nécessaires pour intégrer les AWS comptes clients finaux revendus et appliquer des remises aux partenaires

Avec les API AWS Partner Central Channel, vous pouvez créer une automatisation personnalisée pour :

- Intégrez de nouveaux AWS comptes détenus par des partenaires aux programmes de revente APN

- Intégrez de nouveaux comptes clients finaux aux programmes de revente
- Accélérez l'intégration des clients et l'obtention de remises

Utilisation des comptes de gestion de programme (PMA)

Un compte de gestion de programme (PMA) associe votre compte de AWS gestion à l'inscription au programme de votre chaîne. Les comptes de gestion des programmes sont activés en libre-service à l'aide des poignées de main des chaînes. Lorsque vous créez et activez une PMA, les avantages et les remises du programme de chaîne sont appliqués à toutes les factures envoyées sur le compte de AWS gestion associé. Utilisez ces API pour signaler et gérer les AWS comptes qui gèrent la revente et les remises accordées aux chaînes :

- `CreateProgramManagementAccount`: Ajouter de nouveaux comptes pour la gestion de la facturation des clients
- `UpdateProgramManagementAccount`: Modifier les comptes existants
- `DeleteProgramManagementAccount`: Supprimer des comptes des programmes de revente
- `ListProgramManagementAccounts`: Afficher les comptes existants et leur statut

Travailler avec les poignées de main des chaînes

Les poignées de main des chaînes permettent un consentement mutuel sécurisé pour les opérations critiques de gestion des chaînes.AWS Partner Central utilise des poignées de main pour vérifier le consentement AWS des comptes de gestion à trois fins distinctes : activer les comptes de gestion des programmes (PMA), établir des périodes de service et mettre fin aux périodes de service de manière anticipée. Il en existe trois types :

- `PROGRAM_MANAGEMENT_ACCOUNT`: Les poignées de main du compte de gestion du programme vérifient le consentement lors de l'activation des PMA pour appliquer les avantages du programme de chaîne
- `START_SERVICE_PERIOD`: Les poignées de main relatives à la création de périodes de service établissent des accords mutuels pour les engagements de transfert de facturation avec des délais de préavis minimaux ou des durées fixes
- `REVOKE_SERVICE_PERIOD`: Les poignées de main relatives à la résiliation des périodes de service permettent de mettre fin de manière anticipée aux périodes de service actives lorsque les deux parties y consentent. Tous les types de poignée de main doivent être acceptés par le compte AWS de gestion cible pour prendre effet.

Utilisez ces API pour gérer les échanges de canaux en vue de l'activation de la PMA et de la gestion des périodes de service :

- `CreateChannelHandshake`: Créez des poignées de main pour les invitations PMA, l'établissement d'une période de service ou la fin d'une période de service
- `AcceptChannelHandshake`: Accepte les poignées de main provenant de AWS comptes cibles (peut être utilisée par les partenaires ou les clients selon le type de poignée de main)
- `RejectChannelHandshake`: Refusez les poignées de main provenant de comptes cibles AWS
- `CancelChannelHandshake`: annule les poignées de main en attente avant qu'elles ne soient acceptées, rejetées ou expirées
- `ListChannelHandshakes`: suivez et visualisez les poignées de main grâce au filtrage par type et par statut

Travailler avec les relations entre les canaux

Les relations vous permettent de gérer les clients finaux, les organisations internes ou les vendeurs en aval. Chaque relation inclut :

- Le compte de AWS gestion de l'AWS organisation associée
- Métadonnées du réseau de AWS partenaires requises pour la qualification des chaînes à prix discount

Lorsque vous créez une relation, tous les utilisateurs de l'AWS organisation associée bénéficient des avantages du Channel Program correspondants. Utilisez ces API pour signaler les vendeurs et les clients finaux :

- `CreateRelationship`: Intégrer de nouveaux comptes clients finaux, distributeurs-vendeurs ou internes AWS
- `GetRelationship`: Afficher les détails d'une relation spécifique
- `ListRelationships`: Afficher toutes les relations
- `UpdateRelationship`: Modifier les relations existantes
- `DeleteRelationship`: Supprimer les relations

Utilisation de l'API de mesure des recettes

Rubriques

- Utilisation de vos identifiants d'attribution de revenus
- Utilisation de vos parts de revenus sur le Marketplace

Utilisation de vos identifiants d'attribution de revenus

Un identifiant d'attribution de revenus est un identifiant au niveau de l'offre qui associe les revenus de vos produits AWS Marketplace à des opportunités spécifiques liées àAWS Marketplace Offers and/or AWS Partner Central. Il s'appuie sur la mise en œuvre du Partner Revenue Measurement (PRM) au niveau du produit : le PRM capture le chiffre d'affaires total attribué à un produit Marketplace, et un identifiant d'attribution des revenus associe ces revenus à des offres spécifiques en fonction des pourcentages de répartition des coûts mensuels que vous spécifiez.

Le service d'attribution des revenus expose des API pour créer, récupérer, répertorier et mettre à jour les identifiants d'attribution des revenus et pour gérer leurs entrées de répartition des coûts mensuels de manière asynchrone.

Qu'est-ce qu'un identifiant d'attribution de revenus ?

Un identifiant d'attribution de revenus comporte deux couches :

- L'attribution est une ressource nommée et décrite par le partenaire identifiée par un identifiant de `ra-` préfixe (par exemple,) `ra-aabbccddee001` et un ARN. Chaque attribution est limitée à un `Catalog` (`AWS`pour la production, `Sandbox` pour les tests) et au compte du partenaire.
- Une ou plusieurs entrées de répartition des coûts mensuelles. Chaque entrée associe une seule combinaison (numéro d'offre ou numéro d'opportunité, mois de facturation) à un pourcentage de répartition des coûts. Les entrées sont gérées par lots de manière asynchrone via un modèle de tâches d'allocation.

Un identifiant d'attribution de revenus peut être utilisé de deux manières :

- En tant que superposition au niveau de la transaction à votre implémentation PRM existante au niveau du produit. Créez un identifiant d'attribution des revenus, associez vos opportunités Marketplace Offers and/or ACE applicables et AWS associez les revenus de vos produits déjà mesurés à ces offres spécifiques. Aucune modification de vos balises de ressources ou de vos chaînes d'agent utilisateur existantes n'est requise.

- En tant qu'identifiant autonome utilisé directement comme valeur d'étiquette de ressource (`aws-apn-id=<RA ID>`) ou dans une chaîne d'agent utilisateur (`APN_1.1/pc_<RA ID>$`) pour attribuer la AWS consommation à une offre spécifique dès le départ.

Utilisation de vos identifiants d'attribution de revenus

Les partenaires peuvent gérer les identifiants d'attribution des recettes et leurs saisies mensuelles de répartition des coûts par le biais du service d'attribution des recettes. Le cycle de vie suit deux flux indépendants : le flux d'attribution et d'enregistrement (création, mise à jour, récupération, liste) et le flux d'allocations (démarrage d'une tâche par lots, interrogation des résultats, récupération des entrées individuelles, liste les entrées pour une attribution).

Création d'un identifiant d'attribution de revenus

La première étape consiste à créer un ID d'attribution de revenus à l'aide de l'action `CreateRevenueAttribution` API. Le `Id` et renvoyé `Arn` peut être utilisé immédiatement comme valeur de balise de ressource ou dans une chaîne d'agent utilisateur pour attribuer la consommation.AWS

Lors de la création d'un identifiant d'attribution des revenus, les partenaires doivent fournir :

- **`Catalog`**— `AWS` pour la production ou `Sandbox` pour les tests.
- **`Name`**— un nom lisible par l'homme pour l'attribution. Doit être unique dans le catalogue et dans le compte du partenaire. 128 caractères maximum.

Les partenaires peuvent éventuellement fournir :

- **`Description`**— description en texte libre de l'attribution. 1 024 caractères maximum.
- **`MarketplaceProduct`**— le produit AWS Marketplace à associer à cette attribution. Indiquez `ProductIdentifier` (l'identifiant du produit Marketplace à 25 caractères) et/ou `TenancyModel` (`MULTI_TENANT`ou`SINGLE_TENANT`). En cas d'omission lors de la création, l'attribution s'applique à toutes les consommations mesurées sous le Revenue Attribution ID, quel que soit le produit Marketplace acheté par le client, ce qui est utile lorsqu'un même déploiement couvre plusieurs offres Marketplace.
- **`Tags`**— jusqu'à 200 paires clé-valeur pour l'organisation des ressources. Les clés de balise doivent être uniques dans la demande.

Bonne pratique : indiquez `MarketplaceProduct.ProductIdentifier` chaque fois que votre offre est liée à une annonce Marketplace spécifique. Cela permet de AWS valider que le produit appartient au compte du partenaire appelant ou à un compte subsidiaire connecté via Partner Account Connection (PAC), et de faire apparaître le problème résolu `ProductCode` et `ProductType` (par exemple`SaaS`,`AMI`,,`ML`) dans la réponse. Si le produit n'appartient pas à un compte autorisé, l'API revient `ValidationException` avec raison`PRODUCT_NOT_FOUND_OR_NOT_OWNED`.

La réponse renvoie le nouveau `Id` (format`ra-[a-z0-9]{13}`)`Arn`, les `MarketplaceProduct` attributs résolus et le `Version` numéro initial (qui commence à 1 et augmente à chaque mise à jour suivante).

Ajouter des entrées de répartition des coûts mensuels

Une fois qu'un identifiant d'attribution de revenus est créé, les partenaires y associent AWS Marketplace Offers and/or ACE Opportunities en soumettant un lot de saisies mensuelles de répartition des coûts via l'action de l'`StartRevenueAttributionAllocationsTask`API. Il s'agit d'une opération asynchrone qui accepte jusqu'à 250 modifications d'allocation (`CREATE` and/or `UPDATE`) par tâche.

Chaque entrée d'allocation indique :

- ID d'offre ou ID d'opportunité : identifiant unique de l'offre associée. Si une offre Marketplace est déjà liée à une opportunité ACE, les partenaires n'ont qu'à fournir l'identifiant de l'offre ; les revenus AWS sont automatiquement attribués à l'opportunité associée.
- Mois de facturation : mois civil pour lequel la répartition des coûts s'applique (par exemple,`2026-04`).
- % de répartition des coûts : part de la AWS consommation totale du produit attribuable à cette offre au cours du mois de facturation en cours. Nécessaire pour les produits SaaS multi-locataires, y compris les composants hébergés par des partenaires pour les déploiements hybrides, où les clients partagent une infrastructure.
- Numéro de AWS compte client :AWS compte du client qui consomme le produit. Nécessaire pour les produits SaaS à locataires multiples.

Le pourcentage total de répartition des coûts entre toutes les entrées d'allocation pour un ID d'attribution de revenus donné et le même mois de facturation ne doit pas dépasser 100 %. Si le total dépasse 100 %, l'entrée incriminée est rejetée lors de la validation commerciale asynchrone avec un code d'erreur par enregistrement.

Validation synchrone : `StartRevenueAttributionAllocationsTask` effectue la validation de forme de base de manière synchrone (types de champs, modèles, taille du lot). Si la validation de base réussit, l'API renvoie un état `TaskId` avec`IN_PROGRESS`. La validation commerciale (contrôle des plafonds, règles d'immuabilité, recherches de dépendances par rapport à Marketplace et ACE) s'exécute de manière asynchrone et les résultats par enregistrement sont découverts via. `GetRevenueAttributionAllocationsTask`

Pour surveiller une tâche soumise, les partenaires interrogent à l'`GetRevenueAttributionAllocationsTask`aide de l'identifiant ou de l'ARN de la ressource d'attribution des revenus. La réponse passe de `IN_PROGRESS` à `COMPLETED` (indépendamment de la réussite ou de l'échec des enregistrements individuels) et renvoie :

- **`TaskStatus`**— `IN_PROGRESS``COMPLETED`, ou `FAILED` (échec de tâche de haut niveau).
- **`RecordResults`**— pour chaque enregistrement d'entrée : le statut attribué `AllocationId` (en cas de réussite), le statut par enregistrement (`SUCCEEDED`ou`FAILED`), et un code d'erreur structuré et un message en cas d'échec de l'enregistrement.

Les partenaires doivent récupérer rapidement les résultats des tâches. Une fois la tâche terminée, les résultats par enregistrement peuvent être rapprochés et toute entrée ayant échoué peut être corrigée et soumise à nouveau dans le cadre d'une nouvelle tâche.

Modification des entrées de répartition des coûts mensuelles

Les saisies de répartition des coûts sont gérées par mois de facturation selon les règles suivantes :

- Les partenaires peuvent à tout moment ajouter une nouvelle entrée mensuelle pour le mois de facturation en cours ou futur.
- Les partenaires peuvent à tout moment mettre à jour une saisie mensuelle pour le mois de facturation en cours ou futur.
- Les partenaires peuvent mettre à jour leur inscription du mois dernier jusqu'au 7 du mois en cours. Cette fenêtre permet aux partenaires de consulter l'utilisation réelle dans AWS Cost Explorer avant de finaliser l'allocation du mois précédent.

Après le 7 du mois en cours, l'attribution pour le mois de facturation précédent ne peut plus être modifiée. Les mises à jour de la saisie d'un mois en cours ou à venir s'appliquent à compter du prochain cycle de facturation mensuel. L'attribution historique pour les mois précédents n'est pas recalculée rétroactivement.

Les mises à jour des entrées d'allocation sont soumises par le biais de la même action d'`StartRevenueAttributionAllocationsTask`API, `Operation` définie sur `UPDATE` pour chaque entrée affectée. Le modèle de tâche asynchrone gère les mises à jour et les créations de manière uniforme.

Mettre à jour un identifiant d'attribution de revenus

Les partenaires peuvent mettre à jour un identifiant d'attribution `Description` de revenus existant à l'aide de l'action de l'`UpdateRevenueAttribution`API. L'enregistrement d'attribution lui-même est par ailleurs immuable — `NameMarketplaceProduct`, et les valeurs tag-on-create ne peuvent pas être modifiées après la création. Pour réétiqueter la ressource, utilisez les opérations de AWS balisage standard sur l'ARN de l'attribution.

Lors de la mise à jour d'un identifiant d'attribution de revenus, les partenaires doivent fournir :

- **`Catalog`**— `AWS` ou`Sandbox`.
- **`Identifier`**— l'`ra-`ID de l'attribution à mettre à jour.
- **`Version`**— la version actuelle de l'attribution pour un verrouillage optimiste.

Les partenaires peuvent éventuellement fournir :

- **`Description`**— la description en texte libre mise à jour. 1 024 caractères maximum.
- **`ClientToken`**— un jeton d'idempuissance pour la mise à jour.

Verrouillage optimiste : le `Version` champ garantit que les mises à jour ne s'appliquent que si l'attribution n'a pas changé depuis sa dernière extraction. Si le document soumis `Version` ne correspond pas à la version actuelle de la ressource, la mise à jour est rejetée avec `ConflictException` un message contenant les numéros de version soumis et actuel. La meilleure pratique consiste à récupérer la dernière version `GetRevenueAttribution` avant chaque mise à jour.

Afficher les détails de l'identifiant d'attribution des revenus

Les partenaires peuvent récupérer des informations complètes pour un identifiant d'attribution de revenus unique à l'aide de l'action `GetRevenueAttribution` API. Cela renvoie :

Métadonnées des ressources :

- Identifiant unique (`Id`) et nom de ressource Amazon (`Arn`).

- L'`Catalog`endroit dans lequel l'attribution existe.
- `Name` et `Description`.
- Le récupéré `Version` et le `LatestVersion` (utilisez le dernier pour les mises à jour ultérieures).

Attributs du produit Marketplace (s'`MarketplaceProduct`ils ont été définis lors de la création) :

- `ProductId`— l'identifiant du produit Marketplace fourni par le partenaire.
- `ProductCode`— le code produit AWS Marketplace résolu à partir de`ProductId`.
- `ProductType`—`SaaS`,`AMI`, `ContainerML`,`Data`, ou`Professional Services`.
- `TenancyModel`— `MULTI_TENANT` ou`SINGLE_TENANT`.

Informations d'audit :

- `CreatedDate` et `LastModifiedDate`.

Les partenaires peuvent éventuellement fournir un élément spécifique `Version` dans la demande pour récupérer une version historique de l'attribution. Omettez `Version` de renvoyer la dernière version.

Pour récupérer une entrée de répartition des coûts mensuelle spécifique, les partenaires utilisent l'action `GetRevenueAttributionAllocation` API associée à cette entrée`AllocationId`. Cela renvoie le numéro d'offre ou l'identifiant d'opportunité, le mois de facturation, le % de répartition des coûts, l'identifiant du AWS compte client, ainsi que le statut et les informations d'audit de l'entrée.

Identifiants d'attribution des revenus de la liste

Les partenaires peuvent consulter tous les identifiants d'attribution des revenus de leur compte à l'aide de l'action `ListRevenueAttributions` API. Cela renvoie une liste paginée de résumés d'attribution avec des fonctionnalités de filtrage et de tri.

Les partenaires peuvent filtrer les résultats par :

- **`Catalog`**— `AWS` ou `Sandbox` (obligatoire).
- **`Identifiers`**— une liste contenant jusqu'à 100 `ra-` identifiants spécifiques à récupérer.
- **`CreatedAfter`**/**`CreatedBefore`**— filtre par plage d'horodatage de création (inclus).

Les partenaires peuvent configurer le tri des résultats à l'aide du `Sort` paramètre :

- `SortBy`— `CreatedDate` ou`LastModifiedDate`.
- `SortOrder`— `ASCENDING` ou`DESCENDING`.

La réponse inclut un résumé pour chaque attribution :`Arn`,,`Id`, `CatalogName`, les `MarketplaceProduct` attributs résolus,`CreatedDate`,`LastModifiedDate`, les plus récents `Version` et `TotalRevenueAttributionAssociationCount` (le nombre d'entrées de répartition des coûts mensuelles actives liées à l'attribution).

Utilisez `MaxResults` (par défaut 25, maximum 100) et `NextToken` pour paginer dans de grands ensembles de résultats. La réponse indique `NextToken` si des pages supplémentaires sont disponibles.

Répertorier les entrées de répartition des coûts mensuelles

Les partenaires peuvent répertorier les entrées de répartition des coûts mensuelles pour un ID d'attribution de revenus spécifique à l'aide de l'action `ListRevenueAttributionAllocations` API. Cela renvoie une liste paginée de résumés d'allocation avec des fonctionnalités de filtrage.

Les partenaires peuvent filtrer les résultats par :

- **`MarketplaceOfferId`**— liste uniquement les entrées associées à une offre Marketplace spécifique.
- **`AwsPartnerCentralOpportunityId`**— liste uniquement les entrées associées à une opportunité Partner Central spécifique.
- **`BillingMonth`**— liste uniquement les entrées pour un mois calendaire spécifique.

La réponse inclut des informations récapitulatives pour chaque entrée : le numéro d'offre ou l'identifiant d'opportunité associé`AllocationId`, l'identifiant du AWS compte client, le mois de facturation, le % de répartition des coûts, le nom et le statut de l'entrée, ainsi que les horodatages d'audit.

Création de tableaux de bord : combinaison de `ListRevenueAttributions` et `ListRevenueAttributionAllocations` conçue pour faciliter la création de tableaux de bord pour les partenaires. Les partenaires peuvent développer des points de vue tels que :

- Tous les identifiants d'attribution des revenus créés au cours des 30 derniers jours, triés par date de création.
- Toutes les entrées de répartition des coûts mensuelles pour une offre Marketplace spécifique sur plusieurs identifiants d'attribution de revenus.
- Toutes les entrées de répartition des coûts pour le mois de facturation en cours, afin de valider que les totaux ne dépassent pas 100 % par attribution.
- Tous les identifiants d'attribution de revenus comportant au moins une entrée d'allocation active (`TotalRevenueAttributionAssociationCount > 0`).

Utilisation de vos parts de revenus sur le Marketplace

Une Marketplace Revenue Share est une entrée pour un produit AWS Marketplace qui déclare la part du chiffre d'affaires total du produit collecté via AWS Marketplace.AWS utilise cette entrée pour corréler les Marketplace-collected revenus d'un partenaire avec d'autres signaux de revenus.

Le Marketplace Revenue Share Service propose des API pour créer, récupérer et répertorier les ressources Marketplace Revenue Share, et pour gérer leurs allocations (les pourcentages de partage des revenus déclarés par les partenaires et les dates d'entrée en vigueur). Les opérations de AWS balisage standard sont prises en charge sur la ressource Marketplace Revenue Share.

Qu'est-ce qu'une Marketplace Revenue Share ?

Une Marketplace Revenue Share comporte deux niveaux :

- Le partage : entrée pour un seul produit AWS Marketplace, identifié par le Marketplace`ProductId`. Il existe exactement une Marketplace Revenue Share par produit.
- Une ou plusieurs allocations : chaque allocation déclare une `RevenueSharePercent` et une fenêtre de date d'entrée en vigueur (`EffectiveDate`facultativement`ExpirationDate`). Une action peut faire l'objet de plusieurs allocations au fil du temps, mais une seule `ACTIVE` allocation peut s'appliquer à tout moment.

Les allocations présentent les principales propriétés suivantes :

- Statut — `ACTIVE` ou`INACTIVE`. Lors de la création, le statut par défaut est. `ACTIVE` Seules `ACTIVE` les allocations participent au calcul des recettes et aux contrôles des chevauchements.

- Invariable de chevauchement : au sein d'une même action, aucune fourchette ne peut se chevaucher `[EffectiveDate, ExpirationDate]` entre deux `ACTIVE` allocations. Au plus une `ACTIVE` allocation illimitée (non`ExpirationDate`) est autorisée par action à tout moment.

Utilisation de vos parts de revenus sur le Marketplace

Les partenaires gèrent les Marketplace Revenue Shares et leurs allocations via le Marketplace Revenue Share Service. Le cycle de vie suit deux flux : le flux de partage (création, récupération, liste, balise) et le flux d'allocations (création, mise à jour, suppression logicielle, récupération, liste).

Création d'un partage des revenus sur le Marketplace

La première étape consiste à créer une part des revenus de la Marketplace pour un produit AWS Marketplace à l'aide de l'action `CreateMarketplaceRevenueShare` API. L'action est identifiée par le Marketplace `ProductId` et sert de conteneur pour toutes les allocations ultérieures.

Lors de la création d'une Marketplace Revenue Share, les partenaires doivent fournir :

- **`Catalog`**— `AWS` pour la production ou `Sandbox` pour les tests.
- **`ProductId`**— l'identifiant du produit Marketplace à 25 caractères. Motif :`[a-z0-9]{25}`.

Les partenaires peuvent éventuellement fournir :

- **`Tags`**— jusqu'à 50 paires clé-valeur pour l'organisation des ressources au moment de la création. `TagResource`À utiliser pour ajouter des balises après la création.
- **`ClientToken`**— un jeton d'idempuissance pour empêcher les créations dupliquées lors d'une nouvelle tentative. 64 caractères maximum. Les SDK génèrent automatiquement ce jeton.

Validation du produit : lors de la création, le service vérifie auprès de AWS Marketplace que le produit appartient au compte de l'appelant ou à un compte subsidiaire connecté à l'appelant via Partner Account Connection (PAC). Si le produit n'existe pas ou si le compte propriétaire n'est pas autorisé pour l'appelant, l'API revient `ValidationException` avec raison`PRODUCT_NOT_FOUND_OR_NOT_OWNED`.

Une action par produit : Un deuxième `CreateMarketplaceRevenueShare` appel pour obtenir le même `ProductId` résultat pour les mêmes `Catalog` retours`ConflictException`.

La réponse renvoie le partage `Arn` (format`arn:{partition}:partnercentral:{region}:{account-id}:catalog/{catalog}/marketplace-revenue-share/{productId}`), le `CatalogProductId`, le `Version` numéro initial (qui commence à 1 et augmente à chaque mutation d'allocation) et toutes les balises appliquées lors de la création.

Ajouter des allocations

Une fois qu'une part des revenus de la Marketplace est créée, les partenaires déclarent un pourcentage de part des revenus et une fenêtre de date d'entrée en vigueur en créant une allocation via l'action de l'`CreateMarketplaceRevenueShareAllocation`API.

Lors de la création d'une allocation, les partenaires doivent fournir :

- **`Catalog`**— `AWS` ou`Sandbox`.
- **`MarketplaceRevenueShareIdentifier`**— l'identifiant du partage parent.
- **`RevenueSharePercent`**— le pourcentage du chiffre d'affaires du produit collecté via Marketplace fourni sous forme décimale (par exemple, `0.45` pour 45 %).
- **`EffectiveDate`**— la date calendaire à laquelle cette allocation prend effet. Format :`YYYY-MM-DD`.

La réponse renvoie l'allocation `Id` (format`mrsa-[A-Za-z0-9]{13}`), le statut de l'allocation (`ACTIVE`lors de la création) et l'augmentation de l'action parent `Version` (un seul entier de version par action est augmenté à chaque mutation d'allocation, utilisé pour un verrouillage optimiste lors des mises à jour d'allocation suivantes).

Mettre à jour les allocations

Les partenaires peuvent mettre à jour une allocation à l'aide de `UpdateMarketplaceRevenueShareAllocation` l'action API. Les champs modifiables varient selon que l'allocation est future, actuellement effective ou dépassée.

Lors de la mise à jour d'une allocation, les partenaires doivent fournir :

- **`Catalog`**— `AWS` ou`Sandbox`.
- **`MarketplaceRevenueShareIdentifier`**— l'identifiant du partage parent.
- **`AllocationId`**— l'`mrsa-`ID de l'allocation à mettre à jour.

- **`MarketplaceRevenueShareVersion`**— la version actuelle de la part de la société mère pour un blocage optimiste de l'allocation. Les conflits liés aux bosses `ConflictException` réapparaissent.

Les partenaires peuvent éventuellement fournir :

- **`RevenueSharePercent`**— le pourcentage de part des recettes actualisé. Modifiable uniquement pour les allocations à date future.
- **`EffectiveDate`**— la date d'entrée en vigueur mise à jour. Modifiable uniquement pour les allocations à date future.
- **`ExpirationDate`**— la date d'expiration mise à jour. Modifiable sur les allocations à date future (n'importe quelle valeur> `EffectiveDate`) et sur les allocations actuellement en vigueur ou à durée indéterminée (n'importe quelle valeur). ≥ `today`
- **`Status`**— `ACTIVE` ou`INACTIVE`. Le réglage permet `INACTIVE` de supprimer progressivement l'allocation. Modifiable uniquement pour les allocations à date future.
- **`ClientToken`**— un jeton d'impuissance.

Afficher les détails du Marketplace Revenue Share

Les partenaires peuvent récupérer des informations complètes pour une seule Marketplace Revenue Share à l'aide de l'action `GetMarketplaceRevenueShare` API. Cela renvoie le partage`Arn`, `CatalogProductId`, `Version` (la version actuelle du partage parent), `CreatedDateLastModifiedDate`, et toutes les balises appliquées au partage.

Les partenaires peuvent récupérer une allocation unique à l'aide de l'action de l'`GetMarketplaceRevenueShareAllocation`API associée à l'allocation`Id`. Cela renvoie :

- L'allocation `Id` et la part des parents`MarketplaceRevenueShareArn`.
- `RevenueSharePercent`, `EffectiveDate`, `ExpirationDate`.
- `Status`(`ACTIVE`ou`INACTIVE`).
- La part du parent est `MarketplaceRevenueShareVersion` destinée à un verrouillage optimiste des mises à jour ultérieures.
- `CreatedDate` et `LastModifiedDate`.

Liste des parts de revenus de Marketplace

Les partenaires peuvent consulter toutes les Marketplace Revenue Shares détenues par leur compte à l'aide de l'action `ListMarketplaceRevenueShares` API. Cela renvoie une liste paginée des résumés des partages.

Les partenaires peuvent filtrer les résultats par :

- **`Catalog`**— `AWS` ou `Sandbox` (obligatoire).
- **`ProductIds`**— une liste d'identifiants de produits Marketplace spécifiques à récupérer.

La réponse inclut un résumé pour chaque partage :`Arn`,`Catalog`,`ProductId`, le plus récent `VersionCreatedDate`, et`LastModifiedDate`.

Utilisez `MaxResults` et `NextToken` pour paginer dans de grands ensembles de résultats.

Allocations d'annonces

Les partenaires peuvent répertorier les allocations pour un Marketplace Revenue Share spécifique à l'aide de l'action `ListMarketplaceRevenueShareAllocations` API. Cela renvoie une liste paginée de résumés d'allocation avec des fonctionnalités de filtrage.

Les partenaires peuvent filtrer les résultats par :

- **`Status`**— `ACTIVE` ou`INACTIVE`.
- Plage de dates d'entrée en vigueur —`EffectiveDateOnOrAfter`/`EffectiveDateOnOrBefore`pour récupérer les allocations dont la date d'entrée en vigueur se situe dans une fenêtre spécifique.

La réponse inclut des informations récapitulatives pour chaque allocation :`Id`,,,`RevenueSharePercent`,`EffectiveDate`, `ExpirationDate` `Status`MarketplaceRevenueShareVersion, et les horodatages d'audit.

Marquage des parts de revenus de Marketplace

Les opérations de AWS balisage standard sont prises en charge sur la ressource Marketplace Revenue Share :

- **`TagResource`**— ajoutez ou remplacez des tags sur un Marketplace Revenue Share. 50 tags maximum par partage.

- **`UntagResource`**— supprimez des tags spécifiques par touche.
- **`ListTagsForResource`**— récupérer tous les tags actuellement appliqués à une Marketplace Revenue Share.

Les allocations ne prennent pas en charge les balises. Marquez le partage parent pour organiser les allocations associées.

Chaque opération de balisage accepte le partage `Arn` (format`arn:{partition}:partnercentral:{region}:{account-id}:catalog/{catalog}/marketplace-revenue-share/{productId}`) comme identifiant de ressource.
Le AWS tag-on-create standard est également pris en charge via le champ sur`Tags`.
`CreateMarketplaceRevenueShare`

Pris en charge AWS Régions

Les sections suivantes fournissent des informations sur les régions AWS prises en charge.

Rubriques

- Pris en charge AWS régions pour AWS API du compte Partner Central
- Pris en charge AWS régions pour AWS API de vente centralisée pour les partenaires
- Pris en charge AWS régions pour AWS API Partner Central Benefits
- Pris en charge AWS régions pour AWS API Partner Central Channel
- Pris en charge AWS régions pour AWS API de mesure du chiffre d'affaires de Partner Central

Pris en charge AWS régions pour AWS API du compte Partner Central

Vous pouvez accéder au service de compte AWS Partner Central depuis n'importe quelle région AWS USA Est (Virginie du Nord) avec le point de terminaison suivant.

```
partnercentral-account.us-east-1.api.aws
```

Pris en charge AWS régions pour AWS API de vente centralisée pour les partenaires

Vous pouvez accéder au service de vente AWS Partner Central depuis n'importe quelle région AWS USA Est (Virginie du Nord) avec le point de terminaison suivant.

```
partnercentral-selling.us-east-1.api.aws
```

Pris en charge AWS régions pour AWS API Partner Central Benefits

Vous pouvez accéder au service AWS Partner Central Benefits depuis n'importe quelle région AWS USA Est (Virginie du Nord) avec le point de terminaison suivant.

```
partnercentral-benefits.us-east-1.api.aws
```

Pris en charge AWS régions pour AWS API Partner Central Channel

Vous pouvez accéder au service de gestion des canaux AWS Partner Central depuis n'importe quelle région commerciale AWS avec le point de terminaison suivant.

```
partnercentral-channel.global.api.aws
```

Signature de demandes avec SigV4A

Comme l'API AWS Partner Central Channel utilise un point de terminaison global, les demandes sont signées avec Signature Version 4a (SigV4a), une extension de SigV4 qui prend en charge la signature par cryptographie asymétrique, permettant ainsi des signatures vérifiables dans plusieurs régions. AWS

Utilisation du AWS SDK ou de la CLI

Si vous utilisez un AWS SDK ou la AWS CLI, la signature SigV4a est gérée automatiquement lorsque vous appelez le point de terminaison global. Aucune configuration supplémentaire n’est requise.

Signature manuelle des demandes

Si vous signez des demandes manuellement sans SDK, tenez compte des différences suivantes par rapport à la norme SigV4 :

- SigV4A déduit une paire de clés ECDSA (Elliptic Curve Digital Signature Algorithm) à partir de votre clé d'accès AWS secrète existante, plutôt que d'utiliser la dérivation de clés. HMAC-based
- L'étendue des informations d'identification est utilisée * pour le composant régional au lieu d'un nom de région spécifique, car la signature est valide dans toutes les régions.

Pour plus d'informations sur le fonctionnement de SIGv4a, consultez la version 4 d'AWS Signature pour les demandes d'API. Pour des exemples de code de signature Sigv4a, consultez le projet sigv4a-signing-examples sur. GitHub

Pris en charge AWS régions pour AWS API de mesure du chiffre d'affaires de Partner Central

Vous pouvez accéder au service AWS Partner Central Revenue Measurement depuis n'importe quelle région commerciale AWS avec le point de terminaison suivant.

```
partnercentral-prm.global.api.aws
```

Signature de demandes avec SigV4A

Comme le service de mesure des recettes utilise un point de terminaison mondial, les demandes sont signées avec Signature Version 4a (SigV4a), une extension de SigV4 qui prend en charge la signature par cryptographie asymétrique, permettant ainsi des signatures vérifiables dans plusieurs régions. AWS

Utilisation du AWS SDK ou de la CLI

Si vous utilisez un AWS SDK ou la AWS CLI, la signature SigV4a est gérée automatiquement lorsque vous appelez le point de terminaison global. Aucune configuration supplémentaire n’est requise.

Signature manuelle des demandes

Si vous signez des demandes manuellement sans SDK, tenez compte des différences suivantes par rapport à la norme SigV4 :

- SigV4A déduit une paire de clés ECDSA (Elliptic Curve Digital Signature Algorithm) à partir de votre clé d'accès AWS secrète existante, plutôt que d'utiliser la dérivation de clés. HMAC-based
- L'étendue des informations d'identification est utilisée * pour le composant régional au lieu d'un nom de région spécifique, car la signature est valide dans toutes les régions.

Pour plus d'informations sur le fonctionnement de SIGv4a, consultez la version 4 d'AWS Signature pour les demandes d'API. Pour des exemples de code de signature Sigv4a, consultez le projet sigv4a-signing-examples sur. GitHub

Permissions

Les sections suivantes fournissent des informations sur les autorisations.

Rubriques

- Accès à l'API du compte
- Accès à l'API de vente
- Accès à l'API Benefits
- Accès à l'API Channel
- Accès à l'API de mesure des recettes

Accès à l'API du compte

Le contrôle d'accès et les autorisations sont gérés par Gestion des identités et des accès AWS(IAM). Cette section fournit des conseils pour configurer les autorisations nécessaires pour interagir avec l'API du compte.

Conditions préalables

Avant de configurer les autorisations, assurez-vous Compte AWS que vous êtes lié à et que vous avez créé les rôles et utilisateurs IAM nécessaires. Pour de plus amples informations, veuillez consulter Configuration et authentification.

