



Entwicklerhandbuch

AWS Partner Central



AWS Partner Central: Entwicklerhandbuch

Table of Contents

.....	vi
Willkommen	1
Arbeiten mit AWS SDKs	1
Einrichtung und Authentifizierung	2
Verknüpfen Sie Ihre AWS-Konto mit Partner Central	2
Einrichten von IAM	2
Authentifizierung von API-Aufrufen	3
Signieren Sie Ihre Anrufe mit einem benutzerdefinierten Benutzeragenten	5
MCP-Server für Partner Central-Agenten	7
-Übersicht	7
Funktionen der Agenten	7
Wichtigste Vorteile	8
Verwendungsbeispiele	9
Schaffung von Chancen	9
Klonen von Geschäftschancen	9
Einblicke in die Pipeline	10
Zusammenfassung der Geschäftschancen	10
Verkäufe spielen bei der Generierung	10
Erstellung eines Kundenprofils	10
Empfehlung für eine Lösung	11
Empfehlung für eine Finanzierung	11
Empfehlungen für den nächsten Schritt	11
Weiterentwicklung der Geschäftschance	12
Agenturerfahrung für das Onboarding von Partnern	12
Automatisierung von Partnerprofilen	13
Anleitung zur Einrichtung für Verkäufer	13
Richtlinien zur Einhaltung von PRM-Vorschriften	14
Erste Schritte	14
Erste Schritte	14
Voraussetzungen	14
Schritt 1: Richten Sie IAM-Berechtigungen ein	15
Schritt 2: Connect Ihren MCP-Client	22
Signieren Sie Ihre Anrufe mit dem MCP-Header	23
Schritt 3: Überprüfen Sie Ihr Setup	25

Schritt 4: Führen Sie Ihre ersten Aufgaben aus	26
Sicherheitsüberlegungen	28
Nächste Schritte	28
Konfigurationsreferenz	28
Endpoint	29
IAM-Berechtigungen	29
Katalogumgebungen	34
Verwaltung von Sitzungen	34
Datei-Upload	35
Fehlercodes	36
Ratenbegrenzungen	36
Streaming-Ereignistypen (SSE)	37
Nächste Schritte	38
Referenz zu Tools	38
Überblick über die Tools	38
Nachricht senden	39
GetSession	47
Fehlerbehandlung	49
Über die APIs	51
Verwenden der APIs	51
Verwenden der Konto-API	51
Verwenden der Verkaufs-API	66
Verwenden der Benefits API	125
Verwenden der Channel-API	140
Verwendung der API zur Umsatzmessung	142
Unterstützte Regionen	155
Regionen für die Account-API	155
Regionen für die Verkaufs-API	155
Regionen für die Benefits API	155
Regionen für die Channel-API	155
Regionen für die API zur Umsatzmessung	156
Berechtigungen	157
Zugriff für die Konto-API	158
Zugriff für die Verkaufs-API	161
Zugang für die Benefits API	168
Zugriff für die Channel-API	170

Zugriff für die Revenue Measurement API	179
Kontingente	188
Kontingente für die Account-API	189
Kontingente für die Verkaufs-API	192
Kontingente für die Benefits API	195
Kontingente für die Channel-API	197
Kontingente für die API zur Umsatzmessung	201
Protokollierung	203
Protokollierung der Konto-API	204
Protokollierung der Verkaufs-API	207
Protokollierung der Benefits API	209
Protokollierung der Channel-API	211
Protokollierung der API zur Umsatzmessung	214
Benachrichtigungen	217
Benachrichtigungen für AWS Konto-API für Partner Central	217
Benachrichtigungen für die AWS Vertriebs-API von Partner Central	230
Benachrichtigungen für AWS API für Vorteile von Partner Central	246
Benachrichtigungen für AWS Channel-API für Partner Central	260
Testen in einer Sandbox	270
Zugriff auf die Sandbox-Umgebung	270
Wichtige Details zur Sandbox-Umgebung	270
Testen in einer Sandbox für die Account-API	271
Testen in einer Sandbox für die Verkaufs-API	274
Testen der Benefits API in einer Sandbox	279
Testen in einer Sandbox für die Channel-API	283
Testen der API zur Umsatzmessung in einer Sandbox	287
Codebeispiele	293
Grundlagen	294
Aktionen	294
Szenarien	361
Aktualisieren der zugehörigen Entität einer Opportunity	361
Versionshinweise	367
Dokumentverlauf	400