Utilisation AWS stratégies gérées

AWS fournit des politiques gérées qui accordent les autorisations requises pour interagir avec l'API du compte. Pour fournir l'accès nécessaire à la gestion des ressources du compte, associez la `AWSPartnerCentralFullAccess` politique à vos identités IAM. Pour plus d'informations, consultez la section Politiques AWS gérées pour les utilisateurs.

Affectation de politiques aux rôles et aux utilisateurs IAM

Procédez comme suit pour attribuer des politiques aux rôles et aux utilisateurs IAM :

1. Connectez-vous au AWS Management Console.
2. Accédez au service IAM.
3. Sélectionnez des rôles ou des utilisateurs, puis choisissez le rôle ou l'utilisateur IAM auquel vous souhaitez associer une politique.

4. Attachez la `AWSPartnerCentralFullAccess` politique au rôle ou à l'utilisateur IAM sélectionné.

Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

Gestion des autorisations à l'aide de clés de condition

Les clés de condition des politiques IAM fournissent des autorisations au niveau des ressources pour savoir quand appliquer les politiques de déclaration. Vous pouvez utiliser les clés de condition pour spécifier les conditions qui déterminent quand certaines autorisations sont autorisées ou refusées.

Pour plus d'informations, consultez Éléments de politique JSON IAM : opérateurs de condition.

Vue d'ensemble des clés de condition

Clé de condition	Description	Actions applicables	Valeurs valides
Partner Central : catalogue	filtre l'accès en fonction du type de l'entité de catalogue associée	toutes les actions	AWS, bac à sable

Résumé des autorisations requises

Résumé des autorisations requises

Action	Description
Partnercentral : AcceptConnectionInvitation	permet d'accepter les invitations de connexion
Partnercentral : AssociateAwsTrainingCertificationEmailDomain	permet d'associer des domaines d'e-mail de certification de AWS formation
Partnercentral : CancelConnection	permet d'annuler les connexions
Partnercentral : CancelConnectionInvitation	permet d'annuler les invitations de connexion
Partnercentral : CancelProfileUpdateTask	permet d'annuler les tâches de mise à jour du profil
Partnercentral : CreateConnectionInvitation	permet de créer des invitations de connexion

Action	Description
Partnercentral : CreatePartner	permet de créer des partenaires
Partnercentral : DisassociateAwsTrainingCertificationEmailDomain	permet de dissocier les domaines AWS de messagerie de certification de formation
Partnercentral : GetAllianceLeadContact	permet de récupérer les coordonnées du responsable de l'alliance
Partnercentral : GetConnection	permet de récupérer les détails de connexion
Partnercentral : GetConnectionInvitation	permet de récupérer les détails de l'invitation de connexion
Partnercentral : GetConnectionPreferences	permet de récupérer les préférences de connexion
Partnercentral : GetPartner	permet de récupérer les informations du partenaire
Partnercentral : GetProfileUpdateTask	permet de récupérer les détails des tâches de mise à jour du profil
Partnercentral : GetProfileVisibility	permet de récupérer les paramètres de visibilité du profil
Partnercentral : GetVerification	permet de récupérer les détails de vérification
Partnercentral : ListConnectionInvitations	permet de répertorier les invitations de connexion
Partnercentral : ListConnections	permet de répertorier les connexions
Partnercentral : ListPartners	permet de répertorier des partenaires
Partnercentral : PutAllianceLeadContact	permet de mettre à jour les coordonnées du responsable de l'alliance

Action	Description
Partnercentral : PutProfileVisibility	permet de mettre à jour les paramètres de visibilité du profil
Partnercentral : RejectConnectionInvitation	permet de rejeter les invitations de connexion
Partnercentral : SendEmailVerificationCode	permet d'envoyer des codes de vérification par e-mail
Partnercentral : StartProfileUpdateTask	permet de démarrer des tâches de mise à jour du profil
Partnercentral : StartVerification	permet de démarrer les processus de vérificat ion
Partnercentral : UpdateConnectionPreferences	permet de mettre à jour les préférences de connexion

Accès à l'API de vente

Le contrôle d'accès et les autorisations sont gérés par Gestion des identités et des accès AWS(IAM). Cette section fournit des conseils pour configurer les autorisations nécessaires pour interagir avec l'API, y compris les autorisations requises pour répertorier AWS Marketplace les entités.

Conditions préalables

Avant de configurer les autorisations, assurez-vous que vous Compte AWSêtes connecté à Partner Central et que vous avez créé les rôles et utilisateurs IAM nécessaires. Pour plus d'informations, consultez Configuration et authentification.

Utilisation AWS stratégies gérées

AWS fournit des politiques gérées qui accordent les autorisations requises pour interagir avec l'API. Pour fournir l'accès nécessaire à la gestion des opportunités, associez la `AWSPartnerCentralOpportunityManagement` politique à vos identités IAM. Pour plus d'informations, consultez les politiques AWS gérées pour les utilisateurs de AWS Partner Central.

AWSPartnerCentralOpportunityManagement politique

Cette politique donne un accès complet aux actions de gestion des opportunités de Partner Central.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "OpportunityManagement",
            "Effect": "Allow",
            "Action": [
                "partnercentral:AcceptEngagementInvitation",
                "partnercentral:AssignOpportunity",
                "partnercentral:AssociateOpportunity",
                "partnercentral:CreateEngagement",
                "partnercentral:CreateEngagementInvitation",
                "partnercentral:CreateOpportunity",
                "partnercentral:CreateResourceSnapshot",
                "partnercentral:CreateResourceSnapshotJob",
                "partnercentral:DeleteResourceSnapshotJob",
                "partnercentral:DisassociateOpportunity",
                "partnercentral:GetAwsOpportunitySummary",
                "partnercentral:GetEngagement",
                "partnercentral:GetEngagementInvitation",
                "partnercentral:GetOpportunity",
                "partnercentral:GetResourceSnapshot",
                "partnercentral:GetResourceSnapshotJob",
                "partnercentral:ListEngagementByAcceptingInvitationTasks",
                "partnercentral:ListEngagementFromOpportunityTasks",
                "partnercentral:ListEngagementInvitations",
                "partnercentral:ListEngagementMembers",
                "partnercentral:ListEngagementResourceAssociations",
                "partnercentral:ListEngagements",
                "partnercentral:ListOpportunities",
                "partnercentral:ListResourceSnapshotJobs",
                "partnercentral:ListResourceSnapshots",
                "partnercentral:ListSolutions",
                "partnercentral:RejectEngagementInvitation",
                "partnercentral:StartEngagementByAcceptingInvitationTask",
                "partnercentral:StartEngagementFromOpportunityTask",
                "partnercentral:StartResourceSnapshotJob",
```

```
                "partnercentral:StopResourceSnapshotJob",
                "partnercentral:SubmitOpportunity",
                "partnercentral:UpdateOpportunity"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ListingAWSMarketplaceEntities",
            "Effect": "Allow",
            "Action": ["aws-marketplace:ListEntities"],
            "Resource": "*"
        },
        {
            "Sid": "AWSMarketplaceOffersAccess",
            "Effect": "Allow",
            "Action": ["aws-marketplace:DescribeEntity"],
            "Resource": [
              "arn:aws:aws-marketplace:*:*:AWSMarketplace/Offer/*",
              "arn:aws:aws-marketplace:*:*:AWSMarketplace/OfferSet/*"
            ]
        }
    ]
}
```

Politiques personnalisées

Si les politiques gérées ne répondent pas à vos besoins, créez des politiques IAM personnalisées qui accordent les autorisations requises pour votre cas d'utilisation. L'exemple suivant est une politique personnalisée qui accorde des autorisations pour répertorier AWS Marketplace les entités :

Example Exemple de politique personnalisée

JSON

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

```
                "partnercentral:ListOpportunities",
                "aws-marketplace:ListEntities"
            ],
            "Resource": "*"
        }
    ]
}
```

Politique permissive personnalisée

Cette politique fournit un accès étendu aux actions de vente de Partner Central, y compris aux fonctionnalités qui pourraient être ajoutées à l'avenir sans nécessiter de mise à jour de la politique. En utilisant l'action jokerpartnercentral:*, cette politique donne automatiquement accès aux nouvelles fonctionnalités de vente de Partner Central dès qu'elles sont disponibles, réduisant ainsi le besoin de mises à jour manuelles. Cette politique inclut également des autorisations pour interagir avec les AWS Marketplace entités, ce qui permet de garantir le maintien de l'accès pour les actions de vente et pour les actions Marketplace.

JSON

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Action":
            [
                "partnercentral:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action":
            [
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity"
            ],
            "Resource": "*"
        },
```

```
        {
            "Effect": "Allow",
            "Action":
            [
                "aws-marketplace:SearchAgreements",
                "aws-marketplace:DescribeAgreement"
            ],
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "aws-marketplace:PartyType": "Proposer"
                }
            }
        }
    ]
}
```

Affectation de politiques aux rôles et aux utilisateurs IAM

Procédez comme suit pour attribuer des politiques aux rôles et aux utilisateurs IAM :

1. Connectez-vous au AWS Management Console.
2. Accédez au service IAM.
3. Sélectionnez des rôles ou des utilisateurs, puis choisissez le rôle ou l'utilisateur IAM auquel vous souhaitez associer une politique.
4. Attachez la `AWSPartnerCentralOpportunityManagement` politique ou votre politique personnalisée au rôle ou à l'utilisateur IAM sélectionné.

Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

Gestion des autorisations à l'aide de clés de condition

Les clés de condition des politiques IAM fournissent des autorisations au niveau des ressources pour savoir quand appliquer les politiques de déclaration. Vous pouvez utiliser les clés de condition pour spécifier les conditions qui déterminent quand certaines autorisations sont autorisées ou refusées.

Pour plus d'informations, consultez Éléments de politique JSON IAM : opérateurs de condition.

Vue d'ensemble des clés de condition

Clé de condition	Description	Actions applicables	Valeurs valides
Partner Central : catalogue	filtre l'accès en fonction du type de l'entité de catalogue associée	toutes les actions	AWS, bac à sable
AWS Marketplace : PartyType	filtre l'accès en fonction du type de partie (par exemple, proposant)	SearchAgreements, DescribeAgreement	proposant

Résumé des autorisations requises

Résumé des autorisations requises

Action	Description
Partnercentral : CreateOpportunity	permet de créer des opportunités
Partnercentral : UpdateOpportunity	permet de mettre à jour les opportunités
Partnercentral : ListOpportunities	permet de répertorier des opportunités
Partnercentral : GetOpportunity	permet de récupérer les détails de l'opportunité
Partnercentral : ListSolutions	permet de répertorier des solutions
Partnercentral : AssociateOpportunity	permet d'associer des opportunités à d'autres entités
Partnercentral : DisassociateOpportunity	permet de dissocier les opportunités des autres entités
Partnercentral : AcceptEngagementInvitation	permet d'accepter des invitations de fiançailles
Partnercentral : RejectEngagementInvitation	permet de rejeter les invitations à des fiançaill es

Action	Description
Partnercentral : GetEngagementInvitation	permet de récupérer les détails de l'invitation à un engagement
Partnercentral : ListEngagementInvitations	permet de répertorier les invitations à des engagements
Partnercentral : SubmitOpportunity	permet de soumettre des opportunités
Partnercentral : GetAwsOpportunitySummary	permet de récupérer le résumé des AWS opportunités
Partnercentral : StartProspectingFromEngagementTask	Crée une tâche de prospection à partir d'engagements
Partnercentral : GetProspectingFromEngagementTask	Renvoie les détails des tâches de prospection
Partnercentral : ListProspectingFromEngagementTasks	Renvoie une liste de tâches de prospection
AWS Marketplace : ListEntities	permet de répertorier AWS Marketplace les entités
AWS Marketplace : DescribeEntity	permet de décrire AWS Marketplace des entités
AWS Marketplace : SearchAgreements	permet de rechercher des accords dans AWS Marketplace
AWS Marketplace : DescribeAgreement	permet de décrire les accords dans AWS Marketplace

Accès à l'API Benefits

Le contrôle d'accès et les autorisations sont gérés par Gestion des identités et des accès AWS(IAM). Cette section fournit des conseils pour configurer les autorisations nécessaires pour interagir avec l'API Benefits.

Conditions préalables

Avant de configurer les autorisations, assurez-vous Compte AWS que vous êtes lié à et que vous avez créé les rôles et utilisateurs IAM nécessaires. Pour de plus amples informations, veuillez consulter Configuration et authentification.

Utilisation AWS stratégies gérées

AWS fournit des politiques gérées qui accordent les autorisations requises pour interagir avec l'API Benefits. Pour fournir l'accès nécessaire à la gestion des ressources liées aux avantages sociaux, associez la `AWSPartnerCentralFullAccess` politique à vos identités IAM. Pour plus d'informations, consultez la section Politiques AWS gérées pour les utilisateurs.

Affectation de politiques aux rôles et aux utilisateurs IAM

Procédez comme suit pour attribuer des politiques aux rôles et aux utilisateurs IAM :

1. Connectez-vous au AWS Management Console.
2. Accédez au service IAM.
3. Sélectionnez des rôles ou des utilisateurs, puis choisissez le rôle ou l'utilisateur IAM auquel vous souhaitez associer une politique.
4. Attachez la `AWSPartnerCentralFullAccess` politique au rôle ou à l'utilisateur IAM sélectionné.

Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

Gestion des autorisations à l'aide de clés de condition

Les clés de condition des politiques IAM fournissent des autorisations au niveau des ressources pour savoir quand appliquer les politiques de déclaration. Vous pouvez utiliser les clés de condition pour spécifier les conditions qui déterminent quand certaines autorisations sont autorisées ou refusées.

Pour plus d'informations, consultez Éléments de politique JSON IAM : opérateurs de condition.

Vue d'ensemble des clés de condition

Clé de condition	Description	Actions applicables	Valeurs valides
Partner Central : catalogue	filtre l'accès en fonction du type de	toutes les actions	AWS, bac à sable

Clé de condition	Description	Actions applicables	Valeurs valides
	l'entité de catalogue associée		

Récapitulatif des autorisations requises

Récapitulatif des autorisations requises

Action	Description
Partnercentral : AmendBenefitApplication	permet de modifier les demandes de prestations
Partnercentral : AssociateBenefitApplicationResource	permet d'associer des ressources à des applications d'avantages
Partnercentral : CancelBenefitApplication	permet d'annuler les demandes de prestations
Partnercentral : CreateBenefitApplication	permet de créer des demandes de prestations
Partnercentral : DisassociateBenefitApplicationResource	permet de dissocier les ressources des demandes de prestations
Partnercentral : GetBenefit	permet de récupérer le détail des prestations
Partnercentral : GetBenefitAllocation	permet de récupérer les détails de la répartition des avantages
Partnercentral : GetBenefitApplication	permet de récupérer les détails de la demande de prestations
Partnercentral : ListBenefitAllocations	permet de répertorier les allocations d'avantages
Partnercentral : ListBenefitApplications	permet de répertorier les demandes de prestations
Partnercentral : ListBenefits	permet de répertorier les avantages

Action	Description
Partnercentral : RecallBenefitApplication	permet de rappeler les demandes de prestatio ns
Partnercentral : SubmitBenefitApplication	permet de soumettre des demandes de prestations
Partnercentral : UpdateBenefitApplication	permet de mettre à jour les demandes de prestations

Accès à l'API Channel

Le contrôle d'accès et les autorisations sont gérés par Gestion des identités et des accès AWS(IAM). Cette section fournit des conseils pour configurer les autorisations nécessaires pour interagir avec l'API AWS Partner Central Channel.

Configuration du compte de gestion des chaînes

Les fonctionnalités de gestion des canaux sur AWS Partner Central nécessitent le déploiement des rôles IAM dans deux types de AWS comptes :

1. AWS Lien vers Partner Central Compte AWS :

   Ceci est Compte AWS associé à votre réseau de AWS partenaires et à votre compte Partner Central. Il n'existe qu'un seul AWS Partner Central connectéCompte AWS par partenaire. Dans le cadre d'une activité à configurer une fois, vous allez créer un rôle IAM àCompte AWS l'aide d'une politique gérée par les canaux Partner Central.
2. Compte de gestion du programme Compte AWS :

   Il est Compte AWS utilisé comme Bill-Transfer compte pour gérer la facturation et le paiement de la consommation du client final. Ceci Compte AWS est signalé comme un compte de gestion de programme dans la gestion des canaux AWS Partner Central. Vous pouvez avoir plusieurs comptes de gestion de programme et vous devrez créer un rôle IAM dans chacun des comptes que Compte AWS vous signalez en tant que compte de gestion de programme.

Pour chaque Compte AWS type, vous devrez associer différents rôles IAM à des politiques d'autorisation et de confiance spécifiques.

Accès pour le AWS Connecté à Partner Central Compte AWS

Dans le lien AWS Partner Central Compte AWS, créez un rôle IAM pour gérer les ressources du canal, qui sera accordé aux utilisateurs de Partner Central. Ce rôle IAM permet aux utilisateurs de consulter et de gérer les ressources de gestion des canaux sur AWS Partner Central. Le nom du rôle doit commencer par`PartnerCentralRoleFor`, et le nom recommandé est`PartnerCentralRoleForChannel`. Pour configurer le rôle, procédez comme suit :

- Ajoutez la politique de confiance personnalisée suivante pour permettre aux administrateurs cloud de AWS Partner Central et aux responsables d'alliance de mapper les rôles IAM aux utilisateurs de Partner Central :

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

- Associez la politique `AWSPartnerCentralChannelManagement`gérée au `PartnerCentralRoleForChannel` rôle.

Accès au compte de gestion du programme AWS compte (s)

Dans chaque compte de gestion de programme Compte AWS signalé dans AWS Partner Central, créez les rôles IAM suivants, avec les noms requis ci-dessous :

- `PartnerCentralChannelHandshakeApprovalManagement`: utilisez ce rôle pour gérer les poignées de main entre Partner Central et la direction de votre programme.Compte AWS Ce rôle IAM peut être utilisé pour répondre à l'activation de vos comptes de gestion de programme dans AWS Partner Central. Lors de la création de ce rôle, associez la politique `AWSPartnerCentralChannelHandshakeApprovalManagement` gérée.
- `PartnerCentralChannelBillingTransferReadOnly`: utilisez ce rôle multicompte pour avoir une visibilité sur l'état des transferts de facturation depuis AWS Partner Central. Ce rôle

communiquera le statut du transfert de facturation de votre part Compte AWSàAWS Partner Central afin de consulter de manière centralisée le statut du transfert de facturation. Si ce rôle est créé, les utilisateurs de AWS Partner Central peuvent consulter le statut du transfert de facturation dans AWS Partner Central. Toutefois, ce rôle n'autorise pas les utilisateurs de AWS Partner Central à accéder aux transferts de AWS facturation dans la console Billing and Cost Management.

- Stratégie d'approbation :
  - Dans le JSON ci-dessous, remplacez le par votre identifiant àCompte AWS 12 chiffres `{PartnerCentralLinkedAccountID}` associé àAWS Partner Central lors de la création du rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{PartnerCentralLinkedAccountId}:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Politique d'autorisation :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BillingTransferReadOnly",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeResponsibilityTransfer",
        "organizations:ListHandshakesForOrganization",
        "organizations:ListInboundResponsibilityTransfers",
        "organizations:ListOutboundResponsibilityTransfers"
    ],
      "Resource": "*"
    }
  ]
}
```

- `PartnerCentralChannelBillingTransferManagement`(facultatif) : utilisez ce rôle multicompte pour permettre la création et la gestion des transferts de facturation depuis AWS Partner Central. Ce rôle permettra aux utilisateurs de AWS Partner Central d'accéder au rôle IAM contenant une politique `AWSPartnerCentralChannelManagement` gérée pour gérer les transferts de facturation dans le compte Compte AWS de gestion du programme. Grâce à ce rôle multicompte, les utilisateurs de AWS Partner Central pourront accéder aux transferts de facturation et les gérer dans la console AWS Billing and Cost Management à l'aide du bouton « Manage in AWS Billing » dans la gestion des canaux AWS Partner Central.
  - Stratégie d'approbation :
    - Dans le JSON ci-dessous, remplacez le par votre identifiant àCompte AWS 12 chiffres `{PartnerCentralLinkedAccountID}` associé àAWS Partner Central lors de la création du rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{PartnerCentralLinkedAccountId}:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

  - Politique d'autorisation :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BillingTransferManagement",
      "Effect": "Allow",
      "Action": [
        "billingconductor:CreateBillingGroup",
        "billingconductor:ListPricingPlans",
        "organizations:AcceptHandshake",
        "organizations:CancelHandshake",
        "organizations:DeclineHandshake",
        "organizations:DescribeResponsibilityTransfer",
```

```
        "organizations:InviteOrganizationToTransferResponsibility",
        "organizations:ListHandshakesForAccount",
        "organizations:ListHandshakesForOrganization",
        "organizations:ListInboundResponsibilityTransfers",
        "organizations:ListOutboundResponsibilityTransfers",
        "organizations:TerminateResponsibilityTransfer",
        "organizations:UpdateResponsibilityTransfer"
    ],
      "Resource": "*"
    }
  ]
}
```

Utilisation AWS stratégies gérées

AWS fournit des politiques gérées qui accordent les autorisations requises pour interagir avec l'API Channel. Pour fournir l'accès nécessaire à la gestion de toutes les ressources du canal, associez la `AWSPartnerCentralChannelManagement` politique à vos identités IAM. Pour fournir l'accès nécessaire pour consulter et répondre aux demandes de poignée de contact des chaînes, associez la `AWSPartnerCentralChannelHandshakeApprovalManagement` politique à vos identités IAM. Pour plus d'informations, consultez les politiques AWS gérées pour les utilisateurs de AWS Partner Central.

AWSPartnerCentralChannelManagement politique

Cette politique accorde un accès complet aux actions de gestion des canaux Partner Central.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ChannelManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:CreateProgramManagementAccount",
        "partnercentral:UpdateProgramManagementAccount",
        "partnercentral:DeleteProgramManagementAccount",
        "partnercentral:ListProgramManagementAccounts",
        "partnercentral:GetProgramManagementAccount",
        "partnercentral:CreateRelationship",
        "partnercentral:UpdateRelationship",
```

```
        "partnercentral:DeleteRelationship",
        "partnercentral:GetRelationship",
        "partnercentral:ListRelationships",
        "partnercentral:CreateChannelHandshake",
        "partnercentral:AcceptChannelHandshake",
        "partnercentral:RejectChannelHandshake",
        "partnercentral:CancelChannelHandshake",
        "partnercentral:ListChannelHandshakes"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "AWS",
            "Sandbox"
          ]
        }
      }
    },
    {
      "Sid": "ChannelBillingTransferRoleAccess",
      "Effect": "Allow",
      "Action": [
        "sts:AssumeRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/PartnerCentralChannelBillingTransferManagement",
        "arn:aws:iam::*:role/PartnerCentralChannelBillingTransferReadOnly"
      ]
    },
    {
      "Sid": "TaggingAccess",
      "Effect": "Allow",
      "Action": [
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
      ],
      "Resource": [
        "arn:aws:partnercentral:*:*:catalog/*/program-management-account/*",
        "arn:aws:partnercentral:*:*:catalog/*/channel-handshake/*"
      ],
      "Condition": {
        "StringEquals": {
```

```
          "partnercentral:Catalog": [
            "AWS",
            "Sandbox"
          ]
        }
      }
    }
  ]
}
```

AWSPartnerCentralChannelHandshakeApprovalManagement politique

Cette politique permet de consulter les demandes de poignée de main des chaînes et d'y répondre. Cette politique doit être appliquée aux rôles des AWS comptes recevant des demandes de poignée de main sur les chaînes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ChannelHandshakeManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:ListChannelHandshakes",
        "partnercentral:AcceptChannelHandshake",
        "partnercentral:RejectChannelHandshake"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "AWS",
            "Sandbox"
          ]
        }
      }
    }
  ]
}
```

Affectation de politiques aux rôles et aux utilisateurs IAM

Procédez comme suit pour attribuer des politiques aux rôles et aux utilisateurs IAM :

1. Connectez-vous au AWS Management Console.
2. Accédez au service IAM.
3. Sélectionnez des rôles ou des utilisateurs, puis choisissez le rôle ou l'utilisateur IAM auquel vous souhaitez associer une politique.
4. Attachez la `AWSPartnerCentralChannelManagement` politique ou votre politique personnalisée au rôle ou à l'utilisateur IAM sélectionné.

Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

Gestion des autorisations à l'aide de clés de condition

Les clés de condition des politiques IAM fournissent des autorisations au niveau des ressources pour savoir quand appliquer les politiques de déclaration. Vous pouvez utiliser les clés de condition pour spécifier les conditions qui déterminent quand certaines autorisations sont autorisées ou refusées.

Pour plus d'informations, consultez Éléments de politique JSON IAM : opérateurs de condition.

Vue d'ensemble des clés de condition

Clé de condition	Description	Actions applicables	Valeurs valides
Partner Central : catalogue	Filtre l'accès en fonction du type de l'entité de catalogue associée	Toutes les actions de gestion des canaux	AWS, Sandbox
Partnercentral : ChannelHandshakeType	Filtre l'accès en fonction du type de connexion à la chaîne	CreateChannelHandshake, AcceptChannelHandshake, RejectChannelHandshake, CancelChannelHandshake, ListChannelHandshakes	COMPTE_DE GESTION DU PROGRAMME, START_SERVICE_PERIOD, REVOKE_SERVICE_PERIOD

Résumé des autorisations requises

Résumé des autorisations requises

Action	Description
Partnercentral : CreateProgramManagementAccount	permet de créer des comptes de gestion de programme
Partnercentral : UpdateProgramManagementAccount	permet de mettre à jour les comptes de gestion des programmes
Partnercentral : DeleteProgramManagementAccount	permet de supprimer des comptes de gestion de programmes
Partnercentral : ListProgramManagementAccounts	permet de répertorier les comptes de gestion du programme
Partnercentral : GetProgramManagementAccount	permet de récupérer les détails du compte de gestion du programme
Partnercentral : CreateRelationship	permet de créer des relations
Partnercentral : UpdateRelationship	permet de mettre à jour les relations
Partnercentral : DeleteRelationship	permet de supprimer des relations
Partnercentral : GetRelationship	permet de récupérer les détails de la relation
Partnercentral : ListRelationships	permet de répertorier les relations
Partnercentral : CreateChannelHandshake	permet de créer des poignées de main pour les chaînes
Partnercentral : AcceptChannelHandshake	permet d'accepter les poignées de main des chaînes
Partnercentral : RejectChannelHandshake	permet de rejeter les poignées de main des chaînes

Action	Description
Partnercentral : CancelChannelHandshake	permet d'annuler les poignées de main des chaînes
Partnercentral : ListChannelHandshakes	permet de répertorier les poignées de main des chaînes

Accès à l'API de mesure des recettes

Le contrôle d'accès et les autorisations sont gérés par Gestion des identités et des accès AWS(IAM). Cette section fournit des conseils pour configurer les autorisations nécessaires pour interagir avec l'API Revenue Measurement, y compris les autorisations requises pour répertorier AWS Marketplace les entités.

Conditions préalables

Avant de configurer les autorisations, assurez-vous que vous Compte AWSêtes connecté à Partner Central et que vous avez créé les rôles et utilisateurs IAM nécessaires. Pour plus d'informations, consultez Configuration et authentification.

Utilisation AWS stratégies gérées

AWS fournit des politiques gérées qui accordent les autorisations requises pour interagir avec l'API Revenue Measurement. Deux politiques gérées sont disponibles en fonction de votre personnalité :

- Partenaires : associez la `AWSPartnerCentralRevenueAttributionManagement` politique à vos identités IAM.
- Clients : associez la `AWSRevenueAttributionManagement` politique à vos identités IAM.

Pour plus d'informations, consultez les politiques AWS gérées pour les utilisateurs de AWS Partner Central.

AWSPartnerCentralRevenueAttributionManagement politique

Cette politique fournit l'accès nécessaire aux activités de gestion de l'attribution des revenus pour les AWS comptes enregistrés auprès de AWS Partner Central.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RevenueMeasurementCreation",
            "Effect": "Allow",
            "Action": [
                "partnercentral:CreateRevenueAttribution",
                "partnercentral:CreateMarketplaceRevenueShare",
                "partnercentral:CreateMarketplaceRevenueShareAllocation"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "RevenueMeasurementListing",
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListRevenueAttributions",
                "partnercentral:ListRevenueAttributionAllocations",
                "partnercentral:ListMarketplaceRevenueShares",
                "partnercentral:ListMarketplaceRevenueShareAllocations"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "RevenueAttributionManagement",
            "Effect": "Allow",
            "Action": [
                "partnercentral:GetRevenueAttribution",
```

```
                "partnercentral:UpdateRevenueAttribution",
                "partnercentral:GetRevenueAttributionAllocation",
                "partnercentral:StartRevenueAttributionAllocationsTask",
                "partnercentral:GetRevenueAttributionAllocationsTask"
            ],
            "Resource": "arn:aws:partnercentral:*:*:catalog/*/revenue-attribution/*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "MarketplaceRevenueShareManagement",
            "Effect": "Allow",
            "Action": [
                "partnercentral:GetMarketplaceRevenueShare",
                "partnercentral:GetMarketplaceRevenueShareAllocation",
                "partnercentral:UpdateMarketplaceRevenueShareAllocation"
            ],
            "Resource": "arn:aws:partnercentral:*:*:catalog/*/marketplace-revenue-
share/*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "TaggingAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:TagResource",
                "partnercentral:UntagResource",
                "partnercentral:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:partnercentral:*:*:catalog/*/revenue-attribution/*",
```

```
                "arn:aws:partnercentral:*:*:catalog/*/marketplace-revenue-share/*"
            ],
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "OpportunityAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListOpportunities",
                "partnercentral:GetOpportunity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "PartnerResourceAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListPartners",
                "partnercentral:GetPartner"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
```

```
        },
        {
            "Sid": "ListingAWSMarketplaceEntities",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListEntities"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AWSMarketplaceEntityAccess",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:DescribeEntity"
            ],
            "Resource": [
                "arn:aws:aws-marketplace:*:*:AWSMarketplace*/Offer/*"
            ]
        },
        {
            "Sid": "AmazonQPartnerAssistantAccess",
            "Effect": "Allow",
            "Action": [
                "q:StartConversation",
                "q:SendMessage",
                "q:GetConversation",
                "q:ListConversations",
                "q:PassRequest"
            ],
            "Resource": "*"
        }
    ]
}
```

AWSRevenueAttributionManagement politique

Cette politique fournit l'accès nécessaire aux activités de gestion de l'attribution des revenus pour les AWS comptes qui ne sont pas enregistrés auprès de AWS Partner Central.

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

```
            "Sid": "RevenueAttributionCreation",
            "Effect": "Allow",
            "Action": [
                "partnercentral:CreateRevenueAttribution"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "RevenueAttributionListing",
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListRevenueAttributions"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "RevenueAttributionManagement",
            "Effect": "Allow",
            "Action": [
                "partnercentral:GetRevenueAttribution",
                "partnercentral:UpdateRevenueAttribution"
            ],
            "Resource": "arn:aws:partnercentral:*:*:catalog/*/revenue-attribution/*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
```

```
                    ]
                }
            }
        },
        {
            "Sid": "TaggingAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:TagResource",
                "partnercentral:UntagResource",
                "partnercentral:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:partnercentral:*:*:catalog/*/revenue-attribution/*"
            ],
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "AWS",
                        "Sandbox"
                    ]
                }
            }
        }
    ]
}
```

Affectation de politiques aux rôles et aux utilisateurs IAM

Procédez comme suit pour attribuer des politiques aux rôles et aux utilisateurs IAM :

1. Connectez-vous au AWS Management Console.
2. Accédez au service IAM.
3. Sélectionnez des rôles ou des utilisateurs, puis choisissez le rôle ou l'utilisateur IAM auquel vous souhaitez associer une politique.
4. Attachez la `AWSPartnerCentralRevenueAttributionManagement` politique (pour les partenaires) ou la `AWSRevenueAttributionManagement` politique (pour les clients) au rôle ou à l'utilisateur IAM sélectionné.

Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

Gestion des autorisations à l'aide de clés de condition

Les clés de condition des politiques IAM fournissent des autorisations au niveau des ressources pour savoir quand appliquer les politiques de déclaration. Vous pouvez utiliser les clés de condition pour spécifier les conditions qui déterminent quand certaines autorisations sont autorisées ou refusées.

Pour plus d'informations, consultez Éléments de politique JSON IAM : opérateurs de condition.

Vue d'ensemble des clés de condition

Clé de condition	Description	Actions applicables	Valeurs valides
Partner Central : catalogue	Filtre l'accès en fonction du type de l'entité de catalogue associée	Toutes les actions	AWS, Bac à sable

Résumé des autorisations requises

Résumé des autorisations requises

Action	Description
Partnercentral : CreateRevenueAttribution	Permet de créer un nouvel enregistrement d'attribution des revenus
Partnercentral : UpdateRevenueAttribution	Permet de mettre à jour un enregistrement d'attribution de revenus existant
Partnercentral : GetRevenueAttribution	Permet de récupérer les détails d'une attributi on de revenus spécifique
Partnercentral : ListRevenueAttributions	Permet de répertorier les attributions de revenus avec des filtres optionnels
Partnercentral : GetRevenueAttributionAlloca tion	Permet de récupérer une allocation unique par son identifiant
Partnercentral : ListRevenueAttributionAlloc ations	Permet de répertorier les allocations validées avec support de filtrage

Action	Description
Partnercentral : StartRevenueAttributionAllocationsTask	Permet de soumettre un lot contenant jusqu'à 250 modifications d'allocation pour un traitement asynchrone
Partnercentral : GetRevenueAttributionAllocationsTask	Permet de récupérer le statut d'une tâche d'allocation précédemment soumise
Partnercentral : CreateMarketplaceRevenueShare	Permet de créer une nouvelle ressource de partage des revenus du marché
Partnercentral : GetMarketplaceRevenueShare	Permet de récupérer les détails d'une part de revenus spécifique du marché
Partnercentral : ListMarketplaceRevenueShares	Permet de répertorier les parts de revenus du marché avec des filtres optionnels
Partnercentral : CreateMarketplaceRevenueShareAllocation	Permet de créer une nouvelle allocation de parts de revenus sur le marché
Partnercentral : GetMarketplaceRevenueShareAllocation	Permet de récupérer les détails de l'allocation de parts de revenus d'une place de marché spécifique
Partnercentral : UpdateMarketplaceRevenueShareAllocation	Permet de mettre à jour l'allocation de parts de revenus d'une place de marché existante
Partnercentral : ListMarketplaceRevenueShareAllocations	Permet d'attribuer des annonces en fonction du partage des revenus du marché
Partnercentral : TagResource	Permet d'ajouter ou de remplacer des balises pour une ressource
Partnercentral : UntagResource	Permet de supprimer des balises d'une ressource
Partnercentral : ListTagsForResource	Permet de répertorier les balises associées à une ressource

Quotas pour AWS API Partner Central

Les sections suivantes fournissent des informations sur les quotas de service.

Rubriques

- Quotas pour AWS API du compte Partner Central
- Quotas pour le AWS API de vente centralisée pour les partenaires
- Quotas pour le AWS API Partner Central Benefits
- Quotas pour le AWS API Partner Central Channel
- Quotas pour le AWS API de mesure du chiffre d'affaires de Partner Central

Quotas pour AWS API du compte Partner Central

L'API du compte AWS Partner Central possède les quotas suivants.

Quotas supplémentaires

Quotas supplémentaires

Nom d’affichage	Catalogue	Description	Valeur par défaut
Invitations de connexion ouvertes par compte	AWS	Le nombre maximum d'invitations de connexion ouvertes que vous pouvez conserver avec les comptes partenaires du AWS catalogue	1 000
Connexions actives par compte	AWS	Le nombre maximum de connexions actives que vous pouvez maintenir avec les comptes partenaires du AWS catalogue	1 000

Nom d’affichage	Catalogue	Description	Valeur par défaut
Domaines de messagerie par partenaire	AWS	Le nombre maximum de domaines de messagerie pouvant être associés à un compte partenaire pour la certification des AWS formation s dans le AWS catalogue	50
Taux d'invitations de connexion par compte	AWS	Le nombre maximum d'invitations de connexion par jour que vous pouvez envoyer dans le AWS catalogue	50
Taux de tâches de mise à jour du profil par compte	AWS	Le nombre maximum de tâches de mise à jour de profil que vous pouvez lancer par jour dans le AWS catalogue	10
Taux de mises à jour de visibilité du profil par compte	AWS	Le nombre maximum de mises à jour de visibilité du profil par jour que vous pouvez initier dans le AWS catalogue	5

Nom d’affichage	Catalogue	Description	Valeur par défaut
Taux de codes de vérification par adresse e-mail	AWS	Le nombre maximum de codes de vérification par e-mail que vous pouvez demander dans le AWS catalogue par jour	5
Invitations de connexion ouvertes par compte	Sandbox	Le nombre maximum d'invitations de connexion ouvertes que vous pouvez conserver avec les comptes partenaires dans le catalogue Sandbox	1 000
Connexions actives par compte	Sandbox	Le nombre maximum de connexions actives que vous pouvez maintenir avec des comptes partenaires dans le catalogue Sandbox	1 000
Domaines de messagerie par partenaire	Sandbox	Le nombre maximum de domaines de messagerie pouvant être associés à un compte partenaire pour la certification de AWS formation dans le catalogue Sandbox	50

Nom d’affichage	Catalogue	Description	Valeur par défaut
Taux d'invitations de connexion par compte	Sandbox	Le nombre maximum d'invitations de connexion que vous pouvez envoyer par jour dans le catalogue Sandbox	50
Taux de tâches de mise à jour du profil par compte	Sandbox	Le nombre maximal de tâches de mise à jour de profil que vous pouvez lancer par jour dans le catalogue Sandbox	10
Taux de mises à jour de visibilité du profil par compte	Sandbox	Nombre maximal de mises à jour de visibilité du profil par jour que vous pouvez initier dans le catalogue Sandbox	5
Taux de codes de vérification par adresse e-mail	Sandbox	Le nombre maximum de codes de vérificat ion par e-mail que vous pouvez demander dans le catalogue Sandbox par jour. Aucun e-mail n'est envoyé depuis le catalogue Sandbox.	N/A

Comprendre et gérer les quotas

Limitation du débit

Lorsqu'une limite de débit d'API est atteinte, le service répond par un ThrottlingException. Pour mieux gérer la limitation du débit, AWS recommande de mettre en œuvre des stratégies de réduction et de réessai exponentielles dans votre application.

Demande d’augmentation de quota

Si les quotas par défaut ne répondent pas à vos exigences, vous pouvez demander une augmentation de quota via la page Service Quotas. La console Service Quotas est une interface basée sur un navigateur que vous pouvez utiliser pour consulter et gérer vos quotas de service. Vous pouvez accéder aux Quotas de Service depuis n'importe quelle AWS Management Console page en la sélectionnant dans la barre de navigation supérieure ou en recherchant Service Quotas dans le AWS Management Console.

Quotas pour le AWS API de vente centralisée pour les partenaires

AWS L'API de vente de Partner Central applique des quotas afin de garantir une utilisation équitable et de protéger le service contre toute utilisation abusive. Vous trouverez ci-dessous les quotas détaillés pour les différentes opérations et associations d'API par opportunité.

Quotas d'opérations d'API

Type	Opération API	Quota (par compte partenaire)
Lire les actions	GetOpportunity	10 par seconde ; 100 000 par 24 heures
	GetAwsOpportunitySummary	
	ListOpportunities	
	ListSolutions	
	GetEngagementInvitation	
	ListEngagementInvitations	
Prospection, lecture, actions	GetProspectingFrom EngagementTask	100 par seconde

Type	Opération API	Quota (par compte partenaire)
Prospection, lecture, actions	ListProspectingFromEngagementTasks	50 par seconde
Actions d'écriture	CreateOpportunity	1 par seconde ; 10 000 par 24 heures
	UpdateOpportunity	
	AssociateOpportunity	
	DisassociateOpportunity	
	RejectEngagementInvitation	
	AssignOpportunity	
	StartEngagementFromOpportunityTask	
	StartEngagementByAcceptingInvitationTask	
Actions de prospection et d'écriture	StartProspectingFromEngagementTask	2 par seconde
Actions d'engagement et de rédaction	CreateEngagement	15 par seconde

Quotas d'association par opportunité

Entité associée	Quota
AWS produits	20 par opportunité
Solutions partenaires	10 par opportunité
AWS Marketplace solutions	10 par opportunité

Entité associée	Quota
AWS Marketplace produits	10 par opportunité
AWS Marketplace offres privées	1 par opportunité

Comprendre et gérer les quotas

Limitation du débit

Lorsqu'une limite de débit d'API est atteinte, le service répond par un ThrottlingException. Pour mieux gérer la limitation du débit, AWS recommande de mettre en œuvre des stratégies de réduction et de réessai exponentielles dans votre application.

Fenêtre temporelle pour les quotas

Les quotas quotidiens sont réinitialisés sur une période continue de 24 heures. Vos demandes seront limitées, par exemple si vous avez effectué 10 000 actions d'écriture au cours des dernières 24 heures et que vous essayez d'exécuter la 10 001e demande. Assurez-vous que les modèles d'utilisation de votre application en tiennent compte afin d'éviter tout ralentissement involontaire.

Demande d’augmentation de quota

Si les quotas par défaut ne répondent pas à vos exigences, vous pouvez demander une augmentation de quota via la page Service Quotas. La console Service Quotas est une interface basée sur un navigateur que vous pouvez utiliser pour consulter et gérer vos quotas de service. Vous pouvez accéder aux Quotas de Service depuis n'importe quelle AWS Management Console page en la sélectionnant dans la barre de navigation supérieure ou en recherchant Service Quotas dans le AWS Management Console.

Quotas pour le AWS API Partner Central Benefits

L'API AWS Partner Central Benefits possède les quotas suivants.

Quotas de demande

Quotas de demande

opérations d’API	Quota (par AWS compte)
ListBenefits	20 par seconde
GetBenefit	20 par seconde
AmendBenefitApplication	10 par seconde
CancelBenefitApplication	10 par seconde
CreateBenefitApplication	10 par seconde
GetBenefitApplication	20 par seconde
ListBenefitApplications	30 par seconde
RecallBenefitApplication	10 par seconde
SubmitBenefitApplication	10 par seconde
UpdateBenefitApplication	10 par seconde
GetBenefitAllocation	20 par seconde
ListBenefitAllocations	20 par seconde
AssociateBenefitApplicationResource	10 par seconde
DisassociateBenefitApplicationResource	10 par seconde

Quotas supplémentaires

Quotas supplémentaires

Nom d’affichage	Catalogue	Description	Valeur par défaut
Demandes de prestations	AWS	Le nombre maximum de demandes de	10 000

Nom d’affichage	Catalogue	Description	Valeur par défaut
		prestations que vous pouvez créer dans le AWS catalogue	
Demandes de prestations	Sandbox	Le nombre maximum de demandes d'avantages que vous pouvez créer dans le catalogue Sandbox	10 000

Comprendre et gérer les quotas

Limitation du débit

Lorsqu'une limite de débit d'API est atteinte, le service répond par un ThrottlingException. Pour mieux gérer la limitation du débit, AWS recommande de mettre en œuvre des stratégies de réduction et de réessai exponentielles dans votre application.

Demande d’augmentation de quota

Si les quotas par défaut ne répondent pas à vos exigences, vous pouvez demander une augmentation de quota via la page Service Quotas. La console Service Quotas est une interface basée sur un navigateur que vous pouvez utiliser pour consulter et gérer vos quotas de service. Vous pouvez accéder aux Quotas de Service depuis n'importe quelle AWS Management Console page en la sélectionnant dans la barre de navigation supérieure ou en recherchant Service Quotas dans le AWS Management Console.

Quotas pour le AWS API Partner Central Channel

L'API AWS Partner Central Channel possède les quotas suivants.

Quotas de demande

Quotas de demande

opérations d’API	Quota (par AWS compte)
CreateProgramManagementAccount	3 par seconde

opérations d’API	Quota (par AWS compte)
UpdateProgramManagementAccount	3 par seconde
ListProgramManagementAccounts	5 par seconde
DeleteProgramManagementAccount	3 par seconde
CreateChannelHandshake	3 par seconde
ListChannelHandshakes	5 par seconde
AcceptChannelHandshake	3 par seconde
RejectChannelHandshake	3 par seconde
CancelChannelHandshake	3 par seconde
CreateRelationship	3 par seconde
GetRelationship	5 par seconde
ListRelationships	5 par seconde
UpdateRelationship	3 par seconde
DeleteRelationship	3 par seconde

Quotas supplémentaires

Quotas supplémentaires

Nom d’affichage	Catalogue	Description	Valeur par défaut
Comptes de gestion du programme	AWS	Le nombre maximum de comptes de gestion de programmes que vous pouvez créer dans le AWS catalogue	50

Nom d’affichage	Catalogue	Description	Valeur par défaut
Relations par compte de gestion de programme	AWS	Le nombre maximum de relations que vous pouvez établir par compte de gestion de programme dans le AWS catalogue	1 000
Gestion du programme : poignées de main par compte	AWS	Le nombre maximum de poignées de main de comptes de gestion de programmes actives dans le catalogue AWS	100
Poignées de main au début de la période de service par compte	AWS	Le nombre maximum de poignées de main actives pour établir des périodes de service dans le catalogue AWS	200
Révoquer les poignées de main relatives à la période de service par compte	AWS	Le nombre maximum de poignées de main actives pour révoquer des périodes de service dans le catalogue AWS	200

Nom d’affichage	Catalogue	Description	Valeur par défaut
Taux de poignées de main quotidiennes liées aux comptes de gestion du programme	AWS	Le nombre maximum de poignées de main que vous pouvez envoyer à un même AWS compte par jour dans le catalogue AWS	5
Taux de début de la période de service : poignées de main par jour	AWS	Le nombre maximum de poignées de main de début de période de service que vous pouvez envoyer sur le même AWS compte par jour dans le catalogue AWS	5
Taux de poignées de main par jour pendant la période de service révoquée	AWS	Le nombre maximum de poignées de main relatives à la période de révocation que vous pouvez envoyer sur le même AWS compte par jour dans le catalogue AWS	5
Comptes de gestion du programme	Sandbox	Le nombre maximum de comptes de gestion de programmes que vous pouvez créer dans le catalogue Sandbox	50

Nom d’affichage	Catalogue	Description	Valeur par défaut
Relations par compte de gestion de programme	Sandbox	Le nombre maximum de relations que vous pouvez établir par compte de gestion de programme dans le catalogue Sandbox	1 000
Gestion du programme : poignées de main par compte	Sandbox	Nombre maximum de poignées de main de comptes de gestion de programme s actives dans le catalogue Sandbox	100
Poignées de main au début de la période de service par compte	Sandbox	Le nombre maximum de poignées de main actives pour établir des périodes de service dans le catalogue Sandbox	200
Révoquer les poignées de main relatives à la période de service par compte	Sandbox	Le nombre maximum de poignées de main actives pour révoquer des périodes de service dans le catalogue Sandbox	200

Nom d’affichage	Catalogue	Description	Valeur par défaut
Taux de poignées de main quotidiennes liées aux comptes de gestion du programme	Sandbox	Le nombre maximum de poignées de main d'un compte de gestion de programme que vous pouvez envoyer au même AWS compte par jour dans le catalogue Sandbox	5
Taux de début de la période de service : poignées de main par jour	Sandbox	Le nombre maximum de poignées de main de début de période de service que vous pouvez envoyer au même AWS compte par jour dans le catalogue Sandbox	5
Taux de poignées de main par jour pendant la période de service révoquée	Sandbox	Le nombre maximum de poignées de main relatives à la période de service de révocation que vous pouvez envoyer au même AWS compte par jour dans le catalogue Sandbox	5

Comprendre et gérer les quotas

Limitation du débit

Lorsqu'une limite de débit d'API est atteinte, le service répond par un ThrottlingException. Pour mieux gérer la limitation du débit, AWS recommande de mettre en œuvre des stratégies de réduction et de réessai exponentielles dans votre application.

Demande d’augmentation de quota

Si les quotas par défaut ne répondent pas à vos exigences, vous pouvez demander une augmentation de quota via la page Service Quotas. La console Service Quotas est une interface basée sur un navigateur que vous pouvez utiliser pour consulter et gérer vos quotas de service. Vous pouvez accéder aux Quotas de Service depuis n'importe quelle AWS Management Console page en la sélectionnant dans la barre de navigation supérieure ou en recherchant Service Quotas dans le AWS Management Console.

Quotas pour le AWS API de mesure du chiffre d'affaires de Partner Central

AWS L'API Partner Central Revenue Measurement applique des quotas afin de garantir une utilisation équitable et de protéger le service contre toute utilisation abusive. Vous trouverez ci-dessous les quotas détaillés pour les différentes opérations d'API.

Quotas d'opérations d'API

Type	Opération API	Quota (par compte partenaire)
Lire les actions	GetRevenueAttribution	50 par seconde
	GetRevenueAttributionAllocation	
	GetRevenueAttributionAllocationsTask	
	GetMarketplaceRevenueShare	
	GetMarketplaceRevenueShareAllocation	

Type	Opération API	Quota (par compte partenaire)
Lister les actions	ListRevenueAttributions	10 par seconde
	ListRevenueAttributionAllocations	
	ListMarketplaceRevenueShares	
	ListMarketplaceRevenueShareAllocations	
	ListTagsForResource	
Actions d'écriture	CreateRevenueAttribution	2 par seconde
	UpdateRevenueAttribution	
	CreateMarketplaceRevenueShare	
	CreateMarketplaceRevenueShareAllocation	
	UpdateMarketplaceRevenueShareAllocation	
	TagResource	
	UntagResource	
Actions d'écriture asynchrones	StartRevenueAttributionAllocationsTask	1 par seconde

Comprendre et gérer les quotas

Limitation du débit

Lorsqu'une limite de débit d'API est atteinte, le service répond par un ThrottlingException. Pour mieux gérer la limitation du débit, il est AWS recommandé de mettre en œuvre des stratégies de réduction et de réessai exponentielles dans votre application.

Demande d’augmentation de quota

Si les quotas par défaut ne répondent pas à vos exigences, vous pouvez demander une augmentation de quota via la page Service Quotas. La console Service Quotas est une interface basée sur un navigateur que vous pouvez utiliser pour consulter et gérer vos quotas de service. Vous pouvez accéder aux Quotas de Service depuis n'importe quelle AWS Management Console page en la sélectionnant dans la barre de navigation supérieure ou en recherchant Service Quotas dans le AWS Management Console.

Journalisation pour le AWS API Partner Central

Les sections suivantes fournissent des informations sur la journalisation.

Rubriques

- Enregistrement du AWS API du compte Partner Central
- Enregistrement du AWS API de vente centralisée pour les partenaires
- Enregistrement du AWS API Partner Central Benefits
- Enregistrement du AWS API Partner Central Channel
- Enregistrement du AWS API d'attribution des revenus pour Partner Central

Enregistrement du AWS API du compte Partner Central

AWS CloudTrailest un service qui permet la gouvernance, la conformité, l'audit opérationnel et l'audit des risques de votre entreprise Compte AWS. Vous pouvez ainsi enregistrer, surveiller en permanence et conserver l'activité du compte liée aux actions menées au sein de votre AWS infrastructure.AWS CloudTrailAWS L'activité de l'API du compte Partner Central est enregistrée sous forme d'événements dans CloudTrail. Vous pouvez créer un journal, une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3.

Présentation de

L'API du compte AWS Partner Central est intégrée àAWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS Partner Central. CloudTrail capture tous les appels d'API pour l'API du compte AWS Partner Central sous forme d'événements. Les appels capturés incluent des appels provenant de AWS Partner Central et des appels de code vers les opérations de l'API du compte AWS Partner Central.

Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour l'API du compte AWS Partner Central. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements.

À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à l'API AWS Partner Central Account, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Compréhension AWS Entrées dans le fichier journal de l'API Partner Central Account

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3. Lorsque votre parcours suit les événements de l'API AWS Partner Central Account, il les CloudTrail traite sous forme de fichiers journaux dans toutes les régions. Chaque fichier journal peut contenir un ou plusieurs événements.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre l'`CreatePartner`action sur l'API du compte AWS Partner Central :

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/CloudTrailTestUser",
        "accountId": "123456789012",
        "accessKeyId": "ABCDEFGHIJKLMNOP1234",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLE52AGFT725JGDZ",
                "arn": "arn:aws:iam::123456789012:role/ExampleRole",
                "accountId": "123456789012",
                "userName": "ExampleRole"
```

```
            },
            "attributes": {
                "creationDate": "2025-11-07T19:45:28Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2025-11-07T19:45:58Z",
    "eventSource": "partnercentral-account.amazonaws.com",
    "eventName": "CreatePartner",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "PostmanRuntime/7.18.0",
    "requestParameters": {
        "catalog": "AWS",
        "clientToken": "abcdef12-3456-7890-bcde-f123456789ab",
        "legalName": "ExampleLegalName",
        "primarySolutionType": "VALUE_ADDED_RESALE_AWS_SERVICES",
        "allianceLeadContact": {
            "firstName": "ExampleFirstName",
            "lastName": "ExampleLastName",
            "email": "example@domain.com",
            "businessTitle": "ExampleBusinessTitle"
        },
        "emailVerificationCode": "ExampleVerificationCode",
        "tags": [
            {
                "key": "office",
                "value": "1"
            }
        ]
    },
    "responseElements": {
        "catalog": "AWS",
        "arn": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/partner/
EXAMPLE_PARTNER_ID",
        "id": "EXAMPLE_PARTNER_ID",
        "legalName": "ExampleLegalName",
        "createdAt": "2025-11-07T19:45:57.867007329Z",
        "profile": {
            "primarySolutionType": "VALUE_ADDED_RESALE_AWS_SERVICES"
        },
        "allianceLeadContact": {
            "firstName": "ExampleFirstName",
```

```
            "lastName": "ExampleLastName",
            "email": "example@domain.com",
            "businessTitle": "ExampleBusinessTitle"
        }
    },
    "requestID": "12345678-1234-5678-9abc-def012345678",
    "eventID": "87654321-4321-8765-cba9-fed098765432",
    "readOnly": false,
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::PartnerCentral::Partner",
            "ARN": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/partner/
EXAMPLE_PARTNER_ID"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "partnercentral-account.partner.us-east-1.api.aws"
    }
}
```


Dans cet exemple, l'`CreatePartner`action a été appelée par l'utilisateur IAM nommé CloudTrailTestUser. La demande a été faite le 7 novembre 2025 à 19:45:58 UTC. La demande a créé un nouveau partenaire avec un identifiant `EXAMPLE_PARTNER_ID` pour le catalogue AWS avec le nom légal « ExampleLegalName ».

Champs dans AWS Entrées dans le fichier journal de l'API Partner Central Account

Chaque entrée d'un fichier CloudTrail journal contient des informations sur l'auteur de la demande, les ressources utilisées dans la demande et les éléments de réponse renvoyés par l'API AWS Partner Central Account. La liste des champs d'une entrée de journal, tels que `eventVersion`et`userIdentity`, et`eventTime`, fournit des informations détaillées sur l'action. Par exemple, le `sourceIPAddress` champ indique l'adresse IP à partir de laquelle la demande a été faite.

Enregistrement du AWS API de vente centralisée pour les partenaires

AWS CloudTrailest un service qui permet la gouvernance, la conformité, l'audit opérationnel et l'audit des risques de votre entreprise Compte AWS. Vous pouvez ainsi enregistrer, surveiller en permanence et conserver l'activité du compte liée aux actions menées au sein de votre AWS infrastructure.AWS CloudTrailAWS L'activité de l'API Partner Central est enregistrée sous forme d'événements dans CloudTrail. Vous pouvez créer un journal, une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3.

Présentation de

L'API AWS Partner Central est intégrée àAWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS Partner Central. CloudTrail capture tous les appels d'API pour AWS Partner Central sous forme d'événements. Les appels capturés incluent des appels provenant de AWS Partner Central et des appels de code vers les opérations de l'API AWS Partner Central.

Si vous créez un suivi, vous pouvez activer la diffusion continue des CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour AWS Partner Central. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements.

À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée àAWS Partner Central, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite, ainsi que des informations supplémentaires.

Compréhension AWS Entrées dans le fichier journal de l'API Partner Central Selling

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3. Lorsque votre parcours suit les événements de AWS Partner Central, les CloudTrail traite sous forme de fichiers journaux dans toutes les régions. Chaque fichier journal peut contenir un ou plusieurs événements.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`ListOpportunities`action sur AWS Partner Central :

```
{
    "eventVersion": "1.05",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "ABCDEFGHIJKLMNOP12345",
        "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
        "accountId": "123456789010",
        "accessKeyId": "ABCDEFGHIJKLMNOP1234",
        "userName": "CloudTrailTestUser"
    },
    "eventTime": "2023-10-17T21:49:23Z",
    "eventSource": "partnercentral-selling.amazonaws.com",
    "eventName": "ListOpportunities",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "PostmanRuntime/7.18.0",
    "requestParameters": {
        "MaxResults": 20
    },
    "responseElements": null,
    "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789010"
}
```


Dans cet exemple, l'`ListOpportunities`action a été appelée par l'utilisateur IAM nommé CloudTrailTestUser. L'action a été appelée dans le Région AWS us-east-1, et la demande a été faite le 17 octobre 2023 à 21:49:23 UTC.

Champs dans AWS Entrées dans le fichier journal de l'API Partner Central Selling

Chaque entrée d'un fichier CloudTrail journal contient des informations sur l'auteur de la demande, les ressources utilisées dans le cadre de la demande et les éléments de réponse renvoyés par AWS Partner Central. La liste des champs d'une entrée de journal, tels que `eventVersion`ou `userIdentity`, et`eventTime`, fournit des informations détaillées sur l'action. Par exemple, le `sourceIPAddress` champ indique l'adresse IP à partir de laquelle la demande a été faite.

Enregistrement du AWS API Partner Central Benefits

AWS CloudTrailest un service qui permet la gouvernance, la conformité, l'audit opérationnel et l'audit des risques de votre entreprise Compte AWS. Vous pouvez ainsi enregistrer, surveiller

en permanence et conserver l'activité du compte liée aux actions menées au sein de votre AWS infrastructure.AWS CloudTrailAWS L'activité de l'API Partner Central Benefits est enregistrée sous forme d'événements dans CloudTrail. Vous pouvez créer un journal, une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3.

Présentation de

L'API AWS Partner Central Benefits est intégrée àAWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS Partner Central. CloudTrail capture tous les appels d'API pour l'API AWS Partner Central Benefits sous forme d'événements. Les appels capturés incluent des appels provenant de AWS Partner Central et des appels de code vers les opérations de l'API AWS Partner Central Benefits.

Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour l'API AWS Partner Central Benefits. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements.

À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à l'API AWS Partner Central Benefits, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite et des informations supplémentaires.

Compréhension AWS Entrées dans le fichier journal de l'API Partner Central Benefits

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3. Lorsque votre parcours suit les événements de l'API AWS Partner Central Benefits, il les CloudTrail traite sous forme de fichiers journaux dans toutes les régions. Chaque fichier journal peut contenir un ou plusieurs événements.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre l'`ListBenefitApplications`action sur l'API AWS Partner Central Benefits :


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/CloudTrailTestUser",
        "accountId": "123456789012",
        "accessKeyId": "EXAMPLE_KEY_ID",
```

```
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "EX_PRINCIPAL_ID",
                "arn": "arn:aws:iam::123456789012:role/TestRole",
                "accountId": "123456789012",
                "userName": "TestRole"
            },
            "attributes": {
                "creationDate": "2025-10-21T03:08:23Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2025-10-21T03:10:59Z",
    "eventSource": "partnercentral-benefits.amazonaws.com",
    "eventName": "ListBenefitApplications",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "python-requests/2.32.4",
    "requestParameters": {
        "catalog": "AWS"
    },
    "responseElements": null,
    "requestID": "12345678-1234-5678-9abc-def012345678",
    "eventID": "87654321-4321-8765-cba9-fed098765432",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


Dans cet exemple, l'`ListBenefitApplications`action a été appelée par l'utilisateur IAM Alice. La demande a été faite le 21 octobre 2025 à 03:10:59 UTC. La demande listait les demandes d'avantages pour le catalogue AWS.

Champs dans AWS Entrées dans le fichier journal de l'API Partner Central Benefits

Chaque entrée d'un fichier CloudTrail journal contient des informations sur l'auteur de la demande, les ressources utilisées dans la demande et les éléments de réponse renvoyés par l'API AWS Partner Central Benefits. La liste des champs d'une entrée de journal, tels que `eventVersion`, `userIdentity`, et`eventTime`, fournit des informations détaillées sur l'action. Par

exemple, le `sourceIPAddress` champ indique l'adresse IP à partir de laquelle la demande a été faite.

Enregistrement du AWS API Partner Central Channel

AWS CloudTrailest un service qui permet la gouvernance, la conformité, l'audit opérationnel et l'audit des risques de votre entreprise Compte AWS. Vous pouvez ainsi enregistrer, surveiller en permanence et conserver l'activité du compte liée aux actions menées au sein de votre AWS infrastructure.AWS CloudTrailAWS L'activité de l'API Partner Central Channel est enregistrée sous forme d'événements dans CloudTrail. Vous pouvez créer un journal, une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3.

Présentation de

L'API AWS Partner Central Channel est intégrée àAWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS Partner Central. CloudTrail capture tous les appels d'API pour l'API AWS Partner Central Channel sous forme d'événements. Les appels capturés incluent des appels provenant de AWS Partner Central et des appels de code vers les opérations de l'API AWS Partner Central Channel.

Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour l'API AWS Partner Central Channel. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements.

À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande envoyée à l'API AWS Partner Central Channel, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite et des informations supplémentaires.

Compréhension AWS Entrées dans le fichier journal de l'API Partner Central Channel

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3. Lorsque votre parcours suit les événements de l'API AWS Partner Central Channel, il les CloudTrail traite sous forme de fichiers journaux dans toutes les régions. Chaque fichier journal peut contenir un ou plusieurs événements.

L'exemple suivant montre une entrée de CloudTrail journal qui illustre l'`CreateProgramManagementAccount`action sur l'API AWS Partner Central Channel :

```
{
```

```
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLE52AGFT725JGDZ:example-user-Isengard",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/example-user-Isengard",
        "accountId": "123456789012",
        "accessKeyId": "ASIAEXAMPLE52AGL6IXQLZ5",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLE52AGFT725JGDZ",
                "arn": "arn:aws:iam::123456789012:role/ExampleRole",
                "accountId": "123456789012",
                "userName": "ExampleRole"
            },
            "attributes": {
                "creationDate": "2025-10-21T17:06:47Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2025-10-21T17:07:26Z",
    "eventSource": "partnercentral-channel.amazonaws.com",
    "eventName": "CreateProgramManagementAccount",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "PostmanRuntime/7.18.0",
    "requestParameters": {
        "catalog": "AWS",
        "program": "SOLUTION_PROVIDER",
        "displayName": "ExampleDisplayName",
        "accountId": "987654321098",
        "clientToken": "abcdef12-3456-7890-bcde-f123456789ab"
    },
    "responseElements": {
        "programManagementAccountDetail": {
            "id": "pma-example123456789",
            "arn": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/program-
management-account/pma-example123456789"
        }
    },
    "requestID": "12345678-1234-5678-9abc-def012345678",
    "eventID": "87654321-4321-8765-cba9-fed098765432",
    "readOnly": false,
```

```
    "resources": [
        {
            "accountId": "123456789012",
            "type": "AWS::PartnerCentralChannel::ProgramManagementAccount",
            "ARN": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/program-
management-account/pma-example123456789"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "clientProvidedHostHeader": "partnercentral-channel.global.api.aws"
    }
}
```

Dans cet exemple, l'`CreateProgramManagementAccount`action a été appelée par le rôle IAM nommé ExampleRole via une session de rôle assumé. La demande a été faite le 21 octobre 2025 à 17:07:26 UTC. La demande a créé un nouveau compte de gestion du programme avec un identifiant `pma-example123456789` pour le programme Solution Provider.

Champs dans AWS Entrées dans le fichier journal de l'API Partner Central Channel

Chaque entrée d'un fichier CloudTrail journal contient des informations sur l'auteur de la demande, les ressources utilisées dans le cadre de la demande et les éléments de réponse renvoyés par l'API AWS Partner Central Channel. La liste des champs d'une entrée de journal, tels que `eventVersion`, `userIdentity`, et`eventTime`, fournit des informations détaillées sur l'action. Par exemple, le `sourceIPAddress` champ indique l'adresse IP à partir de laquelle la demande a été faite.

Enregistrement du AWS API d'attribution des revenus pour Partner Central

AWS CloudTrailest un service qui permet la gouvernance, la conformité, l'audit opérationnel et l'audit des risques de votre entreprise Compte AWS. Vous pouvez ainsi enregistrer, surveiller en permanence et conserver l'activité du compte liée aux actions menées au sein de votre AWS infrastructure.AWS CloudTrailAWS L'activité de l'API Partner Central Revenue Measurement est enregistrée sous forme d'événements dans CloudTrail. Vous pouvez créer un journal, une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3.

Présentation de

L'API de mesure des recettes de AWS Partner Central est intégrée à un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS Partner Central.AWS CloudTrail CloudTrail capture tous les appels d'API pour AWS Partner Central Revenue Attribution sous forme d'événements. Les appels capturés incluent des appels provenant de la console AWS Partner Central et des appels de code vers les opérations de l'API AWS Partner Central Revenue Measurement.

Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour la mesure des revenus de AWS Partner Central. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements.

À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite àAWS Partner Central Revenue Measurement, l'adresse IP à partir de laquelle la demande a été faite, l'auteur de la demande, la date à laquelle elle a été faite et des informations supplémentaires.

Les actions de l'API AWS Partner Central Revenue Measurement suivantes sont enregistrées CloudTrail :

- `CreateRevenueAttribution`
- `UpdateRevenueAttribution`
- `GetRevenueAttribution`
- `ListRevenueAttributions`
- `TagResource`
- `UntagResource`
- `ListTagsForResource`
- `CreateMarketplaceRevenueShare`
- `GetMarketplaceRevenueShare`
- `ListMarketplaceRevenueShares`
- `CreateMarketplaceRevenueShareAllocation`
- `GetMarketplaceRevenueShareAllocation`
- `UpdateMarketplaceRevenueShareAllocation`

- `ListMarketplaceRevenueShareAllocations`
- `StartRevenueAttributionAllocationsTask`
- `GetRevenueAttributionAllocationsTask`
- `ListRevenueAttributionAllocations`
- `GetRevenueAttributionAllocation`

Compréhension AWS Entrées du fichier journal de l'API Partner Central Revenue Measurement

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3. Lorsque votre parcours suit les événements de mesure des recettes de AWS Partner Central, il CloudTrail traite ces événements sous forme de fichiers journaux dans toutes les régions. Chaque fichier journal peut contenir un ou plusieurs événements.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateRevenueAttribution`action sur l'attribution des revenus de AWS Partner Central :

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAEXAMPLEID",
        "arn": "arn:aws:iam::123456789012:user/CloudTrailTestUser",
        "accountId": "123456789012",
        "accessKeyId": "AKIAEXAMPLEKEY",
        "userName": "CloudTrailTestUser"
    },
    "eventTime": "2026-06-02T11:53:54Z",
    "eventSource": "partnercentral-prm.amazonaws.com",
    "eventName": "CreateRevenueAttribution",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.1",
    "userAgent": "aws-sdk-java/2.20.0",
    "requestParameters": {
        "catalog": "AWS",
        "name": "my-revenue-attribution",
        "marketplaceProduct": {
            "tenancyModel": "MULTI_TENANT"
        }
```

```
    },
    "responseElements": {
        "id": "ra-0123456789abcdef0",
        "arn": "arn:aws:partnercentral:us-east-1:123456789012:catalog/AWS/revenue-
attribution/ra-0123456789abcdef0",
        "name": "my-revenue-attribution",
        "revision": "1"
    },
    "requestID": "5ce2f907-3850-412a-b1a4-r012r1234567",
    "eventID": "80b39b72-eefe-4578-8db0-01ee2e34ee56",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```


Dans cet exemple, l'`CreateRevenueAttribution`action a été appelée par l'utilisateur IAM nommé CloudTrailTestUser. L'action a été appelée dans le Région AWS us-east-1, et la demande a été faite le 2 juin 2026 à 11:53:54 UTC. Une nouvelle ressource d'attribution des revenus a été créée avec l'ID`ra-0123456789abcdef0`.

Champs dans AWS Entrées du fichier journal de l'API Partner Central Revenue Measurement

Chaque entrée d'un fichier CloudTrail journal contient des informations sur l'auteur de la demande, les ressources utilisées dans le cadre de la demande et les éléments de réponse renvoyés par AWS Partner Central Revenue Measurement. La liste des champs d'une entrée de journal, tels que `eventVersion`u`serIdentity`, et`eventTime`, fournit des informations détaillées sur l'action. Par exemple, le `sourceIPAddress` champ indique l'adresse IP à partir de laquelle la demande a été faite.

Notifications pour le AWS API Partner Central

Les sections suivantes fournissent des informations sur les notifications.

Rubriques

- Notifications pour AWS API du compte Partner Central
- Notifications pour AWS API de vente centralisée pour les partenaires
- Notifications pour AWS API Partner Central Benefits

- Notifications pour AWS API Partner Central Channel

Notifications pour AWS API du compte Partner Central

Les notifications de connexion aux comptes partenaires permettent aux partenaires de rester informés des événements du cycle de vie des connexions qui ont un impact direct sur leurs relations commerciales et leurs flux de travail opérationnels. Ces événements sont essentiels pour que les partenaires puissent :

- Réagissez rapidement à une nouvelle collaboration
- Restez au courant de l'état de leur portefeuille de connexions
- Prendre les mesures appropriées lorsque les connexions sont établies ou interrompues
- Garantissez la continuité des activités en sachant quand les relations changent

Rubriques

- Remplissez le prérequis pour surveiller les événements
- Configurer Amazon EventBridge pour surveiller les événements
- En savoir plus sur les événements de l'API de connexion aux comptes

Remplissez le prérequis pour surveiller les événements

Les utilisateurs doivent disposer des autorisations IAM appropriées pour accéder aux événements publiés par l'API de connexion aux comptes et les gérer. Pour plus d'informations sur les actions, les ressources et les clés de condition disponibles pour EventBridge, consultez la section Utilisation des conditions de politique IAM sur Amazon EventBridge dans le guide de l' EventBridge utilisateur Amazon. L'une des clés de condition est `events:detail-type` celle qui peut être utilisée pour étendre les autorisations à des types d'événements spécifiques.