Die AWS Partner Central API-Referenz wurde neu strukturiert. Weitere Informationen zu den unterstützten API-Vorgängen finden Sie in der [AWS Partner Central API-Referenz](#).

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.

Willkommen auf der AWS Entwicklerhandbuch für Partner Central

In diesem Entwicklerhandbuch wird beschrieben, wie AWS Partner Service-APIs verwenden können, um Bau-, Marketing-, Verkaufs- und Wachstumsaktivitäten zu integrieren und zu verwalten AWS.

Themen

- [Verwenden AWS Partner Central-API mit einem AWS SDK](#)
- [Einrichtung und Authentifizierung](#)

Verwenden AWS Partner Central-API mit einem AWS SDK

AWS Software Development Kits (SDKs) sind für viele beliebte Programmiersprachen verfügbar. Jedes SDK bietet eine API, Codebeispiele und Dokumentation, die es Entwicklern erleichtern, Anwendungen in ihrer bevorzugten Sprache zu erstellen.

SDK-Dokumentation	Codebeispiele
AWS SDK für C++	AWS SDK für C++ Code-Beispiele
AWS CLI	AWS CLI Code-Beispiele
AWS SDK für Go	AWS SDK für Go Code-Beispiele
AWS SDK für Java	AWS SDK für Java Code-Beispiele
AWS SDK für JavaScript	AWS SDK für JavaScript Code-Beispiele
AWS SDK für Kotlin	AWS SDK für Kotlin Code-Beispiele
AWS SDK für .NET	AWS SDK für .NET Code-Beispiele
AWS SDK für PHP	AWS SDK für PHP Code-Beispiele
AWS -Tools für PowerShell	AWS -Tools für PowerShell Code-Beispiele
AWS SDK für Python (Boto3)	AWS SDK für Python (Boto3) Code-Beispiele

SDK-Dokumentation	Codebeispiele
AWS SDK für Ruby	AWS SDK für Ruby Code-Beispiele
AWS SDK für Rust	AWS SDK für Rust Code-Beispiele
AWS SDK für SAP ABAP	AWS SDK für SAP ABAP Code-Beispiele
AWS SDK für Swift	AWS SDK für Swift Code-Beispiele

Beispiel für die Verfügbarkeit

Sie können nicht finden, was Sie brauchen? Fordern Sie ein Codebeispiel an, indem Sie unten den Link Provide feedback (Feedback geben) auswählen.

Einrichtung und Authentifizierung

Die Einrichtung und Authentifizierung mit der AWS Partner Central-API umfasst drei Schritte. Hier ist ein Überblick über den Prozess:

1. Verknüpfen Sie Ihr AWS Marketplace Verkäuferkonto mit Partner Central.
2. Richten Sie Berechtigungen mithilfe von IAM ein.
3. Authentifizieren Sie Ihre API-Aufrufe mit Signature Version 4 (Sigv4).

Verknüpfen Sie Ihre AWS-Konto mit Partner Central

Die Verknüpfung mit Partner AWS-Konto Central ist eine Voraussetzung für die Nutzung der API. Weitere Informationen finden Sie unter [AWS Partner Central-Konten mit AWS Marketplace Verkäuferkonten verknüpfen](#). Sie müssen sich bei Partner Central mit einem Konto anmelden, das über Alliance-Lead- oder Cloud-Administratorrechte verfügt, zu dem **Account Linking** Abschnitt navigieren und den Anweisungen folgen.

Einrichten von IAM

Um die AWS Partner Central-API verwenden zu können, benötigen Sie eine AWS Identity and Access Management(IAM-) Rolle oder einen IAM-Benutzer, um Anrufe tätigen zu können. Weitere

Informationen finden Sie unter [Wann verwende ich IAM?](#) . Folgen Sie dazu den Schritten zum [Erstellen von IAM-Rollen](#) und [Erstellen eines IAM-Benutzers](#) in Ihren AWS-Konto Anleitungen. Sie müssen dieses IAM Role/User in Ihrem Central-linked AWS Marketplace Partnerverkäuferkonto erstellen. Für die role/user IAM-Erstellung fallen keine Kosten an.

1. Erstellen Sie ein IAM Role/User

Melden Sie sich bei der AWS-Managementkonsole, navigieren Sie zum IAM-Dienst und folgen Sie den Schritten, um eine IAM-Rolle oder einen IAM-Benutzer zu erstellen.

2. Richtlinien zuweisen:

Hängen Sie nach Bedarf verwaltete Richtlinien an oder erstellen Sie benutzerdefinierte Richtlinien. Um Berechtigungen zu ändern oder zu erweitern, wenden Sie zusätzliche Richtlinien auf die IAM-Rolle an, anstatt den Inhalt zu kopieren und `AWSPartnerCentralOpportunityManagement` mit anderen Berechtigungen zu kombinieren. Vermeiden Sie es, verwaltete Richtlinien zu duplizieren, da Sie dadurch nicht automatisch Zugriff auf neue Funktionen erhalten, sobald diese veröffentlicht werden, und Sie Ihre Richtlinien in future manuell aktualisieren müssen. Weitere Informationen zu Zugriffsrichtlinien finden Sie in der [Dokumentation zur Zugriffskontrolle](#).

Verwalten AWS Marketplace Angebote

Um AWS Marketplace Angebote zu verwalten und sie mit Opportunities zu verknüpfen, müssen Partner der IAM-Rolle die Erlaubnis erteilen, auf Katalog-APIs zuzugreifen. Stellen Sie sicher, dass der role/user über Berechtigungen wie `aws-marketplace:ListEntities` und `aws-marketplace:SearchAgreements` verfügt.

Authentifizierung von API-Aufrufen

AWS Die Partner Central API verwendet Signature Version 4 (Sigv4) für die Authentifizierung. So implementieren Sie es:

Verwendung der AWS SDK

AWS SDKs kümmern sich automatisch um das Signieren von Anfragen. Geben Sie Ihre AWS Anmeldeinformationen ein und das SDK erledigt den Rest.

1. Informationen zu Java finden Sie unter [Bereitstellen temporärer Anmeldeinformationen für das AWS SDK for Java](#).

2. [Für Python \(Boto3\) siehe Credentials.](#)
3. Informationen zu JavaScript (Node.js) finden Sie unter [Anmeldeinformationen einrichten](#) in Node.js
4. Für .NET siehe [Auflösung von Anmeldeinformationen und Profilen.](#)
5. Weitere Programmiersprachen und weitere Beispiele finden Sie unter [Tools to Build On AWS.](#)

Authentifizierung ohne Verwendung von AWS SDK

Wenn für die von Ihnen gewählte Programmiersprache kein AWS SDK verfügbar ist, umfasst die Authentifizierung das manuelle Erstellen einer kanonischen Anfrage, das Signieren der Anfrage und das Verarbeiten der Sitzungstoken. AWS bietet umfassende Anleitungen zur [Verwendung der SigV4-Signatur](#). Bitte beachten Sie jedoch, dass die Verwendung des AWS SDK empfohlen wird, da das manuelle Signieren von Anfragen die Komplexität erhöht und eine sorgfältige Verwaltung der Sicherheitstoken erfordert.