L'exemple de politique suivant montre comment personnaliser et étendre les autorisations pour les événements proposés. L'`AllowPutRuleForPartnercentralAccountEvents`instruction permet de créer des règles, mais uniquement pour les événements provenant de la `aws.partnercentral-account` source.

Pour obtenir des exemples détaillés de politique IAM, consultez la documentation AWS sur EventBridge les autorisations.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralAccountEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-account.connection"
        }
      }
    }
  ]
}
```

Configurer Amazon EventBridge pour surveiller les événements

Pour surveiller les événements de l'API de connexion aux comptes, vous devez créer une EventBridge règle qui correspond aux événements que vous souhaitez capturer. Vous pouvez utiliser le AWS Management Console ou les AWS SDK pour créer et gérer des règles. Les sections suivantes expliquent comment créer des règles à l'aide des deux méthodes. Quelle que soit la méthode utilisée, vous devez créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

AWS Management Console configuration

Pour configurer une EventBridge règle à l'aide du AWS Management Console, suivez les étapes décrites dans la section Création de règles réagissant aux événements survenus sur Amazon EventBridge dans le guide de EventBridge l'utilisateur Amazon. Lors de la création de la règle, vous devez définir le bus d'événements par défaut et créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

Voici un exemple de règle d'événement :

```
{
    "source": ["aws.partnercentral-account"],
    "detail": {
        "catalog": ["AWS"]
    }
}
```

AWS Configuration du SDK

Vous pouvez utiliser les AWS SDK pour créer et gérer des EventBridge règles par programmation. Pour plus d'informations, consultez PutRulele Amazon EventBridge API Reference.

L'exemple suivant utilise le AWS SDK pour Python (Boto3) :

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='MyConnectionInvitationReceivedRule',
    EventPattern=
    '{
        "source": ["aws.partnercentral-account"],
        "detail-type": ["Partner Connection Invitation Received"],
        "detail": {"catalog": ["AWS"]}
    }',
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])
```

En savoir plus sur les événements de l'API de connexion aux comptes

Les sections suivantes décrivent les types d'événements de l'API de connexion aux comptes, les scénarios qui les déclenchent et des exemples d'événements.

Types d’événements

Vous trouverez ci-dessous les types d'événements et leurs déclencheurs.

- Invitation de connexion avec le partenaire reçue : indique au destinataire qu'un autre partenaire souhaite établir une connexion
- Invitation de connexion avec le partenaire acceptée : indique à l'expéditeur que son invitation a été acceptée et qu'une connexion est désormais active
- Invitation de connexion avec le partenaire rejetée : indique à l'expéditeur que son invitation a été refusée
- Annulation de la connexion avec le partenaire : avertit l'autre participant connecté lorsqu'une connexion active est interrompue

- Invitation de connexion avec le partenaire annulée : indique au destinataire que l'expéditeur a retiré son invitation avant que toute action ne soit prise
- Invitation de connexion avec le partenaire expirée : indique à l'expéditeur et au destinataire que l'invitation a expiré

Exemples d’événements

Les sections suivantes fournissent des exemples des événements répertoriés précédemment dans la section précédente.

Rubriques

- Invitation de connexion avec le partenaire reçue
- Invitation de connexion avec un partenaire acceptée
- Invitation de connexion avec un partenaire rejetée
- Connexion avec le partenaire annulée
- Invitation de connexion avec un partenaire annulée
- Invitation de connexion aux partenaires expirée

Invitation de connexion avec le partenaire reçue

Contexte de déclenchement

Cet événement est généré lorsqu'une invitation de connexion est créée avec succès et conservée dans le magasin de données du PAC via l' CreateConnectionInvitation API. L'événement est envoyé immédiatement une fois la création de l'invitation terminée avec succès et est conservé dans le magasin de données du PAC, et pas seulement lorsque l'appel d'API est effectué.

L'événement sera automatiquement envoyé lorsqu'une nouvelle invitation de connexion sera créée. Un événement par invitation créé, sans qu'il n'y ait d'événements dupliqués pour la même invitation.

Destinataires

Le AWS compte du destinataire dans l'invitation de connexion.

Manipulation attendue

Actions typiques du client :

- Notification immédiate : déclenchez des alertes pour les équipes commerciales concernant les nouvelles opportunités de partenariat
- Automatisation du flux de travail : mise à jour automatique des systèmes de gestion des partenaires avec les invitations en attente
- Support à la décision : acheminer les invitations aux décideurs appropriés en fonction du type de connexion
- Journalisation des audits : enregistrez la réception de l'invitation à des fins de conformité et de suivi des partenariats
- Automatisation des réponses : pour les partenaires de confiance, possibilité d'accepter automatiquement certains types d'invitations

Modèles d'intégration courants :

- Fonction Lambda → Mettre à jour le tableau de bord du portail partenaire
- File d'attente SQS → Invitations de traitement par lots pour révision
- Rubrique SNS → Email/Slack notifications aux équipes commerciales
- Destination de l'API → Webhook vers des systèmes de CRM/partner gestion externes

Exemple

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Received",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "account": "<corresponding partner AWS account>",
  "resources": [ "<<Connection Invitation ARN>>" ],
  "detail": {
    "catalog": "AWS",
    "connectionInvitation" :{
        "arn": "<<Connection Invitation ARN>>",
        "id": "pacinv-****",
        "connectionType": "OpportunityCollaboration",
        "inviterEmail": "abc@def.com",
        "invitationMessage": "We'd like to collaborate on a joint solution for cloud
 security. Please accept this connection invitation to proceed.",
```

```
        "senderCompanyName": "<<Sender Partner Account Name>>",
        "senderProfileId": "pprofile-****",
        "expiresAt": "<ISO 8601 date time>"
    }
  }
}
```

Invitation de connexion avec un partenaire acceptée

Contexte de déclenchement

Cet événement est généré lorsqu'une invitation de connexion est acceptée avec succès et qu'une connexion est créée et conservée dans le magasin de données du PAC via l' AcceptConnectionInvitation API. L'événement est envoyé immédiatement une fois que l'acceptation de l'invitation est terminée avec succès et que la connexion est établie.

L'événement sera automatiquement envoyé lorsqu'une invitation de connexion sera acceptée. Un événement par invitation accepté, aucun événement dupliqué pour la même acceptation.

Destinataires

Les deux AWS comptes concernés par l'invitation de connexion : le compte expéditeur qui a initialement envoyé l'invitation et le compte destinataire qui a accepté l'invitation.

Manipulation attendue

Actions typiques du client :

- Activation du partenariat : déclenchez des flux de travail pour activer les activités commerciales de niveau 2
- Mises à jour du statut : mise à jour des systèmes de gestion des partenaires avec l'état de connexion actif
- Notification : avertir les équipes commerciales de la réussite de l'établissement d'un partenariat
- Provisionnement des accès : accordez automatiquement les autorisations appropriées pour la collaboration
- Analyses : suivez les taux de conversion des partenaires et les indicateurs de réussite

Modèles d'intégration courants :

- Fonction Lambda → Activer les autorisations de partage de données et mettre à jour le portail des partenaires
- File d'attente SQS → Activations de connexions par processus par lots pour la configuration de l'entreprise
- Rubrique SNS → Email/Slack notifications aux équipes commerciales et techniques
- Destination de l'API → Webhook vers les systèmes CRM pour mettre à jour le statut du partenariat

Exemple

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type":  "Partner Connection Invitation Accepted",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "resources": [ "<<Connection Invitation ARN>>" , "<<Connection ARN>>" ],
  "account": "<corresponding partner AWS account>",
  "detail": {
    "catalog": "AWS",
    "connectionInvitation" :{
        "arn": "<<Connection Invitation ARN>>",
        "id": "pacinv-****",
        "connectionType": "OpportunityCollaboration"
    },
    "connection": {
        "arn": "<<Connection ARN>>",
         "id": "pac-*****",
         "Participant1AccountId": "<<Sender AWS AccountId>>",
         "Participant2AccountId": "<<Receiver AWS AccountId>>",
         "status": "ACTIVE"
     }
  }
}
```

Invitation de connexion avec un partenaire rejetée

Contexte de déclenchement

Cet événement est généré lorsqu'une invitation de connexion est rejetée avec succès via l' RejectConnectionInvitation API. L'événement est envoyé immédiatement après le traitement du rejet de l'invitation et est conservé dans le magasin de données du PAC.

L'événement sera automatiquement envoyé lorsqu'une invitation de connexion est rejetée. Un événement par invitation rejeté, aucun événement dupliqué pour le même rejet.

Destinataires

Le AWS compte qui a initialement envoyé l'invitation de connexion (le compte de l'expéditeur).

Manipulation attendue

Actions typiques du client :

- Mises à jour du statut : mettre à jour les systèmes de gestion des partenaires avec le statut de rejet
- Follow-up Actions : déclencher des flux de travail pour des approches de partenariat alternatives
- Notification : avertir les équipes commerciales de la décision de partenariat
- Nettoyage : supprimer les références d'invitation en attente des systèmes internes

Modèles d'intégration courants :

- Fonction Lambda → Mettre à jour le portail des partenaires et déclencher des flux de travail de suivi
- File d'attente SQS → Rejets de processus par lots pour l'analyse commerciale
- Rubrique SNS → Email/Slack notifications aux équipes de développement commercial
- Destination de l'API → Webhook aux systèmes CRM pour mettre à jour le statut des opportunités

Exemple

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Rejected",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
```

```
  "account": "<corresponding partner AWS account>",
  "resources": [ "<<Connection Invitation ARN>>" ],
  "detail": {
    "catalog": "AWS",
    "connectionInvitation" :{
        "arn": "<<Connection Invitation ARN>>",
        "id": "pacinv-****",
        "connectionType": "OpportunityCollaboration",
        "invitationMessage": "We'd like to collaborate on a joint solution for cloud
 security. Please accept this connection invitation to proceed.",
        "receiverProfileId": "pprofile-****"
    }
  }
}
```

Connexion avec le partenaire annulée

Contexte de déclenchement

Cet événement est généré lorsqu'une connexion active est annulée avec succès et que le statut actif est mis à jour pour être interrompu dans le magasin de données du PAC via l' CancelConnection API. L'événement est envoyé immédiatement après le traitement de l'annulation de la connexion et la mise à jour de l'état de la connexion.

L'événement sera automatiquement envoyé lorsqu'une connexion est interrompue. Un événement annulé par connexion, aucun événement dupliqué pour la même annulation.

Destinataires

Le AWS compte de l'autre participant à la connexion (et non celui qui a initié l'annulation).

Manipulation attendue

Actions typiques du client :

- Révocation d'accès : révoquez immédiatement les autorisations et désactivez le partage de données
- Mises à jour du statut : mettre à jour les systèmes de gestion des partenaires lorsque la connexion est interrompue
- Notification : avertir les équipes commerciales et techniques de la résiliation du partenariat
- Nettoyage : supprimer les références de connexion et nettoyer les ressources partagées
- Analyses : suivre la durée de la connexion et les modèles de terminaison

Modèles d'intégration courants :

- Fonction Lambda → Révoquer les autorisations et mettre à jour le statut du portail partenaire
- File d'attente SQS → Terminaisons de connexion aux processus Batch pour les flux de travail de nettoyage
- Rubrique SNS → Email/Slack notifications aux équipes commerciales et techniques
- Destination de l'API → Webhook vers les systèmes CRM pour mettre à jour le statut du partenariat

Exemple

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Cancelled",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "resources": [ "<<Connection ARN>>" ],
  "account": "<corresponding partner AWS account>",
  "detail": {
    "catalog": "AWS",
    "connection" :{
        "arn": "<<Connection Invitation ARN>>",
        "id": "pac-****",
        "connectionType": "OpportunityCollaboration",
        "canceledBy": "<<AWS account ID>>"
    }
  }
}
```

Invitation de connexion avec un partenaire annulée

Contexte de déclenchement

Cet événement est généré lorsqu'une invitation de connexion en attente est annulée avec succès via l' CancelConnectionInvitation API. L'événement est envoyé immédiatement après le traitement de l'annulation de l'invitation et le statut de l'invitation est passé à ANNULÉ.

L'événement sera automatiquement envoyé lorsqu'une invitation de connexion est annulée. Une invitation à un événement par connexion est annulée, sans qu'aucun événement ne soit dupliqué pour la même annulation.

Destinataires

Le AWS compte censé recevoir l'invitation de connexion (le compte du destinataire).

Manipulation attendue

Actions typiques du client :

- Mises à jour de statut : mettez à jour les systèmes de gestion des partenaires pour supprimer les références d'invitation en attente
- Notification : avertir les équipes commerciales qu'une opportunité de partenariat potentielle a été retirée
- Nettoyage : supprimer les références aux invitations des systèmes de suivi internes
- Analyses : suivez les modèles d'annulation des invitations pour obtenir des informations sur les partenariats

Modèles d'intégration courants :

- Fonction Lambda → Mettre à jour le portail des partenaires et supprimer les notifications d'invitation en attente
- File d'attente SQS → Annulations de processus par lots pour l'analyse commerciale
- Rubrique SNS → Email/Slack notifications aux équipes de développement commercial
- Destination de l'API → Webhook aux systèmes CRM pour mettre à jour le statut des opportunités

Exemple

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Cancelled",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "account": "<corresponding partner AWS account>",
  "resources": [ "<<Connection Invitation ARN>>" ],
  "detail": {
    "catalog": "AWS",
    "connectionInvitation" :{
```

```
        "arn": "<<Connection Invitation ARN>>",
        "id": "pacinv-****",
        "connectionType": "OpportunityCollaboration",
        "invitationMessage": "We'd like to collaborate on a joint solution for cloud
 security. Please accept this connection invitation to proceed.",
        "senderProfileId": "pprofile-****"
    }
  }
}
```

Invitation de connexion aux partenaires expirée

Contexte de déclenchement

Cet événement est généré lorsqu'une invitation de connexion en attente expire automatiquement après avoir atteint son ExpiresAt horodatage sans être acceptée ou rejetée par le destinataire. L'événement est automatiquement déclenché par le système lorsque le délai d'expiration de l'invitation est atteint.

L'événement sera automatiquement envoyé à l'expiration d'une invitation de connexion. Un événement par invitation a expiré sans qu'il n'y ait eu de double événement pour la même date d'expiration.

Destinataires

Les deux AWS comptes concernés par l'invitation de connexion : le compte expéditeur qui a initialement envoyé l'invitation et le compte destinataire qui a reçu l'invitation.

Manipulation attendue

Actions typiques du client :

- Mises à jour du statut : mettez à jour les systèmes de gestion des partenaires pour marquer l'invitation comme expirée
- Follow-up Actions : déclencher des flux de travail pour le réengagement ou des approches de partenariat alternatives
- Notification : avertir les équipes commerciales des opportunités de partenariat manquées
- Nettoyage : supprimer les références d'invitation expirées des systèmes de suivi internes

Modèles d'intégration courants :

- Fonction Lambda → Mettre à jour le portail des partenaires et déclencher des flux de travail de suivi
- File d'attente SQS → Expiration des processus par lots pour l'analyse commerciale
- Rubrique SNS → Email/Slack notifications aux équipes de développement commercial
- Destination de l'API → Webhook aux systèmes CRM pour mettre à jour le statut des opportunités

Exemple

```
{
  "version": "1",
  "id": "<event id>",
  "detail-type": "Partner Connection Invitation Expired",
  "source": "aws.partnercentral-account",
  "time": "<ISO 8601 date time>",
  "region": "us-east-1",
  "resources": [ "<<Connection Invitation ARN>>" ],
  "detail": {
    "catalog": "AWS",
    "connectionInvitation": {
        "arn": "<<Connection Invitation ARN>>",
        "id": "pacinv-****",
        "connectionType": "OpportunityCollaboration",
        "inviterEmail": "abc@def.com",
        "invitationMessage": "We'd like to collaborate on a joint solution for cloud
 security. Please accept this connection invitation to proceed.",
        "senderProfileId": "pprofile-****",
        "receiverProfileId": "pprofile-****"
    }
  }
}
```

Notifications pour AWS API de vente centralisée pour les partenaires

Les événements liés à l'API de vente fournissent des notifications en temps réel concernant les changements de statut ou les détails des opportunités. Ces événements permettent de synchroniser vos systèmes avec AWS Partner Central et de garantir des réponses et des mises à jour rapides.

Rubriques

- Remplissez le prérequis pour surveiller les événements
- Configurer Amazon EventBridge pour surveiller les événements

- En savoir plus sur la vente d'événements liés à l'API

Remplissez le prérequis pour surveiller les événements

Les utilisateurs doivent disposer des autorisations IAM appropriées pour accéder aux événements publiés par l'API de vente Partner Central et les gérer. Pour plus d'informations sur les actions, les ressources et les clés de condition disponibles pour EventBridge, consultez la section Utilisation des conditions de politique IAM sur Amazon EventBridge dans le guide de l' EventBridge utilisateur Amazon. L'une des clés de condition est `events:detail-type` celle qui peut être utilisée pour étendre les autorisations à des types d'événements spécifiques.

L'exemple de politique suivant montre comment personnaliser et étendre les autorisations pour les événements proposés. L'`AllowPutRuleForPartnercentralSellingEvents`instruction permet de créer des règles, mais uniquement pour les événements provenant de la `aws.partnercentral-selling` source.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Sid": "AllowPutRuleForPartnercentralSellingEvents",
    "Effect": "Allow",
    "Action": "events:PutRule",
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "events:source": "aws.partnercentral-selling"
      }
    }
  }
}
```

Configurer Amazon EventBridge pour surveiller les événements

Pour surveiller les événements de l'API de vente, vous créez une EventBridge règle qui correspond aux événements que vous souhaitez capturer. Vous pouvez utiliser le AWS Management Console ou les AWS SDK pour créer et gérer des règles. Les sections suivantes expliquent comment créer

des règles à l'aide des deux méthodes. Quelle que soit la méthode utilisée, vous devez créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

AWS Management Console configuration

Pour configurer une EventBridge règle à l'aide du AWS Management Console, suivez les étapes décrites dans la section Création de règles réagissant aux événements survenus sur Amazon EventBridge dans le guide de EventBridge l'utilisateur Amazon. Lors de la création de la règle, vous devez définir le bus d'événements par défaut et créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

Voici un exemple de règle d'événement :

```
{
    "source": ["aws.partnercentral-selling"],
    "detail": {
        "catalog": ["AWS"]
    }
}
```

AWS Configuration du SDK

Vous pouvez utiliser les AWS SDK pour créer et gérer des EventBridge règles par programmation. Pour plus d'informations, consultez PutRulele Amazon EventBridge API Reference.

L'exemple suivant utilise le AWS SDK pour Python (Boto3) :

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='MyOpportunityCreatedRule',
    EventPattern=
    '{
        "source": ["aws.partnercentral-selling"],
        "detail-type": ["Opportunity Created"],
        "detail": {"catalog": ["AWS"]}
    }',
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])
```

En savoir plus sur la vente d'événements liés à l'API

Les sections suivantes décrivent les types d'événements de l'API de vente, les scénarios qui les déclenchent et des exemples d'événements.

Types d’événements

Vous trouverez ci-dessous les types d'événements et leurs déclencheurs.

- Opportunité créée : déclenchée lorsqu'une nouvelle opportunité est créée.
- Opportunité mise à jour : déclenchée lorsqu'une opportunité (opportunité ou son résumé correspondant AWS) est mise à jour.
- Invitation d'engagement créée : déclenchée lorsqu'une invitation AWS de parrainage est créée.
- Invitation d'engagement acceptée : déclenchée lorsqu'un partenaire accepte une invitation d'AWS engagement, confirmant son intérêt à collaborer avec AWS cette opportunité.
- Invitation d'engagement rejetée : déclenchée lorsqu'un partenaire rejette une invitation d'AWS engagement.
- Invitation d'engagement expirée : Cet événement est déclenché lorsqu'un partenaire rejette un EngagementInvitation. Il indique au partenaire expéditeur et destinataire que l'invitation a expiré.
- Membre d'engagement ajouté : Cet événement est déclenché lorsqu'un nouveau membre rejoint l'engagement après avoir accepté une invitation. Il informe tous les membres actuels de l'engagement de l'ajout du nouveau membre à l'engagement.
- Instantané des ressources d'engagement créé : Cet événement est déclenché lorsque la nouvelle révision de l'aperçu des ressources (telles que les opportunités) associées à un engagement est créée. Il informe tous les membres actuels de l'engagement des modifications apportées à la ressource Snapshot associée.
- Engagement créé : déclenché lorsqu'un nouvel engagement est créé.
- Engagement mis à jour : déclenché lorsqu'un engagement est mis à jour.

Scénarios d'événements

Le tableau suivant décrit le fonctionnement des événements dans différents scénarios. Les hypothèses suivantes sont formulées pour ces scénarios :

- AWS est également un partenaire dans ces scénarios.
- `ResourceSnapshotJob`est correctement configuré par le partenaire.

- Les champs mis à jour sur l'opportunité figurent également sur le modèle d'opportunité pour Resource Snapshot.

Action de votre part en tant que partenaire	Événements reçus	Type de participant
Vous créez une opportunité	Opportunité créée	
Vous utilisez `StartEngagementFromOpportunityTask` pour soumettre une opportunité	Opportunité mise à jour, engagement créé, aperçu des ressources d'engagement créé, membre d'engagement ajouté, invitation d'engagement créée	
AWS approuve l'opportunité que vous avez soumise	Invitation d'engagement acceptée, ajout d'un membre à l'engagement, opportunité mise à jour, création d'un instantané des ressources d'engagement	
AWS rejette l'opportunité que vous avez soumise	Invitation d'engagement rejetée, opportunité mise à jour, création d'un instantané des ressources d'engagement	
Vous associez une AWS Marketplace offre à une opportunité	Opportunité actualisée	
Vous associez une solution à une opportunité	Opportunité mise à jour, création d'un aperçu des ressources d'engagement	
Vous mettez à jour une opportunité	Opportunité mise à jour, création d'un instantané des ressources d'engagem	

Action de votre part en tant que partenaire	Événements reçus	Type de participant
	ent (facultatif si cette option `ResourceSnapshotJob` est configurée et que les mises à jour des champs figurent sur le modèle)	
AWS met à jour votre opportunité	Opportunité mise à jour, création d'un aperçu des ressources d'engagement	
Vous invitez un autre partenaire à un engagement	Invitation d'engagement créée	Expéditeur
Votre invitation à participer à un engagement est acceptée par un partenaire	Invitation d'engagement acceptée, membre d'engagement ajouté	Expéditeur
Votre invitation à participer à un engagement est rejetée par un partenaire	Invitation d'engagement rejetée	Expéditeur
Votre invitation à participer à un engagement n'est pas traitée par un partenaire pendant 15 jours	Invitation d'engagement expirée	Expéditeur
Vous recevez une invitation d'un autre partenaire à participer à un engagement	Invitation d'engagement créée	Récepteur
Vous acceptez l'invitation d'un autre partenaire à participer à un engagement via `StartEngagementByAcceptingInvitationTask`	Invitation d'engagement acceptée, ajout d'un membre à l'engagement, opportunité créée, opportunité mise à jour, création d'un instantané des ressources d'engagement	Récepteur

Action de votre part en tant que partenaire	Événements reçus	Type de participant
Vous rejetez l'invitation d'un autre partenaire à participer à un engagement	Invitation d'engagement rejetée	Récepteur
Vous ne répondez pas à une invitation à participer à un engagement d'un autre partenaire pendant 15 jours	Invitation d'engagement expirée	Récepteur
Vous créez un engagement	Engagement créé	Expéditeur
Vous mettez à jour un engagement	Engagement mis à jour	Expéditeur, destinataire

Exemples d’événements

Les sections suivantes fournissent des exemples des événements répertoriés précédemment dans la section précédente.

Rubriques

- Opportunité créée
- Opportunité actualisée
- Invitation d'engagement créée
- Invitation d'engagement acceptée
- Invitation d'engagement rejetée
- Invitation d'engagement expirée
- Membre d'engagement ajouté
- Création d'un aperçu des ressources d'engagement
- Engagement créé
- Engagement mis à jour

Opportunité créée

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs qu'une nouvelle opportunité a été créée.
- Déclenchez des flux de travail automatisés tels que la mise à jour des systèmes CRM ou la notification aux équipes commerciales de la création de nouvelles opportunités.

L'exemple suivant montre un événement Opportunity Created typique.

```
{
    "version": "1",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "source": "aws.partnercentral-selling",
    "detail-type": "Opportunity Created",
    "time": "2023-04-16T15:23:45Z",
    "region": "us-east-1",
    "account": "123456789012",
    "detail": {
        "schemaVersion": "<version number>",
        "catalog": "<Sandbox | AWS>",
        "opportunity": {
            "identifier": "String"
        }
    }
}
```

Opportunité actualisée

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs qu'une opportunité existante a été mise à jour.
- Déclenchez des flux de travail automatisés tels que la mise à jour des systèmes CRM ou la notification des équipes commerciales.

L'exemple suivant montre un événement Opportunity Updated typique.

```
{
    "version": "1",
    "id": "01234567-0123-0123-0123-0123456789ab",
```

```
    "source": "aws.partnercentral-selling",
    "detail-type": "Opportunity Updated",
    "time": "2023-04-16T15:23:45Z",
    "region": "us-east-1",
    "account": "123456789012",
    "detail": {
        "schemaVersion": "<version number>",
        "catalog": "<Sandbox | AWS>",
        "opportunity": {
             "identifier": "String"
        }
    }
}
```

Invitation d'engagement créée

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs des nouvelles EngagementInvitation qu'ils ont reçues.
- Déclenchez des flux de travail automatisés pour accepter ou rejeter l'invitation, par exemple en mettant à jour les systèmes CRM ou en informant les équipes commerciales.

L'exemple suivant montre un événement typique créé par invitation d'engagement.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Invitation Created",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-15T12:34:56Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-
v7p8z56whnauo"
    ],
    "detail": {
      "catalog": "AWS",
      "engagementInvitation": {
        "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/
engi-v7p8z56whnauo",
        "id": "engi-v7p8z56whnauo",
```

```
        "engagementId": "eng-12345678901234",
        "senderAccountId": "string",
        "receiverAccountId": "string",
        "senderCompanyName": "string",
        "expirationDate": "string",
        "participantType": "Enum", //Sender/Receiver
  "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
      }
    }
}
```

Invitation d'engagement acceptée

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs que le EngagementInvitation a été accepté.
- Mettez à jour leurs dossiers internes pour tenir compte du nouveau partenaire dans l'engagement.
- Déclenchez des flux de travail automatisés pour synchroniser les données ou informer les autres membres de l'équipe de l'arrivée du nouveau membre Engagement.

L'exemple suivant montre un événement type d'invitation d'engagement acceptée.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Invitation Accepted",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-16T15:23:45Z",
    "region": "us-east-1",
    "resources": ["arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/
engi-v7p8z56whnauo"],
    "detail": {
        "catalog": "AWS",
        "engagementInvitation": {
            "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-
invitation/engi-v7p8z56whnauo",
            "id": "engi-v7p8z56whnauo",
            "engagementId": "eng-12356whnauo",
            "participantType": "Enum", //Sender/Receiver
            "senderAccountId": "string",
```

```
            "receiverAccountId": "string",
            "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
        }
    }
}
```

Invitation d'engagement rejetée

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs que le EngagementInvitation message a été rejeté.
- Mettez à jour leurs dossiers internes pour refléter l'invitation rejetée.
- Déclenchez tous les flux de travail nécessaires, par exemple en informant l'équipe commerciale du rejet.

L'exemple suivant montre un événement typique d'invitation à un engagement rejeté.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Invitation Rejected",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-17T09:48:27Z",
    "region": "us-east-1",
    "resources": ["arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/
engi-v7p8z56whnauo"],
    "detail": {
        "catalog": "AWS",
        "engagementInvitation": {
            "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-
invitation/engi-v7p8z56whnauo",
            "id": "engi-v7p8z56whnauo",
            "engagementId": "eng-12356whnauo",
            "participantType": "Enum", //Sender/Receiver
            "senderAccountId": "string",
            "receiverAccountId": "string",
            "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
        }
    }
}
```

Invitation d'engagement expirée

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs que le délai EngagementInvitation a expiré et qu'il n'est plus possible d'y donner suite.
- Mettez à jour leurs dossiers internes pour refléter l'expiration de l'invitation.
- Déclenchez tous les flux de travail nécessaires, par exemple en informant l'équipe commerciale de l'expiration de l'invitation.

L'exemple suivant montre un événement typique d'une invitation d'engagement expirée.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Invitation Expired",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-18T18:20:15Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/engi-
v7p8z56whnauo"
    ],
    "detail": {
      "catalog" : "AWS",
      "engagementInvitation": {
        "arn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement-invitation/
engi-v7p8z56whnauo",
        "id": "engi-v7p8z56whnauo",
        "engagementId": "eng-12356whnauo",
        "participantType": "Enum", //Sender/Receiver
        "senderAccountId": "string",
        "receiverAccountId": "string",
        "payloadType": "LeadInvitation" //OpportunityInvitation|LeadInvitation
      }
    }
}
```

Membre d'engagement ajouté

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs des modifications apportées à l'adhésion à Engagement.
- Mettez à jour leurs dossiers internes pour refléter le nouveau membre d'Engagement.
- Déclenchez tous les flux de travail nécessaires, par exemple en informant les membres de l'équipe des modifications.

L'exemple suivant montre un événement typique lié à l'ajout d'un membre Engagement.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Member Added",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-16T15:23:45Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-v7p8z56whnauo"
    ],
    "detail": {
      "catalog" : "AWS",
      "engagement": {
        "id": "eng-v7p8z56whnauo"
      },
      "engagementMember": {
        "accountId": "string",
        "companyName": "string"
      }
    }
}
```

Création d'un aperçu des ressources d'engagement

Les partenaires utilisent cet événement pour suivre les modifications apportées aux ressources associées à un engagement et déclencher des flux de travail automatisés, tels que la mise à jour des systèmes CRM ou la notification des équipes commerciales. Le modèle de capture d'écran détermine le type de mise à jour :

`OpportunitySummaryView`

- Informez tous les membres d'un engagement que celui-ci a été mis à jour.

`AwsOpportunitySummaryFullView`

- Suivez les mises à jour effectuées spécifiquement par AWS concernant les opportunités dans le cadre d'un engagement.
- Cette notification et cet instantané ne sont visibles que par le propriétaire de l'opportunité.
- Inclut les mises à jour du dimensionnement des offres, qui ne sont pas disponibles via le `OpportunitySummaryView` modèle.

L'exemple suivant montre un événement typique créé par Engagement Resource Snapshot.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Resource Snapshot Created",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-18T18:20:15Z",
    "region": "us-east-1",
    "resources": [
      "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-v7p8z56whnauo"
    ],
    "detail": {
      "catalog" : "AWS",
      "resourceSnapshot": {
        "arn": "arn:aws:partnercentral-selling:us-east-1::catalog/AWS/engagement/eng-
v7p8z56whnauo/resource/Opportunity/o-12312/template/temp-name/snapshot/snapshot-19232",
        "engagementId": "eng-v7p8z56whnauo",
        "resourceType": "Opportunity",
        "resourceId" : "O123211231",
        "createdBy": "123123123123",
        "targetMemberAccounts": ["123123123123", "aws"],
        "resourceSnapshotTemplateName": "temp-name"
      }
    }
}
```

Engagement créé

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs qu'un nouvel engagement a été créé.
- Déclenchez des flux de travail automatisés tels que la mise à jour des systèmes CRM ou la notification aux équipes commerciales des nouvelles créations d'engagement.

L'exemple suivant montre un événement typique d'Engagement Created.


```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Created",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-18T18:20:15Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp35lo"
    ],
    "detail": {
        "catalog": "AWS",
        "engagement": {
            "engagementArn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/
eng-567a1j5hxp35lo",
            "engagementId": "eng-567a1j5hxp35lo",
            "CreatedAt": "2023-04-18T18:20:15Z",
            "CreatedBy": "aws",
            "ContextTypes": ["Lead"]

        }
    }
}
```


Engagement mis à jour

Les partenaires profitent de cet événement pour :

- Informez leurs utilisateurs qu'un engagement existant a été mis à jour.
- Déclenchez des flux de travail automatisés tels que la mise à jour des systèmes CRM ou la notification des équipes commerciales.

L'exemple suivant montre un événement typique d'Engagement Updated.

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Engagement Updated",
    "source": "aws.partnercentral-selling",
    "account": "123456789012",
    "time": "2023-04-18T18:20:15Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp35lo"
    ],
    "detail": {
        "catalog": "AWS",
        "engagement": {
            "engagementArn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/
eng-567a1j5hxp35lo",
            "engagementId": "eng-567a1j5hxp35lo",
            "LastModifiedAt": "2023-04-18T18:20:15Z",
            "LastModifiedBy": "aws",
            "ContextTypes": ["Lead", "CustomerProject"]
        }
    }
}
```

Notifications pour AWS API Partner Central Benefits

Le service des avantages sociaux propose EventBridge des événements pour les deux BenefitApplications et BenefitAllocations. Ces événements permettent aux clients de créer des architectures complètes axées sur les événements qui répondent aux changements tout au long du cycle de vie des ressources, de l'application initiale à l'allocation et à l'exécution finale.

Rubriques

- Remplissez le prérequis pour surveiller les événements
- Configurer Amazon EventBridge pour surveiller les événements
- En savoir plus sur les avantages et les événements liés à l'API

Remplissez le prérequis pour surveiller les événements

Les utilisateurs doivent disposer des autorisations IAM appropriées pour accéder aux événements publiés par l'API Benefits et les gérer. Pour plus d'informations sur les actions, les ressources et

les clés de condition disponibles pour EventBridge, consultez la section Utilisation des conditions de politique IAM sur Amazon EventBridge dans le guide de l' EventBridge utilisateur Amazon. L'une des clés de condition est `events:detail-type` celle qui peut être utilisée pour étendre les autorisations à des types d'événements spécifiques.

L'exemple de politique suivant montre comment personnaliser et définir l'étendue des autorisations pour les événements proposés. L'`AllowPutRuleForPartnercentralBenefitsEvents`instruction permet de créer des règles, mais uniquement pour les événements provenant de la `aws.partnercentral-benefits` source.

Pour obtenir des exemples détaillés de politique IAM, consultez la documentation AWS sur EventBridge les autorisations.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralBenefitsEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-benefits"
        }
      }
    }
  ]
}
```

Configurer Amazon EventBridge pour surveiller les événements

Pour surveiller les événements de l'API des avantages, vous créez une EventBridge règle qui correspond aux événements que vous souhaitez capturer. Vous pouvez utiliser le AWS Management Console ou les AWS SDK pour créer et gérer des règles. Les sections suivantes expliquent comment créer des règles à l'aide des deux méthodes. Quelle que soit la méthode que vous utilisez, vous devez créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

AWS Management Console configuration

Pour configurer une EventBridge règle à l'aide du AWS Management Console, suivez les étapes décrites dans la section Création de règles réagissant aux événements survenus sur Amazon EventBridge dans le guide de EventBridge l'utilisateur Amazon. Lors de la création de la règle, vous devez définir le bus d'événements par défaut et créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

Voici un exemple de règle d'événement :

```
{
    "source": ["aws.partnercentral-benefits"],
    "detail": {
        "catalog": ["AWS"]
    }
}
```

AWS Configuration du SDK

Vous pouvez utiliser les AWS SDK pour créer et gérer des EventBridge règles par programmation. Pour plus d'informations, consultez PutRulele Amazon EventBridge API Reference.

L'exemple suivant utilise le AWS SDK pour Python (Boto3) :

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='MyBenefitApplicationCreatedRule',
    EventPattern=
    '{
        "source": ["aws.partnercentral-benefits"],
        "detail-type": ["Benefit Application Created"],
        "detail": {"catalog": ["AWS"]}
    }',
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])
```

En savoir plus sur les avantages et les événements liés à l'API

Les sections suivantes décrivent les avantages, les types d'événements de l'API, les scénarios qui les déclenchent et des exemples d'événements.

Types d’événements

Vous trouverez ci-dessous les types d'événements et leurs déclencheurs.

Demandes de prestations

- Demande de prestations créée : Quand une demande de prestations est créée
- Demande de prestations en cours d'examen : lorsqu'une demande de prestations est soumise ou lorsque la demande passe du statut Action requise à En cours d'examen, cet événement est déclenché.
- Action requise pour la demande de prestations : Cet événement se produit lorsqu'un réviseur interne met à jour le statut de la demande pour indiquer que le partenaire doit fournir des informations supplémentaires, des éclaircissements ou des modifications avant que l'examen ne puisse se poursuivre.
- Demande d'avantages rejetée : cet événement se produit lorsqu'un réviseur interne met à jour le statut de la demande pour indiquer que la demande ne répond pas aux critères ou aux exigences en matière de prestations et qu'elle a été refusée
- Demande d'avantages approuvée : Cet événement se produit lorsqu'un réviseur interne met à jour le statut de la demande pour indiquer que la demande a été approuvée pour l'attribution des avantages.
- Modification de l'étape de la demande de prestations : Cet événement se produit lorsqu'un réviseur interne met à jour l'étape de la demande au cours de processus de révision tels que l'examen commercial, l'examen, l'examen financier, etc.AWS

Répartition des avantages

- Allocation d'avantages activée : lorsqu'une allocation d'avantages est créée ou mise à jour pour revenir à Activée par un service interne.
- Allocation d'avantages désactivée : Cet événement se produit lorsqu'un réviseur interne met à jour le statut d'allocation ou lorsqu'une allocation d'avantages expire automatiquement.
- Affectation des avantages exécutée : Cet événement se produit lorsqu'un réviseur interne met à jour le statut de l'allocation pour indiquer que l'allocation a été entièrement utilisée.

Exemples d’événements

Les sections suivantes fournissent des exemples des événements répertoriés précédemment dans la section précédente.

Rubriques

- Demande de prestations créée
- Demande de prestations en cours d'examen
- Action requise pour présenter une demande de prestations
- Demande de prestations rejetée
- Demande de prestations approuvée
- Modification de l'étape de demande de prestations
- Affectation des avantages activée
- Répartition des avantages désactivée
- Répartition des avantages réalisée

Demande de prestations créée

Contexte de déclenchement

L'événement Benefit Application Created est publié sur Amazon EventBridge lorsqu'une nouvelle demande de prestations est créée avec succès via le Benefits Service. Cet événement se produit lorsque l'appel d' CreateBenefitApplication API aboutit, indiquant qu'un partenaire a lancé une demande pour un avantage spécifique.

Cet événement marque le début du cycle de vie des demandes d'avantages sociaux et fournit aux clients une notification immédiate lorsque de nouvelles demandes sont soumises sur leur compte.

Destinataires

Le compte propriétaire recevra l'événement pour l'allocation des avantages.

Exemple


```
{
  "id": "7gh88765-k0jh-d08g-i292-2ff1796m030n",
  "detail-type": "Benefit Application Created",
  "source": "aws.partnercentral-benefits",
```

```
  "account": "123456789012",
  "time": "2025-02-10T15:45:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "PENDING_SUBMISSION",
      "revision": "hh7e8400-e29b-41d4-a716-446655440012"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Demande de prestations en cours d'examen

Contexte de déclenchement

L'événement Benefit Application In Review est publié sur Amazon EventBridge lorsqu'une demande d'avantages passe au statut IN_REVIEW via le service Benefits. Cet événement se produit lorsque :

- Un partenaire soumet sa candidature via l'appel SubmitBenefitApplication d'API
- Une application passe de ACTION_REQUIRED à IN_REVIEW une fois que le partenaire a répondu aux commentaires
- Une demande est rappelée puis soumise à nouveau

Cet événement indique que la demande est entrée dans le processus d'examen officiel et qu'elle est en cours d'évaluation par des équipes AWS internes.

Destinataires

Le compte propriétaire recevra l'événement pour l'allocation des avantages.

Exemple


```
{
  "id": "8hi99876-l1ki-e19h-j303-3gg2807n141o",
  "detail-type": "Benefit Application In Review",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-10T16:00:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "IN_REVIEW",
      "stage": "BUSINESS_REVIEW",
      "revision": "ii8f9511-f30c-52e5-b827-557766551123"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```


Action requise pour présenter une demande de prestations

Contexte de déclenchement

L'événement Action Required Benefit Application est publié sur Amazon EventBridge lorsqu'une demande de prestations passe au statut ACTION_REQUIRED via le service Benefits. Cet événement se produit lorsqu'un réviseur interne met à jour le statut de la demande via l' UpdateBenefitApplicationInternal API pour indiquer que le partenaire doit fournir des informations supplémentaires, des éclaircissements ou des modifications avant que l'examen ne puisse se poursuivre.

Cet événement représente un point critique du cycle de vie de la demande où l'intervention d'un partenaire est nécessaire pour faire avancer la demande dans le processus d'évaluation.

Destinataires

Le compte propriétaire recevra l'événement pour l'allocation des avantages

Exemple


```
{
  "id": "9ij00987-m2lj-f20i-k414-4hh3918o252p",
  "detail-type": "Benefit Application Action Required",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-12T10:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "ACTION_REQUIRED",
      "stage": "BUSINESS_REVIEW",
      "revision": "jj9g0622-g41d-63f6-c938-668877662234"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```


Demande de prestations rejetée

Contexte de déclenchement

L'événement Rejeté de la demande de prestations est publié sur Amazon EventBridge lorsqu'une demande d'avantages passe au statut REJETÉE via le Service des avantages.

Cet événement se produit lorsqu'un réviseur interne met à jour le statut de la demande via l' UpdateBenefitApplicationInternal API pour indiquer que la demande ne répond pas aux critères ou aux exigences en matière d'avantages et qu'elle a été refusée.

Cet événement représente un état terminal du cycle de vie de l'application, dans lequel celle-ci a été officiellement refusée et ne sera pas affectée à l'allocation des avantages.

Destinataires

Le compte propriétaire recevra l'événement pour l'allocation des avantages.

Exemple

```
{
  "id": "0kl11098-n3mk-g31j-l525-5ii4029p363q",
  "detail-type": "Benefit Application Rejected",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-15T14:20:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "REJECTED",
      "revision": "kk0h1733-h52e-74g7-d049-779988773345"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Demande de prestations approuvée

Contexte de déclenchement

L'événement Approved Benefits Application est publié sur Amazon EventBridge lorsqu'une demande d'avantages passe au statut APPROUVÉE via le Service des avantages. Cet événement se produit lorsqu'un réviseur interne met à jour le statut de la demande via l' UpdateBenefitApplicationInternal API pour indiquer que la demande répond à tous les critères et exigences en matière d'avantages et que l'attribution des avantages a été approuvée.

Cet événement représente la réussite du processus d'examen des demandes et déclenche généralement la création d'une allocation d'avantages correspondante.

Destinataires

Le compte propriétaire recevra l'événement pour l'allocation des avantages.

Exemple

```
{
  "id": "1lm22109-o4nl-h42k-m636-6jj5130q474r",
  "detail-type": "Benefit Application Approved",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-14T16:45:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "APPROVED",
      "revision": "ll1i2844-i63f-85h8-e150-880099884456"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
```

```
  }
}
```

Modification de l'étape de demande de prestations

Contexte de déclenchement

L'événement Benefit Application Stage Changed est publié sur Amazon EventBridge lorsque l'étape de révision d'une demande de prestations est mise à jour via le service Benefits. Cet événement se produit lorsqu'un réviseur interne met à jour le stade de l'application avec l'une des valeurs Enum possibles (FINANCE_REVIEW, BUSINESS_REVIEW, TECH_REVIEW et _REVIEW).AWS

Contrairement aux changements de statut, les changements d'étape sont des mises à jour informatives en lecture seule qui fournissent une visibilité sur la progression du flux de travail de révision interne sans que l'intervention du partenaire ne soit nécessaire.

Destinataires

Le compte propriétaire recevra l'événement pour l'allocation des avantages.

Exemple

```
{
  "id": "2mn33210-p5om-i53l-n747-7kk6241r585s",
  "detail-type": "Benefit Application Stage Changed",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-02-13T11:15:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-application/
benappl-12345abcdef123"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitApplication": {
      "id": "benappl-12345abcdef123",
      "fulfillmentTypes": ["CREDITS"],
      "status": "IN_REVIEW",
      "stage": "FINANCIAL_REVIEW",
      "revision": "mm2j3955-j74g-96i9-f261-991100995567"
    },
    "benefit": {
```

```
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Affectation des avantages activée

Contexte de déclenchement

L'événement Benefit Allocation Activated est publié sur Amazon EventBridge lorsqu'une allocation d'avantages passe au statut ACTIF via le service Benefits. Cet événement se produit lorsqu'un réviseur interne met à jour le statut de l'allocation via l' UpdateBenefitAllocationInternal API pour réactiver une allocation précédemment inactive.

Cet événement se produit généralement lorsqu'une allocation temporairement inactivée (en raison d'une expiration, de modifications de politique ou pour des raisons administratives) redevient active, ce qui permet aux partenaires d'utiliser à nouveau les avantages.

Destinataires

Le AWS compte appartenant au partenaire recevra le message. En d'autres termes, le AWS compte propriétaire de la ressource.

Exemple d'événement

```
{
  "id": "3no44321-q6pn-j64m-o858-8ll7352s696t",
  "detail-type": "Benefit Allocation Activated",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-03-01T08:15:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-allocation/benalloc-
abc123def456"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitAllocation": {
      "id": "benalloc-abc123def456",
      "fulfillmentType": "CREDITS",
```

```
      "status": "ACTIVE"
    },
    "benefitApplication": {
      "id": "benappl-12345abcdef123"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Répartition des avantages désactivée

Contexte de déclenchement

L'événement Inactivation de l'allocation d'avantages est publié sur Amazon EventBridge lorsqu'une allocation d'avantages passe au statut INACTIF via le service Benefits. Cet événement se produit lorsqu'un réviseur interne met à jour le statut de l'allocation via l' UpdateBenefitAllocationInternal API ou lorsqu'une allocation d'avantages expire automatiquement.

Cet événement se produit généralement lorsqu'une allocation est temporairement suspendue (en raison d'une expiration, de modifications de politique, de problèmes de conformité ou pour des raisons administratives), rendant les avantages inutilisables par les partenaires jusqu'à ce qu'ils soient réactivés.

Destinataires

Le AWS compte appartenant au partenaire recevra le message. En d'autres termes, le AWS compte propriétaire de la ressource.

Exemple d'événement

```
{
  "id": "4op55432-r7qo-k75n-p969-9mm8463t707u",
  "detail-type": "Benefit Allocation Inactivated",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-06-30T23:59:59Z",
  "region": "us-east-1",
  "resources": [
```

```
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-allocation/benalloc-
abc123def456"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitAllocation": {
      "id": "benalloc-abc123def456",
      "fulfillmentType": "CREDITS",
      "status": "INACTIVE"
    },
    "benefitApplication": {
      "id": "benappl-12345abcdef123"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Répartition des avantages réalisée

Contexte de déclenchement

L'événement Benefit Allocation Fulfilled est publié sur Amazon EventBridge lorsqu'une allocation d'avantages passe au statut FULFILLED via le service Benefits. Cet événement se produit lorsqu'un réviseur interne met à jour le statut de l'allocation via l' UpdateBenefitAllocationInternal API pour indiquer que l'allocation a été complètement utilisée ou que les conditions des avantages ont été satisfaites.

Cet événement représente la fin du cycle de vie des avantages, indiquant que le partenaire a utilisé avec succès l'avantage alloué ou que l'allocation a atteint son terme naturel.

Destinataires

Le AWS compte appartenant au partenaire recevra le message. En d'autres termes, le AWS compte propriétaire de la ressource.

Exemple d'événement

```
{
  "id": "5pq66543-s8rp-l86o-q070-0nn9574u818v",
```

```
  "detail-type": "Benefit Allocation Fulfilled",
  "source": "aws.partnercentral-benefits",
  "account": "123456789012",
  "time": "2025-12-15T17:30:00Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:partnercentral:us-east-1:123456789012:benefit-allocation/benalloc-
abc123def456"
  ],
  "detail": {
    "catalog": "AWS",
    "benefitAllocation": {
      "id": "benalloc-abc123def456",
      "fulfillmentType": "CREDITS",
      "status": "FULFILLED"
    },
    "benefitApplication": {
      "id": "benappl-12345abcdef123"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

Notifications pour AWS API Partner Central Channel

Les événements liés à l'API du canal fournissent des notifications en temps réel concernant les changements de statut ou les détails des activités liées au canal. Ces événements permettent de synchroniser vos systèmes avec AWS Partner Central et de garantir des réponses et des mises à jour rapides.

Conditions préalables

Vous devez disposer des autorisations IAM appropriées pour accéder aux événements et les gérer à partir de l'API AWS Partner Central Channel. Pour plus d'informations sur les actions, les ressources et les clés de condition disponibles pour EventBridge, consultez la section Utilisation des conditions de politique IAM sur Amazon EventBridge dans le guide de l' EventBridge utilisateur Amazon. L'une des clés de condition est `events:detail-type` que vous pouvez utiliser pour étendre les autorisations à des types d'événements spécifiques.

L'exemple de politique suivant montre comment personnaliser et définir les autorisations pour les événements. L'`AllowPutRuleForPartnercentralChannelEvents`instruction permet de créer des règles, mais uniquement pour les événements provenant de la `aws.partnercentral-channel` source.

Pour obtenir des exemples détaillés de politique IAM, consultez la documentation AWS sur EventBridge les autorisations.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralChannelEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-channel"
        }
      }
    }
  ]
}
```

Configurer Amazon EventBridge pour surveiller les événements

Pour surveiller les événements de l'API du canal, vous créez une EventBridge règle qui correspond aux événements que vous souhaitez capturer. Vous pouvez utiliser le AWS Management Console ou les AWS SDK pour créer et gérer des règles. Les sections suivantes expliquent comment créer des règles à l'aide des deux méthodes. Quelle que soit la méthode que vous utilisez, vous devez créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

AWS Management Console configuration

Pour configurer une EventBridge règle à l'aide du AWS Management Console, suivez les étapes décrites dans la section Création de règles réagissant aux événements survenus sur Amazon EventBridge dans le guide de EventBridge l'utilisateur Amazon. Lors de la création de la règle, vous devez définir le bus d'événements par défaut et créer la règle dans la `us-east-1` région USA Est (Virginie du Nord).

Voici un exemple de règle d'événement :

```
{
    "source": ["aws.partnercentral-channel"],
    "detail": {
        "catalog": ["AWS"]
    }
}
```

AWS Configuration du SDK

Vous pouvez utiliser les AWS SDK pour créer et gérer des EventBridge règles par programmation. Pour plus d'informations, consultez PutRulele Amazon EventBridge API Reference.

L'exemple suivant utilise le AWS SDK pour Python (Boto3) :

```
import boto3

client = boto3.client('events', region_name='us-east-1')

response = client.put_rule(
    Name='ChannelHandshakeCreatedRule',
    EventPattern=
    '{
        "source": ["aws.partnercentral-channel"],
        "detail-type": ["Channel Handshake Created"],
        "detail": {"catalog": ["AWS"]}
    }',
    State='ENABLED'
)
print('Rule ARN:', response['RuleArn'])
```

En savoir plus sur les événements liés à l'API des chaînes

Les sections suivantes décrivent les types d'événements de l'API du canal, les scénarios qui les déclenchent et des exemples d'événements.

Types d’événements

Vous trouverez ci-dessous les types d'événements et leurs déclencheurs.

- Poignée de chaîne créée : déclenchée lorsqu'une nouvelle poignée de main de chaîne est créée.

- Poignée de main de chaîne acceptée : déclenchée lorsqu'une nouvelle poignée de main de chaîne est acceptée.
- Poignée de main de chaîne rejetée : déclenchée lorsqu'une nouvelle poignée de main de chaîne est rejetée.

Chaîne Handshake créée

Les exemples suivants montrent des événements Channel Handshake Created typiques pour différents types de poignée de main.

Poignée de main au début de la période de service :


```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Created",
    "source": "aws.partnercentral-channel",
    "account": "789789789789",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-abc123def456g"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "rs-jkl012mno345p",
        "handshakeType": "START_SERVICE_PERIOD",
        "id": "ch-abc123def456g",
        "receiverAccountId": "789789789789",
        "ownerAccountId": "123123123123",
        "senderAccountId": "456456456456",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "startServicePeriodHandshakeDetail": {
                "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
                "minimumNoticeDays": "14",
                "endDate": null,
                "startDate": "2025-02-02T00:00:00Z",
                "note": "Test note example"
            }
```

```
        }
    }
}
```

Annulation de la période de service Handshake :

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Created",
    "source": "aws.partnercentral-channel",
    "account": "789789789789",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-def456ghi789j"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "rs-jkl012mno345p",
        "handshakeType": "REVOKE_SERVICE_PERIOD",
        "id": "ch-def456ghi789j",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "789789789789",
        "senderAccountId": "456456456456",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "revokeServicePeriodHandshakeDetail": {
                "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
                "minimumNoticeDays": "14",
                "endDate": null,
                "startDate": "2025-02-02T00:00:00Z",
                "note": "Test note example"
            }
        }
    }
}
```

Poignée de main concernant le compte de gestion du programme :

```
{
```

```
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Created",
    "source": "aws.partnercentral-channel",
    "account": "456456456456",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-ghi789jkl012m"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "pma-abc123def456g",
        "handshakeType": "PROGRAM_MANAGEMENT_ACCOUNT",
        "id": "ch-ghi789jkl012m",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "456456456456",
        "senderAccountId": "123123123123",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "programManagementAccountHandshakeDetail": {
                "program": "SOLUTION_PROVIDER"
            }
        }
    }
}
```

Poignée de main de chaîne acceptée

Les exemples suivants montrent des événements Channel Handshake Accepted typiques pour différents types de poignée de main.

Poignée de main au début de la période de service :

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Accepted",
    "source": "aws.partnercentral-channel",
    "account": "123123123123",
    "time": "2025-02-01T00:00:00Z",
```

```
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-abc123def456g"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "rs-jkl012mno345p",
        "handshakeType": "START_SERVICE_PERIOD",
        "id": "ch-abc123def456g",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "789789789789",
        "senderAccountId": "456456456456",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "startServicePeriodHandshakeDetail": {
                "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
                "minimumNoticeDays": "14",
                "endDate": null,
                "startDate": "2025-02-02T00:00:00Z",
                "note": "Test note example"
            }
        }
    }
}
```

Annulation de la période de service Handshake :

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Accepted",
    "source": "aws.partnercentral-channel",
    "account": "123123123123",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-def456ghi789j"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
```

```
        "catalog": "AWS",
        "associatedResourceIdentifier": "rs-jkl012mno345p",
        "handshakeType": "REVOKE_SERVICE_PERIOD",
        "id": "ch-def456ghi789j",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "789789789789",
        "senderAccountId": "456456456456",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "revokeServicePeriodHandshakeDetail": {
                "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
                "minimumNoticeDays": "14",
                "endDate": null,
                "startDate": "2025-02-02T00:00:00Z",
                "note": "Test note example"
            }
        }
    }
}
```

Poignée de main concernant le compte de gestion du programme :

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Accepted",
    "source": "aws.partnercentral-channel",
    "account": "123123123123",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-ghi789jkl012m"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "pma-abc123def456g",
        "handshakeType": "PROGRAM_MANAGEMENT_ACCOUNT",
        "id": "ch-ghi789jkl012m",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "456456456456",
        "senderAccountId": "123123123123",
```

```
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "programManagementAccountHandshakeDetail": {
                "program": "SOLUTION_PROVIDER"
            }
        }
    }
}
```

Poignée de main de chaîne rejetée

Les exemples suivants illustrent les événements Channel Handshake Rejected typiques pour différents types de poignée de main.

Poignée de main au début de la période de service :

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Rejected",
    "source": "aws.partnercentral-channel",
    "account": "123123123123",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-abc123def456g"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "rs-jkl012mno345p",
        "handshakeType": "START_SERVICE_PERIOD",
        "id": "ch-abc123def456g",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "789789789789",
        "senderAccountId": "456456456456",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "startServicePeriodHandshakeDetail": {
                "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
                "minimumNoticeDays": "14",
                "endDate": null,
                "startDate": "2025-02-02T00:00:00Z",
```

```
                "note": "Test note example"
            }
        }
    }
}
```

Annulation de la période de service Handshake :

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Rejected",
    "source": "aws.partnercentral-channel",
    "account": "123123123123",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-def456ghi789j"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "rs-jkl012mno345p",
        "handshakeType": "REVOKE_SERVICE_PERIOD",
        "id": "ch-def456ghi789j",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "789789789789",
        "senderAccountId": "456456456456",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "revokeServicePeriodHandshakeDetail": {
                "servicePeriodType": "MINIMUM_NOTICE_PERIOD",
                "minimumNoticeDays": "14",
                "endDate": null,
                "startDate": "2025-02-02T00:00:00Z",
                "note": "Test note example"
            }
        }
    }
}
```

Poignée de main concernant le compte de gestion du programme :

```
{
    "version": "0",
    "id": "01234567-0123-0123-0123-0123456789ab",
    "detail-type": "Channel Handshake Rejected",
    "source": "aws.partnercentral-channel",
    "account": "123123123123",
    "time": "2025-02-01T00:00:00Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:partnercentral:us-east-1:123123123123:catalog/AWS/channel-handshake/
ch-ghi789jkl012m"
    ],
    "detail": {
        "requestId": "12345678-1234-1234-1234-123456789abc",
        "catalog": "AWS",
        "associatedResourceIdentifier": "pma-abc123def456g",
        "handshakeType": "PROGRAM_MANAGEMENT_ACCOUNT",
        "id": "ch-ghi789jkl012m",
        "ownerAccountId": "123123123123",
        "receiverAccountId": "456456456456",
        "senderAccountId": "123123123123",
        "senderDisplayName": "TestDisplayName",
        "handshakeDetail": {
            "programManagementAccountHandshakeDetail": {
                "program": "SOLUTION_PROVIDER"
            }
        }
    }
}
```


Tester dans un bac à sable

Les partenaires peuvent utiliser un bac à sable pour tester et valider leurs interactions avec l'API AWS Partner Central dans un environnement sécurisé et isolé, afin de garantir le bon fonctionnement avant de promouvoir leur solution dans l'environnement de production.AWS propose aux utilisateurs de l'API AWS Partner Central un bac à sable dynamique qui renvoie des réponses similaires à celles de l'environnement de production.AWS ne fournit pas d'interface utilisateur à l'environnement sandbox. Les partenaires doivent donc s'appuyer sur les réponses programmatiques pour tester leurs solutions.

Accès à l'environnement sandbox

Les partenaires ont accès à l'environnement de test dès qu'ils les associent Compte AWS au compte Partner Central. Pour plus d'informations, consultez Lier votre AWS compte au compte AWS Partner Central. Chaque demande inclut un `catalog` paramètre qui détermine l'environnement des données. Lorsqu'il `catalog` est défini sur`AWS`, il fait référence aux données de production, et lorsqu'il est défini sur`Sandbox`, il fait référence aux données du sandbox.

Informations importantes concernant l'environnement sandbox

1. Actualisation des données :AWS actualise les données une fois par an dans l'environnement sandbox (généralement au début de l'année). Après cette actualisation, vous risquez de perdre certaines données de votre environnement de test.
2. Étendue du test : l'environnement sandbox est généralement utilisé pour les tests fonctionnels et non pour tester l'évolutivité ou les performances.

Rubriques

- Tester dans un bac à sable pour AWS API du compte Partner Central
- Tester dans un bac à sable pour AWS API de vente centralisée pour les partenaires
- Tester dans un bac à sable pour AWS API Partner Central Benefits
- Tester dans un bac à sable pour AWS API Partner Central Channel
- Tester dans un bac à sable pour AWS API de mesure du chiffre d'affaires de Partner Central

Tester dans un bac à sable pour AWS API du compte Partner Central

Comment utiliser le bac à sable

Pour utiliser le bac à sable, procédez comme suit :

1. Créez un rôle IAM :

   Créez un rôle IAM dans le compte Compte AWS associé à votre compte AWS Partner Central.
2. Attribuer une politique :

   Attribuez la politique suivante au rôle IAM. Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AccountManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:AcceptConnectionInvitation",
        "partnercentral:AssociateAwsTrainingCertificationEmailDomain",
        "partnercentral:CancelConnection",
        "partnercentral:CancelConnectionInvitation",
        "partnercentral:CancelProfileUpdateTask",
        "partnercentral:CreateConnectionInvitation",
        "partnercentral:CreatePartner",
        "partnercentral:DisassociateAwsTrainingCertificationEmailDomain",
        "partnercentral:GetAllianceLeadContact",
        "partnercentral:GetConnection",
        "partnercentral:GetConnectionInvitation",
        "partnercentral:GetConnectionPreferences",
        "partnercentral:GetPartner",
        "partnercentral:GetProfileUpdateTask",
        "partnercentral:GetProfileVisibility",
  "partnercentral:GetVerification",
        "partnercentral:ListConnectionInvitations",
        "partnercentral:ListConnections",
        "partnercentral:ListPartners",
        "partnercentral:PutAllianceLeadContact",
        "partnercentral:PutProfileVisibility",
        "partnercentral:RejectConnectionInvitation",
        "partnercentral:SendEmailVerificationCode",
        "partnercentral:StartProfileUpdateTask",
  "partnercentral:StartVerification",
        "partnercentral:UpdateConnectionPreferences"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "Sandbox"
          ]
        }
      }
    },
```

```
    {
      "Sid": "TaggingAccess",
      "Effect": "Allow",
      "Action": [
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
      ],
      "Resource": [
        "arn:aws:partnercentral:*:*:catalog/Sandbox/partner/*"
      ],
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "Sandbox"
          ]
        }
      }
    }
  ]
}
```

3. Utilisez les informations d'identification du rôle IAM :

   Utilisez les informations d'identification (clé secrète et clé d'accès) de ce rôle IAM dans votre solution pour effectuer les actions d'API.

Test AWS actions

Pendant la phase de test, il est souvent nécessaire de simuler AWS des actions. Cette simulation permet aux partenaires de tester de manière approfondie le flux complet de bout en bout de leur intégration aux AWS services. Chaque demande inclut un paramètre de catalogue qui détermine l'environnement des données.

Simulation de la création d'un partenaire

Pour simuler la création d'un partenaire, `"catalog": "Sandbox"` incluez-le dans la charge utile de l'`CreatePartner`action.

Par exemple :

```
{
```

```
  "ClientToken": "client-generated-uuid",
  "Catalog": "Sandbox",
  "LegalName": "Your Name",
  "PrimarySolutionType": "SOLUTION_TYPE",
  "AllianceLeadContact": {
    "FirstName": "Example First Name",
    "LastName": "Example Last Name",
    "BusinessTitle": "Example Title",
    "Email": "example@domain.com"
  },
  "EmailVerificationCode": "123456"
}
```

Remarque : Dans Sandbox, la vérification des e-mails est désactivée. Vous pouvez utiliser n'importe quel chiffre à six chiffres EmailVerificationCode pour les tests. N'utilisez pas non plus de « tags » dans la demande.

Remarques de test supplémentaires

1. Pour tester d'autres actions, définissez `"catalog": "Sandbox"` la charge utile.
2. Pour tester les connexions aux comptes partenaires dans le sandbox, vous devez exécuter une `StartProfileUpdateTask` action après avoir créé le partenaire dans le catalogue sandbox.
3. Pour passer à la production, remplacez la valeur du `Sandbox` catalogue par`AWS`.

Événements de test dans l'environnement sandbox

Les partenaires peuvent utiliser les événements de l'environnement sandbox pour aider à tester les implémentations basées sur les événements. Configurez EventBridge de la même manière Compte AWS des règles pour écouter les événements du sandbox en les spécifiant `catalog: Sandbox` dans les détails de l'événement. Pour de plus amples informations, veuillez consulter Notifications pour AWS API du compte Partner Central.

Exemple de règle d'événement :

```
{
  "source": ["aws.partnercentral-account"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

Les règles d'événements qui spécifient « `catalog` comme » ne `Sandbox` correspondront qu'aux événements provenant du sandbox, générés par les actions que vous effectuez dans l'environnement sandbox.

Tester dans un bac à sable pour AWS API de vente centralisée pour les partenaires

Comment utiliser le bac à sable

1. Créez un rôle IAM :

   Créez un rôle IAM dans le compte Compte AWS associé à votre compte AWS Partner Central.
2. Attribuer une politique :

   Attribuez la politique suivante au rôle IAM. Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "partnercentral:CreateOpportunity",
                "partnercentral:UpdateOpportunity",
                "partnercentral:ListOpportunities",
                "partnercentral:GetOpportunity",
                "partnercentral:GetAwsOpportunitySummary",
                "partnercentral:ListSolutions",
                "partnercentral:AssociateOpportunity",
                "partnercentral:DisassociateOpportunity",
                "partnercentral:AssignOpportunity",
                "partnercentral:SubmitOpportunity",
                "partnercentral:AcceptEngagementInvitation",
                "partnercentral:CreateEngagementInvitation",
                "partnercentral:RejectEngagementInvitation",
                "partnercentral:GetEngagementInvitation",
                "partnercentral:ListEngagementInvitations",
                "partnercentral:StartEngagementFromOpportunityTask",
                "partnercentral:StartEngagementByAcceptingInvitationTask",
```

```
                "partnercentral:CreateResourceSnapshotJob",
                "partnercentral:StartResourceSnapshotJob",
                "partnercentral:CreateEngagement"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": "Sandbox"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListEntities",
                "aws-marketplace:DescribeEntity"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:SearchAgreements",
                "aws-marketplace:DescribeAgreement"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws-marketplace:PartyType": "Proposer"
                }
            }
        }
    ]
}
```

3. Utilisez les informations d'identification du rôle IAM :

   Utilisez les informations d'identification (clé secrète et clé d'accès) de ce rôle IAM dans votre solution pour effectuer les actions d'API.

Test AWS actions

Pendant la phase de test, il est souvent nécessaire de simuler AWS des actions. Cette simulation permet aux partenaires de tester de manière approfondie le flux complet de bout en bout de leur intégration aux AWS services.

Simulation de la création d'un AWS opportunité créée

Pour simuler la création d'une AWS opportunité créée, incluez `"Catalog": "Sandbox"` et `"Origin": "AWS Referral"` dans la charge utile de l'`CreateOpportunity`action.

Par exemple :

```
{
    "Catalog": "Sandbox",
    "Origin": "AWS Referral",
    "OpportunityIdentifier": "O123456",
    "Title": "Test Opportunity",
    ...
}
```

Simulation de mises à jour d'une opportunité créée par un partenaire

Pour simuler des mises à jour ou d'autres actions sur une opportunité créée par un partenaire, utilisez l'`UpdateOpportunity`action avec `Lifecycle.ReviewStatus: “Approved”` et/ou `"Catalog": "Sandbox"` `“Action Required”` dans la charge utile.

Si vous utilisez la charge utile liée à l'`GetOpportunity`action de mise à jour, assurez-vous de remplacer l'ID par et supprimez les champs suivants : `Identifier`

- `CreatedDate`
- `OpportunityTeam`
- `RelatedEntityIdentifiers`

Par exemple :

```
{
    "Catalog": "Sandbox",
    "Identifier": "O123456",
    "Title": "Updated Test Opportunity",
```

```
    "Lifecycle": {
        "ReviewStatus": "Approved"
    }
    ...
}
```

Événements de test dans l'environnement sandbox

Les partenaires peuvent utiliser les événements d'opportunité issus de l'environnement sandbox pour aider à tester les implémentations basées sur les événements. Configurez EventBridge de la même manière Compte AWS des règles pour écouter les événements du sandbox en les spécifiant `catalog: sandbox` dans les détails de l'événement. Pour de plus amples informations, veuillez consulter Notifications pour AWS API de vente centralisée pour les partenaires.

Exemple de règle d'événement :

```
{
    "source": ["aws.partnercentral-selling"],
    "detail": {
        "catalog": ["Sandbox"]
    }
}
```

Les règles d'événements qui spécifient « `catalog` comme » ne `Sandbox` correspondront qu'aux événements provenant du sandbox, générés par les actions que vous effectuez dans l'environnement sandbox.

Remarques de test supplémentaires

1. `AssociateOpportunity`Action de test :
   a. Utilisez la solution par défaut `"S-1234567"` pour tester l'`AssociateOpportunity`action.
   b. Pour tester les offres Marketplace, utilisez un véritable identifiant d'offre associé à votre compte.
2. Pour passer à la production, remplacez la `catalog` valeur par `SandboxAWS`.

Obtenir de l'aide

Si vous rencontrez des difficultés pour intégrer votre CRM àAWS, ou si vous devez tester un scénario spécifique qui n'est pas abordé ici, veuillez contacter l'assistance en soulevant un dossier en suivant les étapes suivantes :

1. Connectez-vous à AWS Partner Central à l'aide de vos identifiants AWS Partner Network.
2. Dans le Support Center de Partner Central, choisissez `Open New Case` d'enregistrer un nouveau dossier. Complétez les champs comme suit :
   a. Type de coque de support : `AWS Partner Central`
   b. Question concernant :`Partner Central Tools or ACE leads and opportunities`.
   c. Soyez précis : sélectionnez le type de dossier d'intégration CRM le plus approprié.
   d. Objet : Incluez une brève description de la demande.
   e. Description : fournissez une description détaillée des problèmes, des questions, des erreurs et des étapes de résolution des problèmes.
   f. Pièces jointes : incluez des journaux et des captures d'écran, le cas échéant.

Tester dans un bac à sable pour AWS API Partner Central Benefits

Comment utiliser le bac à sable

Pour utiliser le bac à sable, procédez comme suit :

1. Créez un rôle IAM :

   Créez un rôle IAM dans le compte Compte AWS associé à votre compte AWS Partner Central.
2. Attribuer une politique :

   Attribuez la politique suivante au rôle IAM. Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

```
{
  "Version": "2012-10-17",
    "Statement": [{
            "Sid": "BenefitsManagement",
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListBenefits",
                "partnercentral:GetBenefit",
                "partnercentral:CreateBenefitApplication",
                "partnercentral:AmendBenefitApplication",
                "partnercentral:UpdateBenefitApplication",
                "partnercentral:SubmitBenefitApplication",
                "partnercentral:GetBenefitApplication",
```

```
                "partnercentral:CancelBenefitApplication",
                "partnercentral:RecallBenefitApplication",
                "partnercentral:ListBenefitApplications",
                "partnercentral:AssociateBenefitApplicationResource",
                "partnercentral:DisasssociateBenefitApplicationResource",
                "partnercentral:ListBenefitAllocations",
                "partnercentral:GetBenefitAllocation",
                "partnercentral:TagResource",
                "partnercentral:UntagResource",
                "partnercentral:ListTagsForResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "AWSPartnerOpportunityAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:GetAwsOpportunitySummary",
                "partnercentral:GetOpportunity",
                "partnercentral:ListOpportunities"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "ListingAWSMarketplaceEntities",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListEntities"
            ],
            "Resource": "*"
```

```
        },
        {
            "Sid": "AWSMarketplaceOffersAccess",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:DescribeEntity"
            ],

            "Resource": [
                "arn:aws:aws-marketplace:::AWSMarketplace/Offer/*"
            ]
        },
        {
            "Sid": "S3PutObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject"
            ],
            "Resource": ["arn:aws:s3:::aws-partner-central-marketplace-ephemeral-
writeonly-files/*"]
        },
        {
            "Sid": "TaggingAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:TagResource",
                "partnercentral:UntagResource",
                "partnercentral:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:partnercentral:us-east-1:*:catalog/Sandbox/benefit-
application/*",
                "arn:aws:partnercentral:us-east-1:*:catalog/Sandbox/benefit-
allocation/*"
            ],
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        }
    ]
```

```
}
```

3. Utilisez les informations d'identification du rôle IAM :

   Utilisez les informations d'identification (clé secrète et clé d'accès) de ce rôle IAM dans votre solution pour effectuer les actions d'API.

Test AWS actions

Pendant la phase de test, il est souvent nécessaire de simuler AWS des actions. Cette simulation permet aux partenaires de tester de manière approfondie le flux complet de bout en bout de leur intégration aux AWS services. Chaque demande inclut un paramètre de catalogue qui détermine l'environnement des données.

Simulation de la création d'une demande de prestations

Pour simuler la création d'une demande de prestations, `"catalog": "Sandbox"` incluez dans la charge utile de l'`CreateBenefitApplication`action la charge utile.

Par exemple :

```
{
  "Catalog": "Sandbox",
  "ClientToken": "String",
  "BenefitIdentifier": "String",
  "FulfillmentTypes": ["String"],
  "BenefitApplicationDetails": JsonDocument,
  "Name": "String",
  "Description": "String",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "PartnerContacts": [
    {
      "BusinessTitle": "string",
      "Email": "string",
      "FirstName": "string",
      "LastName": "string",
      "Phone": "string"
```

```
    }
  ],
  "FileDetails": [
    {
      "FileURI": "String",
      "BusinessUseCase": "String"
    }
  ],
  "AssociatedResources": [
    {
      "ResourceType": "BENEFIT_ALLOCATION | OPPORTUNITY",
      "ResourceArn": "ARN"
    }
  ]
}
```

Remarques de test supplémentaires

1. Pour tester d'autres actions, définissez `"catalog": "Sandbox"` la charge utile.
2. Les identifiants Benefit du catalogue Sandbox et du catalogue AWS sont différents. Passez un appel d'`GetBenefit`API avec `"catalog": "Sandbox"` pour obtenir l'identifiant Benefit dans le Sandbox.
3. L'allocation des avantages dans le catalogue Sandbox et dans le catalogue AWS est différente.
4. Pour passer à la production, remplacez la valeur du `Sandbox` catalogue par`AWS`.

Événements de test dans l'environnement sandbox

Les partenaires peuvent utiliser les événements de l'environnement sandbox pour aider à tester les implémentations basées sur les événements. Configurez EventBridge de la même manière Compte AWS des règles pour écouter les événements du sandbox en les spécifiant `catalog: Sandbox` dans les détails de l'événement. Pour de plus amples informations, veuillez consulter Notifications pour AWS API Partner Central Benefits.

Exemple de règle d'événement :

```
{
  "source": ["aws.partnercentral-benefits"],
  "detail": {
    "catalog": ["Sandbox"]
```

```
  }
}
```

Les règles d'événements qui spécifient « `catalog` comme » ne `Sandbox` correspondront qu'aux événements provenant du sandbox, générés par les actions que vous effectuez dans l'environnement sandbox.

Tester dans un bac à sable pour AWS API Partner Central Channel

Les comptes enregistrés dans Sandbox ne sont pas éligibles aux avantages des chaînes et ne sont pas validés par rapport à toutes les exigences du programme.

Comment utiliser le bac à sable

Pour utiliser le bac à sable, procédez comme suit :

1. Créez un rôle IAM :

   Créez un rôle IAM dans le compte Compte AWS associé à votre compte AWS Partner Central.

2. Attribuer une politique :

   Attribuez la politique suivante au rôle IAM. Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ChannelManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:CreateProgramManagementAccount",
        "partnercentral:UpdateProgramManagementAccount",
        "partnercentral:DeleteProgramManagementAccount",
        "partnercentral:ListProgramManagementAccounts",
        "partnercentral:GetProgramManagementAccount",
        "partnercentral:CreateRelationship",
        "partnercentral:UpdateRelationship",
        "partnercentral:DeleteRelationship",
        "partnercentral:GetRelationship",
        "partnercentral:ListRelationships",
```

```
        "partnercentral:CreateChannelHandshake",
        "partnercentral:AcceptChannelHandshake",
        "partnercentral:RejectChannelHandshake",
        "partnercentral:CancelChannelHandshake",
        "partnercentral:ListChannelHandshakes"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "Sandbox"
          ]
        }
      }
    },
    {
      "Sid": "TaggingAccess",
      "Effect": "Allow",
      "Action": [
        "partnercentral:TagResource",
        "partnercentral:UntagResource",
        "partnercentral:ListTagsForResource"
      ],
      "Resource": [
        "arn:aws:partnercentral:*:*:catalog/Sandbox/program-management-account/*",
        "arn:aws:partnercentral:*:*:catalog/Sandbox/channel-handshake/*"
      ],
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "Sandbox"
          ]
        }
      }
    }
  ]
}
```

3. Utilisez les informations d'identification du rôle IAM :

   Utilisez les informations d'identification (clé secrète et clé d'accès) de ce rôle IAM dans votre solution pour effectuer les actions d'API.

Test AWS actions

Pendant la phase de test, il est souvent nécessaire de simuler AWS des actions. Cette simulation permet aux partenaires de tester de manière approfondie le flux complet de bout en bout de leur intégration aux AWS services. Chaque demande inclut un paramètre de catalogue qui détermine l'environnement des données.

Simulation de la création d'un compte de gestion de programme

Pour simuler la création d'un compte de gestion de programme, `"catalog": "Sandbox"` incluez-le dans la charge utile de l'`CreateProgramManagementAccount`action.

Par exemple :

```
{
  "catalog": "Sandbox",
  "accountId": "123456789012",
  "displayName": "Test PMA Name",
  "clientToken": "123abc456def789ghi",
  "program": "SOLUTION_PROVIDER",
  "tags": [
    {
      "key": "testkey",
      "value": "testvalue"
    }
  ]
}
```

Simulation de mises à jour d'une relation

Pour simuler les mises à jour d'une relation, utilisez l'`UpdateRelationship`action avec `"catalog": "Sandbox"` dans la charge utile.

Par exemple :

```
{
  "catalog": "Sandbox",
  "displayName": "Updated Test PMA",
  "identifier": "rs-abc123def456g",
  "programManagementAccountIdentifier": "pma-123abc456def7"
}
```

Remarques de test supplémentaires

1. Pour tester d'autres actions, définissez `"catalog": "Sandbox"` la charge utile.
2. Pour passer à la production, remplacez la valeur du `Sandbox` catalogue par`AWS`.

Événements de test dans l'environnement sandbox

Les partenaires peuvent utiliser les événements de l'environnement sandbox pour aider à tester les implémentations basées sur les événements. Configurez EventBridge de la même manière Compte AWS des règles pour écouter les événements du sandbox en les spécifiant `catalog: Sandbox` dans les détails de l'événement. Pour de plus amples informations, veuillez consulter Notifications pour AWS API Partner Central Channel.

Exemple de règle d'événement :

```
{
  "source": ["aws.partnercentral-channel"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

Les règles d'événements qui spécifient « `catalog` comme » ne `Sandbox` correspondront qu'aux événements provenant du sandbox, générés par les actions que vous effectuez dans l'environnement sandbox.

Obtenir de l'aide

Si vous rencontrez des difficultés lors des tests, ou si vous devez tester un scénario spécifique qui n'est pas abordé ici, veuillez contacter l'assistance en soulevant un dossier en suivant les étapes suivantes :

1. Connectez-vous à AWS Partner Central à l'aide de vos identifiants AWS Partner Network.
2. Dans le Support Center de Partner Central, choisissez `Open New Case` d'enregistrer un nouveau dossier. Complétez les champs comme suit :
   a. Type de coque de support : `Channel Program`
   b. Question concernant : `General Program Inquiry`
   c. Soyez précis : sélectionnez le type de boîtier de chaîne le plus approprié.

d. Objet : Incluez une brève description de la demande.

e. Description : fournissez une description détaillée des problèmes, des questions, des erreurs et des étapes de résolution des problèmes.

f. Pièces jointes : incluez des journaux et des captures d'écran, le cas échéant.

Tester dans un bac à sable pour AWS API de mesure du chiffre d'affaires de Partner Central

Comment utiliser le bac à sable

Pour utiliser le bac à sable, procédez comme suit :

1. Créez un rôle IAM :

   Créez un rôle IAM dans le compte Compte AWS associé à votre compte AWS Partner Central.

2. Attribuer une politique :

   Attribuez la politique suivante au rôle IAM. Pour de plus amples informations, consulter Ajout et suppression d'autorisations d'identité IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RevenueMeasurementCreation",
            "Effect": "Allow",
            "Action": [
                "partnercentral:CreateRevenueAttribution",
                "partnercentral:CreateMarketplaceRevenueShare",
                "partnercentral:CreateMarketplaceRevenueShareAllocation"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        },
```

```
        {
            "Sid": "RevenueMeasurementListing",
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListRevenueAttributions",
                "partnercentral:ListRevenueAttributionAllocations",
                "partnercentral:ListMarketplaceRevenueShares",
                "partnercentral:ListMarketplaceRevenueShareAllocations"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "RevenueAttributionManagement",
            "Effect": "Allow",
            "Action": [
                "partnercentral:GetRevenueAttribution",
                "partnercentral:UpdateRevenueAttribution",
                "partnercentral:GetRevenueAttributionAllocation",
                "partnercentral:StartRevenueAttributionAllocationsTask",
                "partnercentral:GetRevenueAttributionAllocationsTask"
            ],
            "Resource": "arn:aws:partnercentral:*:*:catalog/Sandbox/revenue-
attribution/*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "MarketplaceRevenueShareManagement",
            "Effect": "Allow",
            "Action": [
                "partnercentral:GetMarketplaceRevenueShare",
                "partnercentral:GetMarketplaceRevenueShareAllocation",
```

```
                "partnercentral:UpdateMarketplaceRevenueShareAllocation"
            ],
            "Resource": "arn:aws:partnercentral:*:*:catalog/Sandbox/marketplace-
revenue-share/*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "TaggingAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:TagResource",
                "partnercentral:UntagResource",
                "partnercentral:ListTagsForResource"
            ],
            "Resource": [
                "arn:aws:partnercentral:*:*:catalog/Sandbox/revenue-attribution/*",
                "arn:aws:partnercentral:*:*:catalog/Sandbox/marketplace-revenue-
share/*"
            ],
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "OpportunityAccess",
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListOpportunities",
                "partnercentral:GetOpportunity"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "partnercentral:Catalog": [
```

```
                        "Sandbox"
                    ]
                }
            }
        },
        {
            "Sid": "ListingAWSMarketplaceEntities",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:ListEntities"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AWSMarketplaceEntityAccess",
            "Effect": "Allow",
            "Action": [
                "aws-marketplace:DescribeEntity"
            ],
            "Resource": [
                "arn:aws:aws-marketplace:*:*:AWSMarketplace*/Offer/*"
            ]
        }
    ]
}
```

3. Utilisez les informations d'identification du rôle IAM :

   Utilisez les informations d'identification (clé secrète et clé d'accès) de ce rôle IAM dans votre solution pour effectuer les actions d'API.

Test AWS actions

Pendant la phase de test, il est souvent nécessaire de simuler AWS des actions. Cette simulation permet aux partenaires de tester de manière approfondie le flux complet de bout en bout de leur intégration aux AWS services. Chaque demande inclut un paramètre de catalogue qui détermine l'environnement des données.

Simulation de la création d'un identifiant d'attribution de revenus

Pour simuler la création d'un ID d'attribution de revenus, `"Catalog": "Sandbox"` incluez-le dans la charge utile de l'`CreateRevenueAttribution`action.

Par exemple :

```
{
  "Catalog": "Sandbox",
  "ClientToken": "String",
  "Name": "String",
  "Description": "String",
  "TenancyModel": "MULTI_TENANT | SINGLE_TENANT",
  "ProductIdentifier": "String",
  "Tags": [
    {
      "Key": "String",
      "Value": "String"
    }
  ]
}
```

Remarques de test supplémentaires

1. Pour tester d'autres actions, définissez `"Catalog": "Sandbox"` la charge utile.
2. Pour passer à la production, remplacez la valeur du `Sandbox` catalogue par`AWS`.

Exemples de code pour l'utilisation de Partner Central AWS Kits SDK

Les exemples de code suivants montrent comment utiliser Partner Central avec un kit de développement AWS logiciel (SDK).

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Les scénarios sont des exemples de code qui vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions au sein d'un même service ou combinés à d'autres Services AWS.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit de développement logiciel (SDK).

Exemples de code

- Exemples de base pour l'utilisation de Partner Central AWS Kits SDK
  - Actions destinées à l'utilisation de Partner Central AWS Kits SDK
    - À utiliser AssignOpportunityavec un AWS Kit SDK
    - À utiliser AssociateOpportunityavec un AWS Kit SDK
    - À utiliser CreateOpportunityavec un AWS Kit SDK
    - À utiliser DisassociateOpportunityavec un AWS Kit SDK
    - À utiliser GetAwsOpportunitySummaryavec un AWS Kit SDK
    - À utiliser GetEngagementInvitationavec un AWS Kit SDK
    - À utiliser GetOpportunityavec un AWS Kit SDK
    - À utiliser ListEngagementInvitationsavec un AWS Kit SDK
    - À utiliser ListOpportunitiesavec un AWS Kit SDK
    - À utiliser ListSolutionsavec un AWS Kit SDK
    - À utiliser RejectEngagementInvitationavec un AWS Kit SDK
    - À utiliser StartEngagementByAcceptingInvitationTaskavec un AWS Kit SDK

    - À utiliser StartEngagementFromOpportunityTaskavec un AWS Kit SDK
    - À utiliser UpdateOpportunityavec un AWS Kit SDK
- Scénarios d'utilisation de Partner Central AWS Kits SDK
    - Mise à jour de l’entité associée à une opportunité

Exemples de base pour l'utilisation de Partner Central AWS Kits SDK

Les exemples de code suivants montrent comment utiliser les principes de base de AWS Partner Central avec AWS les SDK.

Exemples

- Actions destinées à l'utilisation de Partner Central AWS Kits SDK
    - À utiliser AssignOpportunityavec un AWS Kit SDK
    - À utiliser AssociateOpportunityavec un AWS Kit SDK
    - À utiliser CreateOpportunityavec un AWS Kit SDK
    - À utiliser DisassociateOpportunityavec un AWS Kit SDK
    - À utiliser GetAwsOpportunitySummaryavec un AWS Kit SDK
    - À utiliser GetEngagementInvitationavec un AWS Kit SDK
    - À utiliser GetOpportunityavec un AWS Kit SDK
    - À utiliser ListEngagementInvitationsavec un AWS Kit SDK
    - À utiliser ListOpportunitiesavec un AWS Kit SDK
    - À utiliser ListSolutionsavec un AWS Kit SDK
    - À utiliser RejectEngagementInvitationavec un AWS Kit SDK
    - À utiliser StartEngagementByAcceptingInvitationTaskavec un AWS Kit SDK
    - À utiliser StartEngagementFromOpportunityTaskavec un AWS Kit SDK
    - À utiliser UpdateOpportunityavec un AWS Kit SDK

Actions destinées à l'utilisation de Partner Central AWS Kits SDK

Les exemples de code suivants montrent comment effectuer des actions individuelles de Partner Central à l'aide des AWS SDK. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Ces extraits appellent l'API Partner Central et sont des extraits de code de programmes plus volumineux qui doivent être exécutés en contexte. Vous pouvez voir les actions dans leur contexte dans Scénarios d'utilisation de Partner Central AWS Kits SDK.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, consultez la Référence des API AWS Partner Central.

Exemples

- À utiliser AssignOpportunityavec un AWS Kit SDK
- À utiliser AssociateOpportunityavec un AWS Kit SDK
- À utiliser CreateOpportunityavec un AWS Kit SDK
- À utiliser DisassociateOpportunityavec un AWS Kit SDK
- À utiliser GetAwsOpportunitySummaryavec un AWS Kit SDK
- À utiliser GetEngagementInvitationavec un AWS Kit SDK
- À utiliser GetOpportunityavec un AWS Kit SDK
- À utiliser ListEngagementInvitationsavec un AWS Kit SDK
- À utiliser ListOpportunitiesavec un AWS Kit SDK
- À utiliser ListSolutionsavec un AWS Kit SDK
- À utiliser RejectEngagementInvitationavec un AWS Kit SDK
- À utiliser StartEngagementByAcceptingInvitationTaskavec un AWS Kit SDK
- À utiliser StartEngagementFromOpportunityTaskavec un AWS Kit SDK
- À utiliser UpdateOpportunityavec un AWS Kit SDK

À utiliser **`AssignOpportunity`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `AssignOpportunity`.

Java

SDK pour Java 2.x

Réattribuez une opportunité existante à un autre utilisateur.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.AssignOpportunityRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.AssignOpportunityResponse;
import
 software.amazon.awssdk.services.partnercentralselling.model.AssigneeContact;

/*
Purpose
PC-API-07 Assigning a new owner
*/

public class AssignOpportunity {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

     String assigneeFirstName = "John";
```

```
    String assigneeLastName = "Doe";

    String assigneeEmail = "test@test.com";

    String businessTitle = "PartnerAccountManager";

    AssignOpportunityResponse response = getResponse(opportunityId,
assigneeFirstName, assigneeLastName, assigneeEmail, businessTitle);

    ReferenceCodesUtils.formatOutput(response);
   }

static AssignOpportunityResponse getResponse(String opportunityId, String
assigneeFirstName, String assigneeLastName, String assigneeEmail, String
businessTitle) {

 AssignOpportunityRequest assignOpportunityRequest =
AssignOpportunityRequest.builder()
   .catalog(Constants.CATALOG_TO_USE)
         .identifier(opportunityId)
         .assignee(AssigneeContact.builder()
           .firstName(assigneeFirstName)
           .lastName(assigneeLastName)
           .email(assigneeEmail)
           .businessTitle(businessTitle)
           .build())
         .build();

       AssignOpportunityResponse response =
client.assignOpportunity(assignOpportunityRequest);

       return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous AssignOpportunityà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Réattribuez une opportunité existante à un autre utilisateur.

```
#!/usr/bin/env python

"""
Purpose
PC-API-07 Assigning a new owner
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def assign_opportunity(identifier):
    assign_opportunity_request ={
        "Catalog": CATALOG_TO_USE,
     "Identifier": identifier,
        "Assignee": {
            "BusinessTitle": "OpportunityOwner",
            "Email": "test@test.com",
            "FirstName": "John",
            "LastName": "Doe"
        }
    }
    try:
        # Perform an API call
        response =
 partner_central_client.assign_opportunity(**assign_opportunity_request)
        return response
```

```
    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    identifier = "04236468"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Assigning a new owner to an opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(assign_opportunity(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez AssignOpportunityle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`AssociateOpportunity`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `AssociateOpportunity`.

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 :

- Mise à jour de l’entité associée à une opportunité

Java

SDK pour Java 2.x

Créez une association officielle entre une opportunité et diverses entités connexes.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityResponse

/*
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
*/

public class AssociateOpportunity {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

     String entityType = "Solutions";

     String entityIdentifier = "S-0000000";
```

```
     AssociateOpportunityResponse response = getResponse(opportunityId,
 entityType, entityIdentifier );

     ReferenceCodesUtils.formatOutput(response);
    }

static AssociateOpportunityResponse getResponse(String opportunityId, String
 entityType, String entityIdentifier) {

        AssociateOpportunityRequest associateOpportunityRequest =
 AssociateOpportunityRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .opportunityIdentifier(opportunityId)
          .relatedEntityType(entityType)
          .relatedEntityIdentifier(entityIdentifier)
          .build();

        AssociateOpportunityResponse response =
 client.associateOpportunity(associateOpportunityRequest);

        return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous AssociateOpportunityà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Créez une association officielle entre une opportunité et diverses entités connexes.

```
#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""
```

```
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def associate_opportunity(entity_type, entity_identifier, opportunityIdentifier):
    associate_opportunity_request ={
        "Catalog": CATALOG_TO_USE,
     "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : entity_type,
        "RelatedEntityIdentifier" : entity_identifier
    }
    try:
        # Perform an API call
        response =
 partner_central_client.associate_opportunity(**associate_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    #entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
    entity_type = "Solutions"
    entity_identifier = "S-0059717"
    opportunityIdentifier = "O5465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Associate Opportunity.")
    print("-" * 88)
```

```
    helper.pretty_print_datetime(associate_opportunity(entity_type,
 entity_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez AssociateOpportunityle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`CreateOpportunity`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `CreateOpportunity`.

.NET

SDK pour .NET

Créez une opportunité.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// PDX-License-Identifier: Apache-2.0

using System;
using Newtonsoft.Json;
using Amazon;
using Amazon.Runtime;
using Amazon.PartnerCentralSelling;
using Amazon.PartnerCentralSelling.Model;

namespace AWSExample
{
    class Program
    {
        static readonly string catalogToUse = "AWS";
        static async Task Main(string[] args)
        {
            // Initialize credentials from .aws/credentials file
```

```
            var credentials = new
Amazon.Runtime.CredentialManagement.SharedCredentialsFile();
            if (credentials.TryGetProfile("default", out var profile))
            {
                AWSCredentials awsCredentials =
profile.GetAWSCredentials(credentials);

                var client = new
AmazonPartnerCentralSellingClient(awsCredentials);

                var request = new CreateOpportunityRequest
                {
                    Catalog = catalogToUse,
                    Origin = "Partner Referral",
                    Customer = new Customer
                    {
                        Account = new Account
                        {
                            Address = new Address
                            {
                                CountryCode = "US",
                                PostalCode = "99502",
                                StateOrRegion = "Alaska"
                            },
                            CompanyName = "TestCompanyName",
                            Duns = "123456789",
                            WebsiteUrl = "www.test.io",
                            Industry = "Automotive"
                        },
                        Contacts = new List<Contact>
                        {
                            new Contact
                            {
                                Email = "test@test.io",
                                FirstName = "John  ",
                                LastName = "Doe",
                                Phone = "+14444444444",
                                BusinessTitle = "test title"
                            }
                        }
                    },
                    LifeCycle = new LifeCycle
                    {
                        ReviewStatus = "Submitted",
```

```
                    TargetCloseDate = "2024-12-30"
                },
                Marketing = new Marketing
                {
                    Source = "None"
                },
                OpportunityType = "Net New Business",
                PrimaryNeedsFromAws = new List<string> { "Co-Sell -
Architectural Validation" },
                Project = new Project
                {
                    Title = "Moin Test UUID",
                    CustomerBusinessProblem = "Sandbox is not working as
expected",
                    CustomerUseCase = "AI Machine Learning and Analytics",
                    DeliveryModels = new List<string> { "SaaS or PaaS" },
                    ExpectedCustomerSpend = new List<ExpectedCustomerSpend>
                    {
                        new ExpectedCustomerSpend
                        {
                            Amount = "2000.0",
                            CurrencyCode = "USD",
                            Frequency = "Monthly",
                            TargetCompany = "Ibexlabs"
                        }
                    },
                    SalesActivities = new List<string> { "Initialized
discussions with customer" }
                }
            };

            try
            {
                var response = await client.CreateOpportunityAsync(request);
                Console.WriteLine(response.HttpStatusCode);
                string formattedJson = JsonConvert.SerializeObject(response,
Formatting.Indented);
                Console.WriteLine(formattedJson);
            }
            catch (ValidationException ex)
            {
                Console.WriteLine("Validation error: " + ex.Message);
            }
            catch (AmazonPartnerCentralSellingException e)
```

```
                {
                    Console.WriteLine("Failed:");
                    Console.WriteLine(e.RequestId);
                    Console.WriteLine(e.ErrorCode);
                    Console.WriteLine(e.Message);
                }
            }
            else
            {
                Console.WriteLine("Profile not found.");
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous CreateOpportunityà la section Référence des AWS SDK pour .NET API.

Java

SDK pour Java 2.x

Créez une opportunité.

```
package org.example;

import java.time.Instant;
import java.util.ArrayList;
import java.util.List;

import static org.example.utils.Constants.*;

import org.example.entity.Root;
import org.example.utils.ReferenceCodesUtils;
import org.example.utils.StringSerializer;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import software.amazon.awssdk.services.partnercentralselling.model.Account;
import software.amazon.awssdk.services.partnercentralselling.model.Address;
```

```
import software.amazon.awssdk.services.partnercentralselling.model.Contact;
import
 software.amazon.awssdk.services.partnercentralselling.model.CreateOpportunityRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.CreateOpportunityResponse;
import software.amazon.awssdk.services.partnercentralselling.model.Customer;
import
 software.amazon.awssdk.services.partnercentralselling.model.ExpectedCustomerSpend;
import software.amazon.awssdk.services.partnercentralselling.model.LifeCycle;
import software.amazon.awssdk.services.partnercentralselling.model.Marketing;
import software.amazon.awssdk.services.partnercentralselling.model.MonetaryValue;
import
 software.amazon.awssdk.services.partnercentralselling.model.NextStepsHistory;
import software.amazon.awssdk.services.partnercentralselling.model.Project;
import
 software.amazon.awssdk.services.partnercentralselling.model.SoftwareRevenue;

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

public class CreateOpportunity {

 static final Gson GSON = new GsonBuilder()
   .setObjectToNumberStrategy(ToNumberPolicy.LAZILY_PARSED_NUMBER)
   .registerTypeAdapter(String.class, new StringSerializer())
   .create();

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

 public static void main(String[] args) {

  String inputFile = "CreateOpportunity2.json";

  if (args.length > 0)
   inputFile = args[0];

  CreateOpportunityResponse response = createOpportunity(inputFile);
```

```
 client.close();
}

static CreateOpportunityResponse createOpportunity(String inputFile) {

 String inputString = ReferenceCodesUtils.readInputFileToString(inputFile);

 Root root = GSON.fromJson(inputString, Root.class);

 List<NextStepsHistory> nextStepsHistories = new ArrayList<NextStepsHistory>();
 if ( root.lifeCycle != null && root.lifeCycle.nextStepsHistories != null) {
  for (org.example.entity.NextStepsHistory nextStepsHistoryJson :
root.lifeCycle.nextStepsHistories) {
   NextStepsHistory nextStepsHistory = NextStepsHistory.builder()
     .time(Instant.parse(nextStepsHistoryJson.time))
     .value(nextStepsHistoryJson.value)
                 .build();
   nextStepsHistories.add(nextStepsHistory);
  }
 }

 LifeCycle lifeCycle = null;
 if ( root.lifeCycle != null ) {
  lifeCycle = LifeCycle.builder()
   .closedLostReason(root.lifeCycle.closedLostReason)
   .nextSteps(root.lifeCycle.nextSteps)
   .nextStepsHistory(nextStepsHistories)
   .reviewComments(root.lifeCycle.reviewComments)
   .reviewStatus(root.lifeCycle.reviewStatus)
   .reviewStatusReason(root.lifeCycle.reviewStatusReason)
   .stage(root.lifeCycle.stage)
   .targetCloseDate(root.lifeCycle.targetCloseDate)
   .build();
 }

 Marketing marketing = null;
 if ( root.marketing != null ) {
  marketing = Marketing.builder()
    .awsFundingUsed(root.marketing.awsFundingUsed)
    .campaignName(root.marketing.campaignName)
    .channels(root.marketing.channels)
    .source(root.marketing.source)
    .useCases(root.marketing.useCases)
    .build();
```

```
 }

 Address address = null;
 if ( root.customer != null && root.customer.account != null &&
root.customer.account.address != null ) {
  address = Address.builder()
   .city(root.customer.account.address.city)
               .postalCode(root.customer.account.address.postalCode)
               .stateOrRegion(root.customer.account.address.stateOrRegion)
               .countryCode(root.customer.account.address.countryCode)
               .streetAddress(root.customer.account.address.streetAddress)
               .build();
 }

 Account account = null;
 if ( root.customer != null && root.customer.account!= null) {
  account = Account.builder()
            .address(address)
            .awsAccountId(root.customer.account.awsAccountId)
               .duns(root.customer.account.duns)
               .industry(root.customer.account.industry)
               .otherIndustry(root.customer.account.otherIndustry)
               .companyName(root.customer.account.companyName)
               .websiteUrl(root.customer.account.websiteUrl)
               .build();
 }

 List<Contact> contacts = new ArrayList<Contact>();
 if ( root.customer != null && root.customer.contacts != null) {
  for (org.example.entity.Contact jsonContact : root.customer.contacts) {
   Contact contact = Contact.builder()
                 .email(jsonContact.email)
                 .firstName(jsonContact.firstName)
                 .lastName(jsonContact.lastName)
                 .phone(jsonContact.phone)
                 .businessTitle(jsonContact.businessTitle)
                 .build();
   contacts.add(contact);
  }
 }

 Customer customer = Customer.builder()
   .account(account)
```

```
    .contacts(contacts)
    .build();

 Contact oportunityTeamContact = null;
 if (root.opportunityTeam != null && root.opportunityTeam.get(0) != null ) {
  oportunityTeamContact = Contact.builder()
              .firstName(root.opportunityTeam.get(0).firstName)
              .lastName(root.opportunityTeam.get(0).lastName)
              .email(root.opportunityTeam.get(0).email)
              .phone(root.opportunityTeam.get(0).phone)
              .businessTitle(root.opportunityTeam.get(0).businessTitle)
              .build();
 }

 List<ExpectedCustomerSpend> expectedCustomerSpends = new
ArrayList<ExpectedCustomerSpend>();
 if ( root.project != null && root.project.expectedCustomerSpend != null) {
  for (org.example.entity.ExpectedCustomerSpend expectedCustomerSpendJson :
root.project.expectedCustomerSpend) {
   ExpectedCustomerSpend expectedCustomerSpend = null;
   expectedCustomerSpend = ExpectedCustomerSpend.builder()
     .amount(expectedCustomerSpendJson.amount)
     .currencyCode(expectedCustomerSpendJson.currencyCode)
     .frequency(expectedCustomerSpendJson.frequency)
     .targetCompany(expectedCustomerSpendJson.targetCompany)
     .build();
   expectedCustomerSpends.add(expectedCustomerSpend);
  }
       }

       Project project = null;
       if ( root.project != null) {
        project = Project.builder()
              .title(root.project.title)
              .customerBusinessProblem(root.project.customerBusinessProblem)
              .customerUseCase(root.project.customerUseCase)
              .deliveryModels(root.project.deliveryModels)
              .expectedCustomerSpend(expectedCustomerSpends)
              .salesActivities(root.project.salesActivities)
              .competitorName(root.project.competitorName)
              .otherSolutionDescription(root.project.otherSolutionDescription)
              .build();
       }
```

```
        SoftwareRevenue softwareRevenue = null;
        if ( root.softwareRevenue != null) {
         MonetaryValue monetaryValue = null;
         if ( root.softwareRevenue.value != null) {
          monetaryValue = MonetaryValue.builder()
            .amount(root.softwareRevenue.value.amount)
            .currencyCode(root.softwareRevenue.value.currencyCode)
            .build();
         }
         softwareRevenue = SoftwareRevenue.builder()
           .deliveryModel(root.softwareRevenue.deliveryModel)
           .effectiveDate(root.softwareRevenue.effectiveDate)
           .expirationDate(root.softwareRevenue.expirationDate)
           .value(monetaryValue)
           .build();
        }

  // Building the Actual CreateOpportunity Request
  CreateOpportunityRequest createOpportunityRequest =
 CreateOpportunityRequest.builder()
    .catalog(CATALOG_TO_USE)
    .clientToken(root.clientToken)
    .primaryNeedsFromAwsWithStrings(root.primaryNeedsFromAws)
    .opportunityType(root.opportunityType)
    .lifeCycle(lifeCycle)
    .marketing(marketing)
    .nationalSecurity(root.nationalSecurity)
    .origin(root.origin)
    .customer(customer)
    .project(project)
    .partnerOpportunityIdentifier(root.partnerOpportunityIdentifier)
    .opportunityTeam(oportunityTeamContact)
    .softwareRevenue(softwareRevenue)
    .build();

  CreateOpportunityResponse response =
 client.createOpportunity(createOpportunityRequest);
  System.out.println("Successfully created: " + response);

  return response;
    }

}
```

- Pour plus de détails sur l'API, reportez-vous CreateOpportunityà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Créez une opportunité.

```
#!/usr/bin/env python
import boto3
import logging
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
import utils.helpers as helper
import utils.stringify_details as sd
from botocore.client import ClientError
from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

def create_opportunity(partner_central_client):
    create_opportunity_request = helper.remove_nulls(sd.stringify_json("src/
create_opportunity/createOpportunity.json"))
    try:
        # Perform an API call
        response =
 partner_central_client.create_opportunity(**create_opportunity_request)

        helper.pretty_print_datetime(response)

        # Retrieve the opportunity details
        get_response = partner_central_client.get_opportunity(
            Identifier=response["Id"],
            Catalog=CATALOG_TO_USE
        )
        helper.pretty_print_datetime(get_response)
        return response
    except ClientError as err:
```

```
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Create Opportunity.")
    print("-" * 88)

    partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
    )

    create_opportunity(partner_central_client)

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez CreateOpportunityle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`DisassociateOpportunity`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `DisassociateOpportunity`.

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 :

- Mise à jour de l’entité associée à une opportunité

Java

SDK pour Java 2.x

Supprimez une association existante entre une opportunité et des entités associées.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityReque
import
 software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityRespo

/*
Purpose
PC-API -14 Removing a Solution
PC-API -15 Removing an offer
PC-API -16 Removing a product
entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
*/

public class DisassociateOpportunity {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;
```

```
     String entityType = "Solutions";

     String entityIdentifier = "S-0000000";

     DisassociateOpportunityResponse response = getResponse(opportunityId,
 entityType, entityIdentifier );

     ReferenceCodesUtils.formatOutput(response);
    }

 static DisassociateOpportunityResponse getResponse(String opportunityId, String
 entityType, String entityIdentifier) {

  DisassociateOpportunityRequest disassociateOpportunityRequest =
 DisassociateOpportunityRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .opportunityIdentifier(opportunityId)
          .relatedEntityType(entityType)
          .relatedEntityIdentifier(entityIdentifier)
          .build();

        DisassociateOpportunityResponse response =
 client.disassociateOpportunity(disassociateOpportunityRequest);

        return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous DisassociateOpportunityà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Supprimez une association existante entre une opportunité et des entités associées.

```
#!/usr/bin/env python

"""
Purpose
```

```
PC-API -14 Removing a Solution
PC-API -15 Removing an offer
PC-API -16 Removing a product
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def disassociate_opportunity(entity_type, entity_identifier,
 opportunityIdentifier):
    disassociate_opportunity_request ={
        "Catalog": CATALOG_TO_USE,
     "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : entity_type,
        "RelatedEntityIdentifier" : entity_identifier
    }
    try:
        # Perform an API call
        response =
 partner_central_client.disassociate_opportunity(**disassociate_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    #entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
    entity_type = "Solutions"
    entity_identifier = "S-0049999"
    opportunityIdentifier = "O4397574"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")
```

```
    print("-" * 88)
    print("Get updated Opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(disassociate_opportunity(entity_type,
 entity_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez DisassociateOpportunityle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`GetAwsOpportunitySummary`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `GetAwsOpportunitySummary`.

Java

SDK pour Java 2.x

Récupère le résumé d'une AWS opportunité.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.GetAwsOpportunitySummaryRequ
```

```
import
 software.amazon.awssdk.services.partnercentralselling.model.GetAwsOpportunitySummaryResp

/*
 * Purpose
 * PC-API-25 Retrieves a summary of an AWS Opportunity.
 */

public class GetAwsOpportunitySummary {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

     GetAwsOpportunitySummaryResponse response = getResponse(opportunityId);

     ReferenceCodesUtils.formatOutput(response);
    }

 public static GetAwsOpportunitySummaryResponse getResponse(String opportunityId)
 {

  GetAwsOpportunitySummaryRequest getOpportunityRequest =
 GetAwsOpportunitySummaryRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .relatedOpportunityIdentifier(opportunityId)
          .build();

  GetAwsOpportunitySummaryResponse response =
 client.getAwsOpportunitySummary(getOpportunityRequest);

        return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous GetAwsOpportunitySummaryà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Récupère le résumé d'une AWS opportunité.

```
#!/usr/bin/env python

"""
Purpose
PC-API-25 Retrieves a summary of an AWS Opportunity.
 LifeCycle.ReviewStatus=Approved
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def get_opportunity(identifier):
    get_opportunity_request ={
        "Catalog": CATALOG_TO_USE,
     "RelatedOpportunityIdentifier": identifier
    }
    try:
        # Perform an API call
        response =
 partner_central_client.get_aws_opportunity_summary(**get_opportunity_request)
        return response

    except ClientError as err:
```

```
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    identifier = "O5465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Get AWS Opportunity summary.")
    print("-" * 88)

    helper.pretty_print_datetime(get_opportunity(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez GetAwsOpportunitySummarylle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`GetEngagementInvitation`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `GetEngagementInvitation`.

Java

SDK pour Java 2.x

Récupère les détails d'une invitation d'engagement partagée AWS par un partenaire.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;
```

```
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationReque
import
 software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationRespo

/*
 * Purpose
 * PC-API-22 Get engagement invitation opportunity
 */

public class GetEngagementInvitation {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

     GetEngagementInvitationResponse response = getResponse(opportunityId);

     ReferenceCodesUtils.formatOutput(response);
    }

 static GetEngagementInvitationResponse getResponse(String opportunityId) {

  GetEngagementInvitationRequest getOpportunityRequest =
 GetEngagementInvitationRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .identifier(opportunityId)
          .build();

  GetEngagementInvitationResponse response =
 client.getEngagementInvitation(getOpportunityRequest);
```

```
        return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous GetEngagementInvitationà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Récupère les détails d'une invitation d'engagement partagée AWS par un partenaire.

```
#!/usr/bin/env python

"""
Purpose
PC-API-22  GetOpportunityEngagementInvitation - Retrieves details of a specific
 engagement invitation.
This operation allows partners to view the invitation and its associated
 information,
such as the customer, project, and lifecycle details.
"""
import json
import logging
import boto3
import utils.helpers as helper

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def get_opportunity_engagement_invitation(identifier):
    get_opportunity_engagement_invitation_request ={
        "Catalog": CATALOG_TO_USE,
     "Identifier": identifier
```

```
    }
    try:
        # Perform an API call
        response =
 partner_central_client.get_engagement_invitation(**get_opportunity_engagement_invitation
        return response

    except Exception as err:
        # Catch all client exceptions
        print(json.dumps(err.response))

def usage_demo():
    identifier = "arn:aws:partnercentral-selling:us-east-1:aws:catalog/Sandbox/
engagement-invitation/engi-0000000IS0Qga"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Given the ARN identifier, retrieve details of Opportunity Engagement
 Invitation.")
    print("-" * 88)


 helper.pretty_print_datetime(get_opportunity_engagement_invitation(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez GetEngagementInvitationle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`GetOpportunity`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `GetOpportunity`.

.NET

SDK pour .NET

Profitez d'une opportunité.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// PDX-License-Identifier: Apache-2.0

using System;
using Newtonsoft.Json;
using Amazon;
using Amazon.Runtime;
using Amazon.PartnerCentralSelling;
using Amazon.PartnerCentralSelling.Model;

namespace AWSExample
{
    class Program
    {
        static readonly string catalogToUse = "AWS";
        static readonly string identifier = "O1111111";
        static async Task Main(string[] args)
        {
            // Initialize credentials from .aws/credentials file
            var credentials = new
 Amazon.Runtime.CredentialManagement.SharedCredentialsFile();
            if (credentials.TryGetProfile("default", out var profile))
            {
                AWSCredentials awsCredentials =
 profile.GetAWSCredentials(credentials);

                var client = new
 AmazonPartnerCentralSellingClient(awsCredentials);

                var request = new GetOpportunityRequest
                {
                    Catalog = catalogToUse,
                    Identifier = identifier
                };

                try {
                    var response = await client.GetOpportunityAsync(request);
                    Console.WriteLine(response.HttpStatusCode);
```

```
                    string formattedJson = JsonConvert.SerializeObject(response,
 Formatting.Indented);
                    Console.WriteLine(formattedJson);
                } catch(ValidationException ex) {
                    Console.WriteLine("Validation error: " + ex.Message);
                } catch (AmazonPartnerCentralSellingException e) {
                    Console.WriteLine("Failed:");
                    Console.WriteLine(e.RequestId);
                    Console.WriteLine(e.ErrorCode);
                    Console.WriteLine(e.Message);
                }
            }
            else
            {
                Console.WriteLine("Profile not found.");
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous GetOpportunityà la section Référence des AWS SDK pour .NET API.

Go

Kit SDK pour Go V2

Profitez d’une opportunité.

```
package main

import (
 "context"
 "encoding/json"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/partnercentralselling"
)
```

```
func main() {
 config, err := config.LoadDefaultConfig(context.TODO())

 if err != nil {
  log.Fatal(err)
 }

 config.Region = "us-east-1"

 client := partnercentralselling.NewFromConfig(config)

 output, err := client.GetOpportunity(context.TODO(),
 &partnercentralselling.GetOpportunityInput{
  Identifier: aws.String("O1111111"),
  Catalog:    aws.String("AWS"),
 })

 if err != nil {
  log.Fatal(err)
 }
 log.Println("printing opportuniy...\n")

 jsonOutput, err := json.MarshalIndent(output, "", "    ")

 fmt.Println(string(jsonOutput))
}
```

- Pour plus de détails sur l'API, reportez-vous GetOpportunityà la section Référence des AWS SDK pour Go API.

Java

SDK pour Java 2.x

Profitez d’une opportunité.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
```

```
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityResponse;

/*
 * Purpose
 * PC-API-08 Get updated Opportunity
 */

public class GetOpportunity {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

     GetOpportunityResponse response = getResponse(opportunityId);

     ReferenceCodesUtils.formatOutput(response);
    }

 public static GetOpportunityResponse getResponse(String opportunityId) {

        GetOpportunityRequest getOpportunityRequest =
 GetOpportunityRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .identifier(opportunityId)
          .build();
```

```
        GetOpportunityResponse response =
 client.getOpportunity(getOpportunityRequest);

        return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous GetOpportunityà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Profitez d'une opportunité.

```
#!/usr/bin/env python

"""
Purpose
PC-API -08 Get updated Opportunity given opportunity id
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def get_opportunity(identifier):
    get_opportunity_request ={
        "Catalog": CATALOG_TO_USE,
     "Identifier": identifier
    }
```

```
    try:
        # Perform an API call
        response =
 partner_central_client.get_opportunity(**get_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    identifier = "O5465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Get updated Opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(get_opportunity(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez GetOpportunityle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`ListEngagementInvitations`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `ListEngagementInvitations`.

Java

SDK pour Java 2.x

Extrait une liste des invitations d’engagement envoyées au partenaire.

```
package org.example;

import java.util.ArrayList;
import java.util.List;

import org.example.utils.ReferenceCodesUtils;
import static org.example.utils.Constants.*;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.ListEngagementInvitationsReq
import
 software.amazon.awssdk.services.partnercentralselling.model.ListEngagementInvitationsRes
import
 software.amazon.awssdk.services.partnercentralselling.model.ParticipantType;
import
 software.amazon.awssdk.services.partnercentralselling.model.EngagementInvitationSummary;

public class ListEngagementInvitations {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     List<EngagementInvitationSummary> opportunitySummaries = getResponse();
        ReferenceCodesUtils.formatOutput(opportunitySummaries);
    }

    static List<EngagementInvitationSummary> getResponse() {

  List<EngagementInvitationSummary> opportunitySummaries = new
 ArrayList<EngagementInvitationSummary>();
```

```
  ListEngagementInvitationsRequest listOpportunityRequest =
 ListEngagementInvitationsRequest.builder()
                .catalog(CATALOG_TO_USE)
                .participantType(ParticipantType.RECEIVER)
          .maxResults(5)
          .build();

  ListEngagementInvitationsResponse response =
 client.listEngagementInvitations(listOpportunityRequest);

     opportunitySummaries.addAll(response.engagementInvitationSummaries());

     client.close();

        return opportunitySummaries;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous ListEngagementInvitationsà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Extrait une liste des invitations d’engagement envoyées au partenaire.

```
#!/usr/bin/env python

"""
Purpose
PC-API-21 ListEngagementInvitations - Retrieves a list of engagement invitations
 based on specified criteria.
This operation allows partners to view all invitations to engagement.
"""
import json
import logging
import boto3
import utils.helpers as helper

from utils.constants import CATALOG_TO_USE
```

```
serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def list_engagement_invitations():
    list_engagement_invitations_request ={
        "Catalog": CATALOG_TO_USE,
        "MaxResults": 20
    }
    try:
        # Perform an API call
        response =
 partner_central_client.list_engagement_invitations(**list_engagement_invitations_request
        return response

    except Exception as err:
        # Catch all client exceptions
        print(json.dumps(err.response))

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Retrieve list of Engagement Invitations.")
    print("-" * 88)

    helper.pretty_print_datetime(list_engagement_invitations())

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez ListEngagementInvitationsle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`ListOpportunities`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `ListOpportunities`.

.NET

SDK pour .NET

Répertoriez les opportunités.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// PDX-License-Identifier: Apache-2.0

using System;
using Newtonsoft.Json;
using Amazon;
using Amazon.Runtime;
using Amazon.PartnerCentralSelling;
using Amazon.PartnerCentralSelling.Model;

namespace AWSExample
{
    class Program
    {
        static readonly string catalogToUse = "Sandbox";
        static async Task Main(string[] args)
        {
            // Initialize credentials from .aws/credentials file
            var credentials = new
 Amazon.Runtime.CredentialManagement.SharedCredentialsFile();
            if (credentials.TryGetProfile("default", out var profile))
            {
                AWSCredentials awsCredentials =
 profile.GetAWSCredentials(credentials);

                //var config = new AmazonPartnerCentralSellingConfig()
                //{
                //    ServiceURL = "https://partnercentral-selling.us-
east-1.api.aws",
                //};
                //var client = new
 AmazonPartnerCentralSellingClient(awsCredentials, config);
                var client = new
 AmazonPartnerCentralSellingClient(awsCredentials);
```

```
                var request = new ListOpportunitiesRequest
                {
                    Catalog = catalogToUse,
                    MaxResults = 2
                };

                try {
                    var response = await client.ListOpportunitiesAsync(request);
                    Console.WriteLine(response.HttpStatusCode);
                    foreach (var opportunity in response.OpportunitySummaries)
                    {
                        Console.WriteLine("Opportunity id: " + opportunity.Id);
                    }
                    string formattedJson =
JsonConvert.SerializeObject(response.OpportunitySummaries, Formatting.Indented);
                    Console.WriteLine(formattedJson);
                } catch(ValidationException ex) {
                    Console.WriteLine("Validation error: " + ex.Message);
                } catch (AmazonPartnerCentralSellingException e) {
                    Console.WriteLine("Failed:");
                    Console.WriteLine(e.RequestId);
                    Console.WriteLine(e.ErrorCode);
                    Console.WriteLine(e.Message);
                }
            }
            else
            {
                Console.WriteLine("Profile not found.");
            }
        }
    }
}
```

- Pour plus de détails sur l'API, reportez-vous ListOpportunitiesà la section Référence des AWS SDK pour .NET API.

Go

Kit SDK pour Go V2

Répertoriez les opportunités.

```
package main

import (
 "context"
 "encoding/json"
 "fmt"
 "log"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/partnercentralselling"
)

func main() {
 config, err := config.LoadDefaultConfig(context.TODO())

 if err != nil {
  log.Fatal(err)
 }

 config.Region = "us-east-1"

 client := partnercentralselling.NewFromConfig(config)

 output, err := client.ListOpportunities(context.TODO(),
 &partnercentralselling.ListOpportunitiesInput{
  MaxResults: aws.Int32(2),
  Catalog:    aws.String("AWS"),
 })

 if err != nil {
  log.Fatal(err)
 }

 jsonOutput, err := json.MarshalIndent(output, "", "    ")
 fmt.Println(string(jsonOutput))
}
```

- Pour plus de détails sur l'API, reportez-vous ListOpportunitiesà la section Référence des AWS SDK pour Go API.

Java

SDK pour Java 2.x

Répertoriez les opportunités.

```
package org.example;

import java.util.ArrayList;
import java.util.List;

import org.example.utils.ReferenceCodesUtils;
import static org.example.utils.Constants.*;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.ListOpportunitiesRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.ListOpportunitiesResponse;
import
 software.amazon.awssdk.services.partnercentralselling.model.OpportunitySummary;

/*
 * Purpose
 * PC-API-18 Getting list of Opportunities
 */

public class ListOpportunititesPaging {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {
     List<OpportunitySummary> opportunitySummaries = getResponse();
        ReferenceCodesUtils.formatOutput(opportunitySummaries);
    }
```

```
   private static List<OpportunitySummary> getResponse() {
    List<OpportunitySummary> opportunitySummaries = new
ArrayList<OpportunitySummary>();

 ListOpportunitiesRequest listOpportunityRequest =
ListOpportunitiesRequest.builder()
               .catalog(CATALOG_TO_USE)
         .maxResults(5)
         .build();

    ListOpportunitiesResponse response =
client.listOpportunities(listOpportunityRequest);

    opportunitySummaries.addAll(response.opportunitySummaries());

    while (response.nextToken() != null && response.nextToken().length() > 0) {
     listOpportunityRequest = ListOpportunitiesRequest.builder()
                  .catalog(CATALOG_TO_USE)
            .maxResults(5)
            .nextToken(response.nextToken())
            .build();
     response = client.listOpportunities(listOpportunityRequest);
     opportunitySummaries.addAll(response.opportunitySummaries());
    }

    client.close();

       return opportunitySummaries;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous ListOpportunitiesà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Répertoriez les opportunités.

```
#!/usr/bin/env python

"""
Purpose
PC-API -18 Getting list of Opportunities
"""
import json
import logging
import boto3
import utils.helpers as helper

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def get_list_of_opportunities():

    opportunity_list = []

    list_opportunities_request ={
        "Catalog": CATALOG_TO_USE,
     "MaxResults": 20
    }
    try:
        # Perform an API call
        response =
 partner_central_client.list_opportunities(**list_opportunities_request)
        opportunity_list.extend(response["OpportunitySummaries"])

        while "NextToken" in response and response["NextToken"] is not None:
            list_opportunities_request["NextToken"] = response["NextToken"]
            response =
 partner_central_client.list_opportunities(**list_opportunities_request)
            opportunity_list.extend(response["OpportunitySummaries"])

        return opportunity_list

    except Exception as err:
        # Catch all client exceptions
```

```
        print(json.dumps(err.response))

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Getting list of Opportunities.")
    print("-" * 88)

    helper.pretty_print_datetime(get_list_of_opportunities())

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez ListOpportunitiesle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`ListSolutions`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `ListSolutions`.

Java

SDK pour Java 2.x

Extrait la liste des solutions partenaires que le partenaire a enregistrées sur Partner Central.

```
package org.example;

import java.util.ArrayList;
import java.util.List;

import static org.example.utils.Constants.*;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
```

```
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.ListSolutionsRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.ListSolutionsResponse;
import software.amazon.awssdk.services.partnercentralselling.model.SolutionBase;

/*
 * Purpose
 * PC-API-10 Getting list of solutions
 */

public class ListSolutions {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {
     List<SolutionBase> solutionSummaries = getResponse();
        ReferenceCodesUtils.formatOutput(solutionSummaries);
    }

    static List<SolutionBase> getResponse() {
  List<SolutionBase> solutionSummaries = new ArrayList<SolutionBase>();

  ListSolutionsRequest listSolutionsRequest = ListSolutionsRequest.builder()
    .catalog(CATALOG_TO_USE)
          .maxResults(5)
          .build();

     ListSolutionsResponse response = client.listSolutions(listSolutionsRequest);

     solutionSummaries.addAll(response.solutionSummaries());

        return solutionSummaries;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous ListSolutionsà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Extrait la liste des solutions partenaires que le partenaire a enregistrées sur Partner Central.

```
#!/usr/bin/env python

"""
Purpose
PC-API-10 Getting list of solutions
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def get_list_of_solutions():
    list_solutions_request ={
        "Catalog": CATALOG_TO_USE,
     "MaxResults": 20
    }
    try:
        # Perform an API call
        response =
 partner_central_client.list_solutions(**list_solutions_request)
        return response
```

```
    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Getting list of solutions.")
    print("-" * 88)

    helper.pretty_print_datetime(get_list_of_solutions())

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez ListSolutionsle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`RejectEngagementInvitation`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `RejectEngagementInvitation`.

Java

SDK pour Java 2.x

Rejette une annonce EngagementInvitation qui a AWSété partagée.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
```

```
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.RejectEngagementInvitationRe
import
 software.amazon.awssdk.services.partnercentralselling.model.RejectEngagementInvitationRe

/*
 * Purpose
 * PC-API-05 AWS Originated(AO) rejection
 */

public class RejectEngagementInvitation {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

  RejectEngagementInvitationResponse response = getResponse(opportunityId);

     ReferenceCodesUtils.formatOutput(response);
    }

 static RejectEngagementInvitationResponse getResponse(String invitationId) {

        RejectEngagementInvitationRequest rejectOpportunityRequest =
 RejectEngagementInvitationRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .identifier(invitationId)
          .rejectionReason("Unable to support")
          .build();

  RejectEngagementInvitationResponse response =
 client.rejectEngagementInvitation(rejectOpportunityRequest);
```

```
        return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous RejectEngagementInvitationà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Rejette une annonce EngagementInvitation qui a AWSété partagée.

```
#!/usr/bin/env python

"""
Purpose
PC-API-05 AWS Originated AO rejection - RejectOpportunityEngagementInvitation -
 Rejects a engagement invitation.
This action indicates that the partner does not wish to participate in the
 engagement and
provides a reason for the rejection.
Upon rejection, a OpportunityEngagementInvitationRejected event is triggered.
Subsequently, the invitation will no longer be available for the partner to act
 on.
"""
import json
import logging
import boto3
import utils.helpers as helper

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def reject_opportunity_engagement_invitation(identifier, reject_reason):
```

```
    reject_opportunity_engagement_invitation_request ={
        "Catalog": CATALOG_TO_USE,
     "Identifier": identifier,
        "RejectionReason": reject_reason
    }
    try:
        # Perform an API call
        response =
 partner_central_client.reject_engagement_invitation(**reject_opportunity_engagement_invi
        return response

    except Exception as err:
        # Catch all client exceptions
        print(json.dumps(err.response))

def usage_demo():
    identifier = "arn:aws:partnercentral:us-east-1::catalog/Sandbox/engagement-
invitation/engi-0000002isviga"
    reject_reason = "Customer problem unclear"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Given the ARN identifier and reject reason, reject the Opportunity
 Engagement Invitation.")
    print("-" * 88)


 helper.pretty_print_datetime(reject_opportunity_engagement_invitation(identifier,
 reject_reason))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez RejectEngagementInvitationle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`StartEngagementByAcceptingInvitationTask`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `StartEngagementByAcceptingInvitationTask`.

Java

SDK pour Java 2.x

Commence l'engagement en acceptant un EngagementInvitation.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.StartEngagementByAcceptingIn
import
 software.amazon.awssdk.services.partnercentralselling.model.StartEngagementByAcceptingIn
import
 software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationReque
import
 software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationRespo
import
 software.amazon.awssdk.services.partnercentralselling.model.InvitationStatus;

/*
Purpose
PC-API-04: Start Engagement By Accepting InvitationTask for AWS Originated(AO)
 opportunity
*/

public class StartEngagementByAcceptingInvitationTask {
```

```
static PartnerCentralSellingClient client =
PartnerCentralSellingClient.builder()
           .region(Region.US_EAST_1)
           .credentialsProvider(DefaultCredentialsProvider.create())
           .httpClient(ApacheHttpClient.builder().build())
           .build();

static String clientToken = "test-a30d161";

   public static void main(String[] args) {

    String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

    StartEngagementByAcceptingInvitationTaskResponse response =
getResponse(opportunityId);

    if ( response == null) {
     System.out.println("Opportunity is not AWS Originated.");
    } else {
     ReferenceCodesUtils.formatOutput(response);
    }
   }

   private static GetEngagementInvitationResponse getInvitation(String
invitationId) {

    GetEngagementInvitationRequest getRequest =
GetEngagementInvitationRequest.builder()
         .catalog(Constants.CATALOG_TO_USE)
         .identifier(invitationId)
         .build();

 GetEngagementInvitationResponse response =
client.getEngagementInvitation(getRequest);

       return response;
}

static StartEngagementByAcceptingInvitationTaskResponse getResponse(String
invitationId) {

 if ( getInvitation(invitationId).status().equals(InvitationStatus.PENDING)) {
  StartEngagementByAcceptingInvitationTaskRequest acceptOpportunityRequest =
    StartEngagementByAcceptingInvitationTaskRequest.builder()
```

```
     .catalog(Constants.CATALOG_TO_USE)
           .identifier(invitationId)
           .clientToken(clientToken)
           .build();

   StartEngagementByAcceptingInvitationTaskResponse response =
 client.startEngagementByAcceptingInvitationTask(acceptOpportunityRequest);
         return response;
  }
  return null;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous StartEngagementByAcceptingInvitationTaskà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Commence l'engagement en acceptant un EngagementInvitation.

```
#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
```

```
        region_name='us-east-1'
)

def get_opportunity(identifier):
    get_opportunity_request ={
     "Identifier": identifier,
        "Catalog": CATALOG_TO_USE
    }
    try:
        # Perform an API call
        response =
 partner_central_client.get_engagement_invitation(**get_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def start_engagement_by_accepting_invitation_task(identifier):

    response = get_opportunity(identifier)

    if ( response['Status'] == 'PENDING') :
        accept_opportunity_engagement_invitation_request ={
            "Catalog": CATALOG_TO_USE,
         "Identifier" : identifier,
            "ClientToken": "test-123456"
        }
        try:
            # Perform an API call
            response =
 partner_central_client.start_engagement_by_accepting_invitation_task(**accept_opportunit
            return response

        except ClientError as err:
            # Catch all client exceptions
            print(err.response)
            return None
    else:
        return None

def usage_demo():
    identifier = "arn:aws:partnercentral:us-east-1::catalog/Sandbox/engagement-
invitation/engi-0000002isusga"
```

```
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Get updated Opportunity.")
    print("-" * 88)


 helper.pretty_print_datetime(start_engagement_by_accepting_invitation_task(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez StartEngagementByAcceptingInvitationTaskle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`StartEngagementFromOpportunityTask`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `StartEngagementFromOpportunityTask`.

Java

SDK pour Java 2.x

Lance le processus d'engagement à partir d'une opportunité existante en acceptant l'invitation d'engagement et en créant une opportunité correspondante dans le système du partenaire.

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
```

```
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import software.amazon.awssdk.services.partnercentralselling.model.AwsSubmission;
import
 software.amazon.awssdk.services.partnercentralselling.model.SalesInvolvementType;
import
 software.amazon.awssdk.services.partnercentralselling.model.StartEngagementFromOpportuni
import
 software.amazon.awssdk.services.partnercentralselling.model.StartEngagementFromOpportuni
import software.amazon.awssdk.services.partnercentralselling.model.Visibility;

/*
 * Purpose
 * PC-API-01 Partner Originated (PO) opp submission(Start Engagement From
 Opportunity Task for AO Originated Opportunity)
 */

public class StartEngagementFromOpportunityTask {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

     StartEngagementFromOpportunityTaskResponse response =
 getResponse(opportunityId);

     ReferenceCodesUtils.formatOutput(response);
    }

 static StartEngagementFromOpportunityTaskResponse getResponse(String
 opportunityId) {

  StartEngagementFromOpportunityTaskRequest submitOpportunityRequest =
 StartEngagementFromOpportunityTaskRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .identifier(opportunityId)
          .clientToken("test-annjqwesdsd99")
```

```
 .awsSubmission(AwsSubmission.builder().involvementType(SalesInvolvementType.CO_SELL).vis
          .build();

  StartEngagementFromOpportunityTaskResponse response =
 client.startEngagementFromOpportunityTask(submitOpportunityRequest);

        return response;
 }
}
```

- Pour plus de détails sur l'API, reportez-vous StartEngagementFromOpportunityTaskà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Lance le processus d'engagement à partir d'une opportunité existante en acceptant l'invitation d'engagement et en créant une opportunité correspondante dans le système du partenaire.

```
#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
```

```
        region_name='us-east-1'
)

def start_engagement_from_opportunity_task(identifier):

    start_engagement_from_opportunity_task_request ={
            "AwsSubmission": {
                "InvolvementType": "Co-Sell",
                "Visibility": "Full"
            },
            "Catalog": CATALOG_TO_USE,
         "Identifier" : identifier,
            "ClientToken": "test-annjqwesdsd99"
    }
    try:
            # Perform an API call
            response =
 partner_central_client.start_engagement_from_opportunity_task(**start_engagement_from_op
            return response

    except ClientError as err:
            # Catch all client exceptions
            print(err.response)
            return None

def usage_demo():
    identifier = "O5465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Start Engagement from Opportunity Task.")
    print("-" * 88)


 helper.pretty_print_datetime(start_engagement_from_opportunity_task(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez StartEngagementFromOpportunityTaskle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

À utiliser **`UpdateOpportunity`**avec un AWS Kit SDK

Les exemples de code suivants illustrent comment utiliser `UpdateOpportunity`.

Java

SDK pour Java 2.x

Mettez à jour une opportunité.

```
package org.example;

import java.time.Instant;
import java.util.ArrayList;
import java.util.List;

import static org.example.utils.Constants.*;

import org.example.entity.Root;
import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;
import org.example.utils.StringSerializer;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import software.amazon.awssdk.services.partnercentralselling.model.Account;
import software.amazon.awssdk.services.partnercentralselling.model.Address;
import software.amazon.awssdk.services.partnercentralselling.model.Contact;
import software.amazon.awssdk.services.partnercentralselling.model.Customer;
import
 software.amazon.awssdk.services.partnercentralselling.model.ExpectedCustomerSpend;
import
 software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.GetOpportunityResponse;
import software.amazon.awssdk.services.partnercentralselling.model.LifeCycle;
```

```
import software.amazon.awssdk.services.partnercentralselling.model.Marketing;
import
 software.amazon.awssdk.services.partnercentralselling.model.NextStepsHistory;
import software.amazon.awssdk.services.partnercentralselling.model.Project;
import software.amazon.awssdk.services.partnercentralselling.model.ReviewStatus;
import
 software.amazon.awssdk.services.partnercentralselling.model.UpdateOpportunityRequest;
import
 software.amazon.awssdk.services.partnercentralselling.model.UpdateOpportunityResponse;

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

/*
 * Purpose
 * PC-API-02/06 Update opportunity when LifeCycle.ReviewStatus is not Submitted
 or In-Review
 */

public class UpdateOpportunity {

 static final Gson GSON = new GsonBuilder()
   .setObjectToNumberStrategy(ToNumberPolicy.LAZILY_PARSED_NUMBER)
   .registerTypeAdapter(String.class, new StringSerializer())
   .create();

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

 static String OPPORTUNITY_ORIGIN = ORIGIN_PARTNER_ORIGINATED;

 public static void main(String[] args) {

  String inputFile = "updateOpportunity.json";

  if (args.length > 0)
   inputFile = args[0];

  UpdateOpportunityResponse response = updateOpportunity(inputFile);
```

```
 client.close();
}

public static GetOpportunityResponse getResponse(String opportunityId) {

       GetOpportunityRequest getOpportunityRequest =
GetOpportunityRequest.builder()
   .catalog(Constants.CATALOG_TO_USE)
         .identifier(opportunityId)
         .build();

       GetOpportunityResponse response =
client.getOpportunity(getOpportunityRequest);
       System.out.println(opportunityId + ":" + response);
       return response;
}

public static UpdateOpportunityResponse updateOpportunity(String inputFile) {

 String inputString = ReferenceCodesUtils.readInputFileToString(inputFile);

 Root root = GSON.fromJson(inputString, Root.class);
 GetOpportunityResponse response = getResponse(root.identifier);

 if (response != null
   && response.lifeCycle() != null
   && response.lifeCycle().reviewStatus() != null
   && response.lifeCycle().reviewStatus() != ReviewStatus.SUBMITTED
   && response.lifeCycle().reviewStatus() != ReviewStatus.IN_REVIEW) {

  List<NextStepsHistory> nextStepsHistories = new ArrayList<NextStepsHistory>();
  if ( root.lifeCycle != null && root.lifeCycle.nextStepsHistories != null) {
   for (org.example.entity.NextStepsHistory nextStepsHistoryJson :
root.lifeCycle.nextStepsHistories) {
    NextStepsHistory nextStepsHistory = NextStepsHistory.builder()
      .time(Instant.parse(nextStepsHistoryJson.time))
      .value(nextStepsHistoryJson.value)
                  .build();
    nextStepsHistories.add(nextStepsHistory);
   }
  }

  LifeCycle lifeCycle = null;
```

```
  if ( root.lifeCycle != null ) {
   lifeCycle = LifeCycle.builder()
    .closedLostReason(root.lifeCycle.closedLostReason)
    .nextSteps(root.lifeCycle.nextSteps)
    .nextStepsHistory(nextStepsHistories)
    .reviewComments(root.lifeCycle.reviewComments)
    .reviewStatus(root.lifeCycle.reviewStatus)
    .reviewStatusReason(root.lifeCycle.reviewStatusReason)
    .stage(root.lifeCycle.stage)
    .targetCloseDate(root.lifeCycle.targetCloseDate)
    .build();
  }

  Marketing marketing = null;
  if ( root.marketing != null ) {
   marketing = Marketing.builder()
      .awsFundingUsed(root.marketing.awsFundingUsed)
      .campaignName(root.marketing.campaignName)
      .channels(root.marketing.channels)
      .source(root.marketing.source)
      .useCases(root.marketing.useCases)
      .build();

  }

  Address address = null;
  if (root.customer != null && root.customer.account != null &&
root.customer.account.address != null) {
   address =
Address.builder().postalCode(root.customer.account.address.postalCode)
     .stateOrRegion(root.customer.account.address.stateOrRegion)
     .countryCode(root.customer.account.address.countryCode).build();
  }

  Account account = null;
  if (root.customer != null && root.customer.account != null) {
   account = Account.builder().address(address).duns(root.customer.account.duns)

.industry(root.customer.account.industry).companyName(root.customer.account.companyName)
     .websiteUrl(root.customer.account.websiteUrl).build();
  }

  List<Contact> contacts = new ArrayList<Contact>();
  if ( root.customer != null && root.customer.contacts != null) {
```

```
   for (org.example.entity.Contact jsonContact : root.customer.contacts) {
    Contact contact = Contact.builder()
                  .email(jsonContact.email)
                  .firstName(jsonContact.firstName)
                  .lastName(jsonContact.lastName)
                  .phone(jsonContact.phone)
                  .businessTitle(jsonContact.businessTitle)
                  .build();
    contacts.add(contact);
   }
  }

  Customer customer =
Customer.builder().account(account).contacts(contacts).build();

  List<ExpectedCustomerSpend> expectedCustomerSpends = new
ArrayList<ExpectedCustomerSpend>();
  if ( root.project != null && root.project.expectedCustomerSpend != null) {
   for (org.example.entity.ExpectedCustomerSpend expectedCustomerSpendJson :
root.project.expectedCustomerSpend) {
    ExpectedCustomerSpend expectedCustomerSpend = null;
    expectedCustomerSpend = ExpectedCustomerSpend.builder()
      .amount(expectedCustomerSpendJson.amount)
      .currencyCode(expectedCustomerSpendJson.currencyCode)
      .frequency(expectedCustomerSpendJson.frequency)
      .targetCompany(expectedCustomerSpendJson.targetCompany)
      .build();
    expectedCustomerSpends.add(expectedCustomerSpend);
   }
        }

  Project project = null;
  if (root.project != null) {
   project = Project.builder().title(root.project.title)
     .customerBusinessProblem(root.project.customerBusinessProblem)

.customerUseCase(root.project.customerUseCase).deliveryModels(root.project.deliveryModel
     .expectedCustomerSpend(expectedCustomerSpends)

.salesActivities(root.project.salesActivities).competitorName(root.project.competitorNam
     .otherSolutionDescription(root.project.otherSolutionDescription).build();
  }

  // Building the Actual CreateOpportunity Request
```

```
    UpdateOpportunityRequest updateOpportunityRequest =
 UpdateOpportunityRequest.builder().catalog(root.catalog)

 .identifier(root.identifier).lastModifiedDate(Instant.parse(root.lastModifiedDate))

 .primaryNeedsFromAwsWithStrings(root.primaryNeedsFromAws).opportunityType(root.opportuni
     .lifeCycle(lifeCycle)
     .customer(customer)
     .project(project)
     .partnerOpportunityIdentifier(root.partnerOpportunityIdentifier)
     .marketing(marketing)
     .nationalSecurity(root.nationalSecurity)
     .opportunityType(root.opportunityType)
     .build();

    UpdateOpportunityResponse updateResponse =
 client.updateOpportunity(updateOpportunityRequest);
    System.out.println("Successfully updated opportunity: " + updateResponse);

    return updateResponse;
  } else {
    System.out.println("Opportunity cannot be updated.");
    return null;
  }
     }
}
```

- Pour plus de détails sur l'API, reportez-vous UpdateOpportunityà la section Référence des AWS SDK for Java 2.x API.

Python

Kit SDK for Python (Boto3)

Mettez à jour une opportunité.

```
#!/usr/bin/env python

"""
Purpose
PC-API-2  Updating Partner Originated Opportunity
```

```
"""
import logging
import boto3
import sys
import os
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
import utils.helpers as helper
from botocore.client import ClientError
import utils.stringify_details as sd
from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def get_opportunity(identifier):
    get_opportunity_request ={
     "Identifier": identifier,
        "Catalog": CATALOG_TO_USE
    }
    try:
        # Perform an API call
        response =
 partner_central_client.get_opportunity(**get_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def update_opportunity():
    update_opportunity_request_orig = sd.stringify_json("src/update_opportunity/
update_opportunity_technical_validation.json")
    update_opportunity_request =
 helper.remove_nulls(update_opportunity_request_orig)

    try:
        # Perform an API call
        response =
 partner_central_client.update_opportunity(**update_opportunity_request)
        return response
```

```
    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def update_opportunity_if_eligible(identifier):
    response = get_opportunity(identifier)
    if response is not None:
        return update_opportunity()
    else:
        print("Failed to retrieve opportunity details")

def usage_demo():
    identifier = "O5465588"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Updating opportunity.")
    print("-" * 88)

    helper.pretty_print_datetime(update_opportunity_if_eligible(identifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez UpdateOpportunityle AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Scénarios d'utilisation de Partner Central AWS Kits SDK

Les exemples de code suivants vous montrent comment implémenter des scénarios courants dans Partner Central à l'aide de AWS SDK. Ces scénarios vous montrent comment accomplir des tâches spécifiques en appelant plusieurs fonctions dans Partner Central ou en les combinant avec d’autres Services AWS. Chaque exemple inclut un lien vers le code source complet, où vous trouverez des instructions sur la configuration et l’exécution du code.

Les scénarios ciblent un niveau d'expérience intermédiaire pour vous aider à comprendre les actions de service dans leur contexte.

Exemples

- Mise à jour de l'entité associée à une opportunité

Mise à jour de l'entité associée à une opportunité

Les exemples de code suivants montrent comment :

- dissocier une ancienne entité ;
- associer une nouvelle entité.

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 de scénarios.

Mise à jour de l'entité associée à une opportunité

```
package org.example;

import static org.example.utils.Constants.*;

import org.example.utils.Constants;
import org.example.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import software.amazon.awssdk.regions.Region;
import
 software.amazon.awssdk.services.partnercentralselling.PartnerCentralSellingClient;
import
 software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityRequest;
```

```
import
 software.amazon.awssdk.services.partnercentralselling.model.AssociateOpportunityResponse
import
 software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityReque
import
 software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityRespo

/*
Purpose
PC-API -17 Replacing a solution
*/

public class ReplaceSolution {

 static PartnerCentralSellingClient client =
 PartnerCentralSellingClient.builder()
            .region(Region.US_EAST_1)
            .credentialsProvider(DefaultCredentialsProvider.create())
            .httpClient(ApacheHttpClient.builder().build())
            .build();

    public static void main(String[] args) {

     String opportunityId = args.length > 0 ? args[0] : OPPORTUNITY_ID;

     String entityType = "Solutions";
     String originalEntityIdentifier = "S-0000000";
     String newEntityIdentifier = "S-0011111";

     disassociateOppornitityResponse(opportunityId, entityType,
 originalEntityIdentifier );
     AssociateOpportunityResponse associateOpportunityResponse =
 associateOpportunityResponse(opportunityId, entityType, newEntityIdentifier );

     ReferenceCodesUtils.formatOutput(associateOpportunityResponse);
    }

 private static AssociateOpportunityResponse associateOpportunityResponse(String
 opportunityId, String entityType, String entityIdentifier) {

        AssociateOpportunityRequest associateOpportunityRequest =
 AssociateOpportunityRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
          .opportunityIdentifier(opportunityId)
```

```
            .relatedEntityType(entityType)
            .relatedEntityIdentifier(entityIdentifier)
            .build();

        AssociateOpportunityResponse response =
 client.associateOpportunity(associateOpportunityRequest);

        return response;
 }

 private static DisassociateOpportunityResponse
 disassociateOppornitityResponse(String opportunityId, String entityType, String
 entityIdentifier) {
  PartnerCentralSellingClient client = PartnerCentralSellingClient.builder()
             .region(Region.US_EAST_1)
             .credentialsProvider(DefaultCredentialsProvider.create())
             .httpClient(ApacheHttpClient.builder().build())
             .build();

        DisassociateOpportunityRequest disassociateOpportunityRequest =
 DisassociateOpportunityRequest.builder()
    .catalog(Constants.CATALOG_TO_USE)
           .opportunityIdentifier(opportunityId)
           .relatedEntityType(entityType)
           .relatedEntityIdentifier(entityIdentifier)
           .build();

        DisassociateOpportunityResponse response =
 client.disassociateOpportunity(disassociateOpportunityRequest);

        return response;
 }
}
```

- Pour plus de détails sur l'API, consultez les rubriques suivantes dans la Référence des API du kit AWS SDK for Java 2.x.
  - AssociateOpportunity
  - DisassociateOpportunity

Python

Kit SDK for Python (Boto3)

> **Note**
>
> Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le référentiel d'exemples de code AWS.

Mise à jour de l'entité associée à une opportunité

```
#!/usr/bin/env python

"""
Purpose
PC-API -17 Replacing a solution
"""
import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
        service_name=serviceName,
        region_name='us-east-1'
)

def replace_solution(original_entity_identifier, new_entity_identifier,
 opportunityIdentifier):
    disassociate_opportunity_request ={
        "Catalog": CATALOG_TO_USE,
     "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : "Solutions",
        "RelatedEntityIdentifier" : original_entity_identifier
    }

    associate_opportunity_request ={
```

```
        "Catalog": CATALOG_TO_USE,
     "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : "Solutions",
        "RelatedEntityIdentifier" : new_entity_identifier
    }
    try:
        # Perform an API call
        response =
 partner_central_client.disassociate_opportunity(**disassociate_opportunity_request)
        response =
 partner_central_client.associate_opportunity(**associate_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)

def usage_demo():
    original_entity_identifier = "S-0049999"
    new_entity_identifier = "S-0050014"
    opportunityIdentifier = "O4397574"

    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Replacing a solution.")
    print("-" * 88)

    helper.pretty_print_datetime(replace_solution(original_entity_identifier,
 new_entity_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()
```

- Pour plus de détails sur l'API, consultez les rubriques suivantes dans la Référence des API du kit AWS SDK for Python (Boto3).
  - AssociateOpportunity
  - DisassociateOpportunity

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultezUtilisation AWS API Partner Central avec un AWS Kit SDK. Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes du kit SDK.

Notes de mise à jour

Cette page contient un résumé des modifications importantes apportées à la documentation de l'API AWS Partner Central, en commençant par la dernière mise à jour.

Historique des révisions

Modifier	Description	Date
RC 6.0	• AWS Partner Central a ajouté de nouvelles API de mesure des revenus des partenaires.	30/06/2026/
RC 5.3	• Mises à jour d'ACE Resonate : `GetAwsOpportunitySummary` réponse mise à jour.	16/06/2026/
RC 5.2	• Support de devise en EUR : `CreateOpportunity` `UpdateOpportunity` ,, et `CreateEngagementInvitation` maintenant j'accepte EUR en plus du code USD de devise MRR. Lorsque la partition AWS est `aws-eusc` (AWS European Sovereign Cloud), le code de devise doit êtreEUR. • EUR dans les opérations de lecture : `GetOpportunity` `ListOpportunities` ,, et revenez `GetAwsOpportunitySummary` EUR maintenan	30/03/2026/

Modifier	Description	Date
	t `CurrencyCode` sur le terrain pour des opportunités dans la `aws-eusc` partition.	
RC 5.1	• Support de devise en EUR : `CreateOpportunity` `UpdateOpportunity` ,, et `CreateEngagementInvitation` maintenant j'accepte EUR en plus du code USD de devise MRR. Lorsque la partition AWS est `aws-eusc` (AWS European Sovereign Cloud), le code de devise doit êtreEUR. • EUR dans les opérations de lecture : `GetOpportunity` `ListOpportunities` ,, et revenez `GetAwsOpportunitySummary` EUR maintenant `CurrencyCode` sur le terrain pour des opportunités dans la `aws-eusc` partition.	30/03/2026/

Modifier	Description	Date
RC 5	• API de compte introduite : ajout d'une nouvelle API de compte àAWS Partner Central, permettant aux partenaires de gérer les informations de leur compte, leur enregistrement, leurs profils, leur gestion de domaine et leurs connexion s avec les partenaires. • Documentation de référence de l'API du compte : ajout d'une documentation de référence complète pour l'API du compte, y compris l'enregistrement des partenaires, la gestion des profils, la vérification du domaine et les flux de travail de connexion aux comptes. • Enregistrement de l'API du compte : ajout d'un support de journalisation et d'une documentation pour l'API de compte afin de faciliter l'audit et le suivi de l'utilisa tion de l'API. • Notifications de l'API du compte : ajout de la prise en charge des notifications d'événements pour l'API du compte, permettant aux partenaires de recevoir des mises à jour en temps	30/11/2025

Modifier	Description	Date
	réel sur les activités liées au compte, y compris les connexions aux comptes partenaires. • Quotas d'API de compte : ajout de la documentation sur les quotas de service pour l'API de compte. • Tests en sandbox pour l'API des comptes : ajout de fonctionnalités de test en sandbox pour l'API des comptes, permettant aux partenaires de tester les flux de travail de gestion des comptes dans un environnement hors production. • API d'avantages ajoutée : ajout d'une nouvelle API d'avantages àAWS Partner Central, permettant aux partenaires de découvrir, de demander et de gérer les avantages dans l'ensemble des programmes de AWS partenariat via une interface centralisée. • Documentation de référence de l'API Benefits : Ajout d'une documentation de référence complète pour l'API Benefits, couvrant les demandes d'avantages, les	

Modifier	Description	Date
	allocations et les processus d'exécution. • Journalisation de l'API Benefits : ajout d'un support de journalisation et d'une documentation pour l'API Benefits afin de faciliter l'audit et le suivi de l'utilisation de l'API. • Notifications de l'API Benefits : Ajout de la prise en charge des notifications d'événements pour l'API Benefits, permettant aux partenaires de recevoir des mises à jour en temps réel sur les demandes d'avantages et les allocations. • Quotas de l'API Benefits : ajout de la documentation sur les quotas de service pour l'API Benefits. • Tests en sandbox pour l'API Benefits : ajout de fonctionnalités de test en sandbox pour l'API Benefits, permettant aux partenaires de tester les flux de travail de gestion des avantages dans un environnement hors production. • API de vente mise à jour : actions d'engagement de l'API de vente améliorée	

Modifier	Description	Date
	s pour soutenir le flux de travail de gestion des prospects. • Mises à jour de la documentation de l'API de vente : documentation de l'API de vente améliorée avec une structure améliorée et des flux de travail actualisés pour la gestion des prospects. • Mises à jour des notificat ions de l'API de vente : documentation améliorée sur les notifications d'événements pour les événements d'engagement et d'invitation à l'engagem ent. • Extension de la structure des API : structure de documentation étendue pour prendre en charge quatre API distinctes (compte, avantages, vente et canal) avec des sections dédiées pour les autorisat ions, la journalisation, les notifications, les quotas et les tests en sandbox de chaque API. • Documentation améliorée sur l'utilisation des API : ajout de guides d'utilisa	

Modifier	Description	Date
	tion complets pour les API Account and Benefits avec des flux de travail détaillés et des modèles d'intégration. • Documentation sur les régions prises en charge : ajout de la documentation pour les AWS régions prises en charge pour les API Account and Benefits. • Nouveau filtre pour `ListOpportunities` : Ajout d'un `TargetCloseDate` filtre pour permettre de filtrer les opportunités en fonction de leur date de clôture prévue à `AfterTargetCloseDate` l'aide de `BeforeTargetCloseDate` champs YYYY-MM-DD formatés.	

Modifier	Description	Date
RC 4	• API de canal introduite : ajout d'une nouvelle API de canal àAWS Partner Central, permettant aux partenaires de gérer les relations entre les canaux et les collaborations entre partenaires. • Documentation de référence de l'API Channel : ajout d'une documentation de référence complète pour l'API Channel, y compris toutes les actions et tous les modèles de données disponibles. • Autorisations de l'API du canal : ajout de politique s d'autorisation IAM et de documentation de contrôle d'accès spécifiques à l'API du canal. • Journalisation de l'API du canal : ajout du support de journalisation et de la documentation pour l'API du canal afin de faciliter l'audit et le suivi de l'utilisation de l'API. • Notifications de l'API du canal : ajout de la prise en charge des notifications d'événements pour l'API du canal, permettant aux	19/11/2025

Modifier	Description	Date
	partenaires de recevoir des mises à jour en temps réel sur les activités liées au canal. • Quotas d'API de canal : ajout de la documentation sur les quotas de service pour l'API de canal. • Tests en sandbox pour l'API Channel : ajout de fonctionnalités de test en sandbox pour l'API Channel, permettant aux partenair es de tester les flux de travail des canaux dans un environnement hors production. • Réorganisation de la structure de l'API : documentation réorganis ée pour faire la distinction entre l'API Selling et l'API Channel, avec des sections dédiées aux autorisations, à la journalisation, aux notifications et aux quotas de chaque API. • Documentation sur les régions prises en charge : ajout de la documentation pour AWS les régions prises en charge pour les API de vente et de distribution.	

Modifier	Description	Date
RC 3	• Politiques d'autorisation d'API mises à jour : les politiques ont été mises à jour pour refléter les nouvelles actions introduites dans cette version. Assurez-vous que vos rôles et autorisations IAM sont ajustés en conséquence. • Détails d'utilisation de l'en-tête personnalisé : ajout de détails sur la façon d'utiliser l'en-tête personnalisé `X-Amzn-User-Agent` pour aider à auditer et à mesurer l'utilisation de l'API. • Actions remplacées par des actions asynchrones : Remplacé par `AcceptOpportunityEngagementInvitation` l'action asynchrone `StartEngagementByAcceptingInvitationTask` et le format de charge utile a changé par rapport à RC2. • Changement de nom de l'entité et de l'action : remplacé par`0pportunityEngagementInvitation` . `EngagementInvitation` Les attributs de ces actions ont également été modifiés pour	17/10/2024

Modifier	Description	Date
	s'aligner sur la nouvelle entité. Les actions suivantes ont été remplacées : • `GetOpportunityEngagementInvitation` c'est maintenant `GetEngagementInvitation` . Plusieurs attributs de l'invitation ont été modifiés, ajoutés ou supprimés. • `ListOpportunityEngagementInvitations` c'est maintenant `ListEngagementInvitations` . • `RejectOpportunityEngagementInvitation` c'est maintenant `RejectEngagementInvitation` . • `AssignOpportunity` mise à jour des paramètres : prend désormais la `Assignee` place de `AssigneeEmail` . Mettez à jour vos implémentations en conséquence. • `SubmitOpportunity` changement de fonctionnalité : `SubmitOpportunity` est désormais	

Modifier	Description	Date
	effectué par l'action asynchrone. `StartEngagementFromOpportunityTask` `SubmitOpportunity` accepte désormais `InvolvementType` et`Visibility` . • Renommage et extension des attributs : `EstimatedAwsMonthlyRevenue` remplacé par. `ExpectedCustomerSpend` `ExpectedCustomerSpend` inclut désormais de nouveaux attributs, tels que `Frequency` et`TargetCompany` . • Résumé des nouvelles actions pour les AWS opportunités : la visibilité des opportunités est désormais assurée par le biais d'une action distincte appeléee`GetAwsOpportunitySummary` . • Mises à jour du flux de travail pour travailler avec les opportunités : le flux de travail pour travailler avec vos opportunités a été mis à jour pour utiliser les nouvelles actions introduites dans cette version.	

Modifier	Description	Date
	• Mises à jour du flux de travail pour l'utilisation des opportunités : le flux de travail relatif à l'utilisation des opportunités a été mis à jour pour être utiliséEngagementInvitations , ce qui améliore la gestion des opportunités proposées par AWS. • Mises à jour du flux de travail pour le suivi des mises à jour : le flux de travail de suivi des mises à jour fonctionne désormais avec les mises à jour des opportunités, ce qui améliore la clarté et l'efficacité du suivi de l'état des opportunités. • Nouveaux filtres pour répertorier les actions : de nouveaux filtres sont disponibles pour ListSolutions et pour ListOpportunities les actions. • En ListSolutions effet, les nouveaux filtres sont FilterCategory FilterIdentifier , etFilterStatus .	

Modifier	Description	Date
	• En `ListOpportunities` effet, les nouveaux filtres sont `FilterCustomerCompanyName` `FilterIdentifier` `FilterLastModifiedDate` ,`FilterLifeCycleApprovalStatus` ,`FilterLifeCycleStage` , et`FilterOrigin` . • Suppression du `AccessControl` sous-objet : le `AccessControl` sous-objet du modèle d'opportunité a été supprimé. `NationalSecurity` est désormais un attribut de premier niveau. • Attributs déplacés vers `GetAwsOpportunitySummary` : Attributs tels que `AWSOpportunityTeam` insights (`EngagementScore` ,`NextBestAction`), origin, et `InvolvementType` sont désormais disponibles via `GetAwsOpportunitySummary` au lieu de`GetOpportunity` .	

Modifier	Description	Date
	• Présentation du `InvolvementType` champ : introduction du champ pour différencier les `InvolvementType` opportunités de cosell et celles axées uniquement sur la visibilité. • Mises à jour du nom des événements : les événements ont été mis à jour pour refléter les modifications. Par exemple, l'invitation d'engagement d'opportunité créée est désormais une invitation d'engagement créée. • Mises à jour de la charge utile de réponse : toutes les charges utiles de réponse incluent désormais un identifiant ARN and/or . • Mises à jour de la charge utile des demandes : toutes les charges utiles des demandes utilisent désormais des identifiants en entrée, qui acceptent soit un ID, soit un ARN. • Mise à jour de l'entité d'engagement : l'attribut title de l'entité d'engagement a été renommé`EngagementTitle` .	

Modifier	Description	Date
	• Suppression du préfixe de filtre : tous les filtres n'incluent plus le préfixe de filtre, ce qui simplifie le mécanisme de filtrage. • `StartEngagementFromOpportunity` mise à jour de l'action : l'attribut `AwsInvolvement` a été remplacé par`AwsSubmission` . • Mise à jour des détails du projet d'invitation : `Invitation.ProjectDetails.TargetCompletionDate` a été renomméé`Invitation.ProjectDetails.EstimatedCompletionDate` . • Mise à jour de l'estimation des recettes : `Invitation.PartnerRevenueEstimate` c'est maintenant`ExpectedCustomerSpend` . Le type de données est passé d'un objet à un tableau contenant un seul objet. • Mise à jour du compte du destinataire : le champ `ReceiverAccount` a été renommé`AccountReceiver` .	

Modifier	Description	Date
	• Nouvel attribut d'invitation à un engagement : `SenderContact` un attribut a été introduit pour les invitations à un engagement. • `Customer.Contact` est désormais un tableau au lieu d'un objet. `OpportunityOwner` , `PartnerAccountManager` sont des valeurs prises en charge sous la forme`BusinessTitle` . • `PartnerOpportunityTeam` est désormais appelé `OpportunityTeam` . • `APNPrograms` est maintenant `ApnPrograms` • Les statuts des invitations à un engagement ont été mis à jour. • `ListEngagementInvitations` nécessite désormais `ParticipantType` as`RECEIVER`.	

Modifier	Description	Date
RC 2	• A introduit une nouvelle entité appelée OpportunityEngagementInvitation. • A introduit une nouvelle action : ListOpportunityEngagementInvitations. • A introduit une nouvelle action : GetOpportunityEngagementInvitation. • A introduit un nouvel événement : « Opportunity Engagement Invitation Created ». • Remplacé AcceptOpportunity par AcceptOpportunityEngagementInvitation. • Remplacé RejectOpportunity par RejectOpportunityEngagementInvitation. • « Opportunité acceptée » a été remplacé par « Invitation d'engagement d'opportunité acceptée ». • « Opportunité rejetée » par « Invitation d'engagement d'opportunité rejetée ». • `PrimaryNeedsFromAWS` changé en PrimaryNeedsFromAws.	2024-08-07

Modifier	Description	Date
	• `AWSOpportunityTeam` changé en`AwsOpportunityTeam` . • `AWSSalesLifeCycle` changé en`AwsSalesLifeCycle` . • `PrimaryNeedsFromAWSChangeReason` changé en`PrimaryNeedsFromAwsChangeReason` . • `AWSAccountId` changé en`AwsAccountId` . • `ExpectedMonthlyAWSRevenue` changé en`ExpectedMonthlyAwsRevenue` . • `AWSFundingUsed` changé en`AwsFundingUsed` . • `AWSMarketplaceOffers` changé en`AwsMarketplaceOffers` . • `Country`changé en`CountryCode` . • `PartnerOpportunityContact` changé en`PartnerOpportunityTeam` . • Champ mis à jour `Contact.Title` pour`Contact.BusinessTitle` .	

Modifier	Description	Date
	• Champ mis à jour `ContactInfo` pour`OwnerInfo` . • AWS L'équipe est passée d'une carte à une liste. • Ajout de détails sur la liste des personnes acceptées RejectionReasons. • Informations fournies sur la façon d'utiliser le jeton client. • Détails d'utilisation inclus UpdateOpportunity. • Ajout de conseils sur l'utilisation AWS des produits et des solutions partenaires. • Informations détaillées sur l' OpportunityEngagementInvitationentité fournies. • Mis StateOrRegion à jour pour inclure une liste révisée des valeurs acceptées. • Plusieurs corrections de bogues ont été mises en œuvre pour améliorer les performances et les messages d'erreur.	

Modifier	Description	Date
RC 1	• [Rupture] Renommer ApprovalStatus le champ en ReviewStatus : nous avons renommé le ApprovalStatus champ ReviewStatus afin de mieux refléter son objectif. En outre, nous introduis ons un nouveau statut, En attente de soumission, pour indiquer une opportunité de brouillon. Les valeurs suivantes ReviewSta tus seront désormais acceptées : En attente de soumission, Soumis, En cours de révision, Approuvé, Rejeté et Action requise. L'ancien statut de brouillon sera remplacé par le statut En attente de soumission. • [Rupture] Introduction de SubmitOpportunity l'action : Une nouvelle action a été introduite. SubmitOpp ortunity Contrairement au comportement actuel où l'opportunité est CreateOpp ortunity également soumise, vous pouvez désormais créer une opportunité à l'état En attente de soumission sans lier immédiatement une solution ou un AWS produit. Pour terminer	21/06/2021

Modifier	Description	Date
	la soumission, utilisez l' AssociateOpportunity action pour lier une solution ou un produit, puis SubmitOpp ortunity. Cette séparatio n permet de meilleures performances de réponse aux API et une phase de préparation flexible avant la soumission. • [Cause] Exigence d'un paramètre de catalogue et dépréciation du point de terminaison Sandbox : le point de terminaison « gamma » (partnerc entral-selling-gamma.us-east-1.amazonaws.com) ne sera plus disponible après /2024. 7/30 Toutes les actions d'API nécessite nt désormais un paramètre Catalog pour spécifier si l'opération est effectuée sur le Sandbox ou sur le AWS catalogue. Les règles de notification filtreront désormais sur le catalogue plutôt que sur Environme ntName. • [Cause] Renommer OpportunityIdentifier en identifiant : actions AssignOpportunity	

Modifier	Description	Date
	AcceptOpportunity, et RejectOpportunity maintenant prenez un identifiant au lieu de OpportunityIdentifier pour rester cohérent avec UpdateOpportunity. • Passage du type Enum à String pour APNProgram CustomerUseCase, et UseCase : Nous allons convertir Project.APNPrograms Marketing.ActivityUseCases, et Project.CustomerUseCase du type d'énumération au type de chaîne afin de réduire les risques associés à la modification des valeurs. Ces champs n'accepteront que les valeurs issues de listes prédéfinies, mais qui ne sont pas disponibles de manière native dans les SDK. • Gestion améliorée des erreurs : La gestion des erreurs au sein des API a été considérablement améliorée pour faciliter le débogage et accélérer la résolution des problèmes. Le nouveau système	

Modifier	Description	Date
	de gestion des erreurs structure les erreurs en deux étapes : 1. Échec de la validation de la demande : vérification des types et du format des données, ou 2. Échec de la validatio n commerciale : tous les problèmes liés à une charge utile seront répertoriés dans une seule réponse, en spécifiant le champ contenant des erreurs. Ces commentaires consolidé s vous permettent de résoudre et de corriger les erreurs de manière plus efficace. Les codes d'erreur et les formats ont été normalisés. • Identifiant de la source du client : possibilité pour les partenaires d'inclure l' X-Amzn-User-Agent en-tête dans leurs demandes, ce qui permet d'identifier la source du trafic. Utilisez le format [Nom de la solution, fournisseur de solution inclus] | [Version] Par exemple :AWS Partner CRM Connector | v2.0.1 • Correction d'un bogue, déjà déployée, aucune modificat	

Modifier	Description	Date
	ion du SDK] Création d'opportunité et ID de AWS compte : les partenair es peuvent désormais lancer une opportunité en changeant l'ID de AWS compte de zéro à une valeur valide sans rencontre r d'erreur. Auparavant, un bogue empêchait les partenaires d'effectuer ces actions séparément. • Correction d'un bogue, déjà déployée, aucune modification apportée au SDK] Solution associée à de nouvelles opportunités : les partenaires peuvent associer une solution à une opportunité nouvellement créée. Auparavant, lorsque les partenaires créaient une nouvelle opportunité et associaient simultané ment une solution, les informations relatives à la solution étaient perdues. Ce problème a été résolu. • Correction d'un bogue, déjà déployée, aucune modification apportée au SDK] Affichage des détails du propriétaire de l'opportu nité : le problème selon	

Modifier	Description	Date
	lequel certaines opportunités affichaient les informations du propriétaire dans un champ combiné LastName « » au lieu des champs « E-mail », FirstName « » et « » LastName attendus a été corrigé. • Correction de bogue, déjà déployée, aucune modification du SDK] Customer. Account.WebsiteUrl Champ facultatif : le Customer. Account.WebsiteUrl champ est devenu facultatif lorsque le AccessControls.NationalSecurity champ est « Oui », alignant ainsi le comportement de l'API sur le comportement de l'interface utilisateur existant. Si le AccessControls.NationalSecurity champ est « Non » ou nul, le WebsiteUrl champ sera requis pour la validation commerciale. • Correction d'un bogue, déjà déployée, aucune modification de l'exigence d'identifiant de AWS compte [SDK] pour les partenaires consultants : l'incohérence dans l'interface utilisateur	

Modifier	Description	Date
	qui indiquait que le champ ID de AWS compte était facultatif pour les partenair es consultants a été corrigée. Il s'agit désormais d'un champ obligatoire conditionnel, conformément à la documentation. • Correction d'un bogue, à déployer 6/19, aucune modification du SDK] Étape d'une Co-sell opportuni té à Fermée Perdue : Les partenaires peuvent désormais passer avec succès de l'étape d'une Co-sell opportunité à Fermée perdue, à condition qu'une raison de fermeture perdue soit fournie. Cela corrige l'erreur précédente indiquant « Champ obligatoi re manquant : fermé La raison perdue est requise lors de la fermeture de l'opportunité ». • Documentation] Améliorat ions apportées à la documentation : plusieurs améliorations ont été apportées à la documenta tion pour refléter les modifications.	

Modifier	Description	Date

	• Problèmes connus avec un correctif en attente et liste des fonctionnalités à venir :
	  • Lorsque vous effectuez une AssociateOpportunity action avec des solutions ou des produits, les messages d'erreur des offres ne s'affichent pas correctement.
	  • Possibilité de créer des opportunités à distance et de simuler le processus AWS de validation dans Sandbox
	  • Le code postal renvoie la valeur null lorsqu'il est XXXXX-XXXX au format américain.
	  • Next Steps et Next Step History ne sont pas disponibles pour les opportunités marquées comme « Pas besoin d'assistance de »AWS.
	  • Impossible d'obtenir l'historique des opportunités avec des descriptions de problèmes commerciaux de plus de 255 caractères pour les clients.	

Modifier	Description	Date
	• Possibilité de supprimer les opportunités en attente de soumission. • LeadSource n'est pas disponible dans le modèle de données.	
Bêta 5	• Changé LifeCycle.Approval Status en LifeCycle .ReviewStatus • Changé Insights. ReviewComments en LifeCycle.ReviewComments • Ajouté LifeCycle.ReviewStatusReason (nouvel attribut) • Nouvelle valeur ajoutée « En attente » pour PartnerAcceptance.Status • SDK publiés pour dotnet, java, javascript, python, ruby. • Guide de référence des API mis à jour pour refléter les modifications ci-dessus.	19/04/2024

Modifier	Description	Date
Bêta 4	• Test Sandbox ajouté : possibilité de tester la notification des événements d'opportunité dans Sandbox. • Documentation ajoutée : introduction d'exemples de règles de notification d'événements d'opportunité. • Versions de SDK ajoutées : SDK publiés pour Java (V2), Javascript (V2), DotNet (V3) et Python (V3). • Format de date mis à jour : mis à jour pour suivre les normes ISO.	04/03/2024

Modifier	Description	Date
Bêta 3	• Ajout d'un fichier journal des modifications : introduction d'un fichier journal des modifications pour un meilleur suivi des mises à jour et des modifications. • Guide de référence des API mis à jour (PDF) : publication d'une nouvelle version du guide de référence des API au format PDF. • Guide de référence des API mis à jour (version Web) : mise à jour de la version Web du guide de référence des API. • DotNet SDK ajouté : lancement du DotNet SDK, étendant notre support à différents environnements de programmation. • Mise à jour du SDK Python Boto 3 (V2) : sortie de la version 2 du SDK Python Boto 3, introduisant de nouvelles fonctionnalités et améliorations.	24-01-2024

Modifier	Description	Date
Bêta 2	• Ajout du guide de référence des API (version Web) : publication d'une nouvelle version du guide de référence des API (version Web). Remarque : Cette version nécessite l'exécuti on d'un serveur Web local après l'extraction pour un fonctionnement correct.	07-01-2024
Bêta 1	• Ajout de Boto3 (version initiale) : lancement de la version initiale de Boto3, un SDK Python pour nos services. • Ajout du guide du programme bêta : introduct ion d'un guide pour les participants à notre programme de test bêta. • Ajout du guide de référence des API (PDF) : nous avons téléchargé la première version de notre guide de référence des API au format PDF, fournissant une documentation complète pour notre API.	15/12/2023

Modifier	Description	Date
Bêta 0	Cette version initiale fournit une suite de fonctionnalités permettant de gérer et d'optimiser les engagemen ts et les opportunités des AWS partenaires dans Partner Central.	15/11/2023

Historique du document

Vous trouverez ci-dessous une liste des publications de documentation pour cette référence d'API.

Modification	Description	Date
Documentation ajoutée pour les API de mesure des revenus des partenaires	AWS Partner Central a ajouté de nouvelles API pour la mesure des revenus des partenaires (PRM).	30 juin 2026
Documentation mise à jour pour ACE Resonate	AWS Partner Central a mis à jour la réponse de l'API pour GetAwsOpportunityS ummary prendre des mesures en faveur de Cosell Motion.	16 juin 2026
Documentation ajoutée pour le serveur MCP Partner Central	AWS Le serveur MCP des agents centraux partenair es fournit des outils centralis és aux partenaires via le protocole MCP (Model Context Protocol).	16 mars 2026
Version RC 5	Sortie de la version Candidate 5 de l'API AWS Partner Central. Une nouvelle API de compte a été introduite dans AWS Partner Central, qui permet aux partenair es de gérer les informati ons de leur compte, leur enregistrement, leurs profils, leur gestion de domaine et leurs connexions avec les partenaires. A introduit une nouvelle API d'avantages	30 novembre 2025

	dans AWS Partner Central, permettant aux partenaires de découvrir, de demander et de gérer les avantages dans l'ensemble des programmes de AWS partenariat via une interface centralisée. Actions d'engagement améliorées de l'API de vente pour soutenir le flux de travail de gestion des prospects. Ajout de guides d'utilisation complets et d'une documentation d'API pour les API Account, Benefits avec des flux de travail détaillés et des modèles d'intégration. Documentation améliorée de l'API de vente pour la gestion de l'engagement avec le contexte des prospects.

Version RC 4	Sortie de la version Candidate 4 de l'API AWS Partner Central. Introduction de l'API Channel pour gérer les relations entre les canaux et les collaborations entre partenaires, permettant aux partenaires de bénéficier des avantages du programme lorsqu'ils effectuent des transactions avec des clients finaux à l'aide du transfert de facturation. La documenta tion a été réorganisée pour prendre en charge plusieurs API AWS Partner Central.	19 novembre 2025
Documentation ajoutée pour les opportunités multi-par tenaires	Multi-partner opportunities (en version préliminaire) est une expérience intégrée dans AWS Partner Central qui permet aux AWS partenaires de trouver d'autres partenair es, de communiquer avec eux et de co-vendre ensemble des offres clients.	4 décembre 2024
Version RC 3	Sortie de la version Candidate 3 de l'API AWS Partner Central	17 octobre 2024
Version RC 2	Sortie de la version Candidate 2 de l'API AWS Partner Central	7 août 2024

Version RC 1	Sortie de la version Candidate 1 de l'API AWS Partner Central	21 juin 2024
Sortie de la version bêta 5	Sortie de la version bêta 5 de l'API AWS Partner Central	19 avril 2024
Sortie de la version bêta 4	Sortie de la version bêta 4 de l'API AWS Partner Central	4 mars 2024
Sortie de la version bêta 3	Sortie de la version bêta 3 de l'API AWS Partner Central	24 janvier 2024
Sortie de la version bêta 2	Sortie de la version bêta 2 de l'API AWS Partner Central	7 janvier 2024
Version bêta 1	Sortie de la version bêta 1 de l'API AWS Partner Central	15 décembre 2023
Sortie de la version bêta 0	AWS Version initiale de l'API Partner Central	15 novembre 2023