Jede Anfrage für das awsjson1_0-Protokoll MUSS mithilfe der HTTP- „POST“ -Methode unter Verwendung der folgenden Header an die Root-URL (/) gesendet werden.

Erforderliche Header

Jede API-Anfrage für das awsjson1_0-Protokoll MUSS mithilfe der HTTP- „POST“ -Methode unter Verwendung der folgenden Header an die Root-URL (/) gesendet werden.

Header	Erforderlich	Description
Content-Type	true	Dieser Header hat einen statischen Wert von. <code>application/x-amz-json-1.0</code>
Content-Length	true	Der durch RFC 9110 #section -8.6 definierte Content-Length Standard-Header.
X-Amz-Target	gilt für Anfragen	Der Wert für den Betrieb <code>CreatePartner</code> des Dienstes <code>PartnerCentralAccount</code> ist beispielsweise <code>PartnerCentralAccount.CreatePartner</code> .

Signieren Sie Ihre Anrufe mit einem benutzerdefinierten Benutzeragenten

Wenn Sie API-Anfragen an AWS Partner Central stellen, empfehlen wir, den `X-Amzn-User-Agent` Header anzugeben, um die Quelle der Client-Anwendung zu AWS identifizieren, die Nutzung zu überwachen und die Leistung zu überprüfen. AWS verwendet diesen Header, um den Typ der Client-Anwendung zu unterscheiden, die den Anruf tätigt, und um Erkenntnisse über die Erfolgsquote verschiedener Client-Implementierungen zu gewinnen.

Benutzerdefinierter User-Agent-Header

Name der Kopfzeile: `X-Amzn-User-Agent`

Zweck: Unterscheidet den Typ des Clients, der die API-Anfrage stellt, und kategorisiert die Quelle der Interaktion.

Format: `{Integrator}|{Product}`

Beispielwert: `AWS|AWS Partner CRM Connector`

Wenn Sie diesen Header in jede Anfrage aufnehmen AWS, können Sie Anforderungsmuster analysieren, Integrationen verfolgen und das API-Erlebnis für verschiedene Clientsysteme verbessern.

Verwendung benutzerdefinierter Header in SDKs

Um den `X-Amzn-User-Agent` Header in SDK-Aufrufe einzubeziehen, können Sie das Verhalten der Client-Anfrage ändern, bevor Sie den API-Aufruf ausführen. Im Folgenden finden Sie ein Beispiel für eine Verkaufs-API mit dem AWS SDK for 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,
    region='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
```

Dieses Beispiel zeigt, wie ein Ereignis im Boto3-SDK registriert wird, um den X-Amzn-User-Agent Header automatisch an jede API-Anfrage anzuhängen. Derselbe Ansatz kann auf andere AWS SDKs angewendet werden, indem ihre jeweiligen Mechanismen zum Abfangen von Anfragen geändert werden.

Note

Das X-Amzn-User-Agent Header-Format wurde vereinfacht. Wir empfehlen, das neue Format zu verwenden. Das vorherige Format (CompanyName | ProductName | CRMName | ProductVersion) wird aus Gründen der Abwärtskompatibilität weiterhin unterstützt. Bestehende Integrationen funktionieren weiterhin ohne Änderungen.

Partner Central-Agenten MCP Server

Der Partner Central Agents MCP Server stellt Partner Central-Tools über das Model Context Protocol (MCP) bereit, sodass Ihre KI-Agenten und Tools Opportunity-Management, Kundeninformationen und Finanzierungsprogramme in natürlicher Sprache entdecken und mit ihnen interagieren können.

Der Server kümmert sich um die Authentifizierung, die Sitzungsverwaltung und die Genehmigung von Schreibvorgängen durch einen Mitarbeiter, wobei er die Kontrolle behält und gleichzeitig die Arbeitsabläufe optimiert.

-Übersicht

Der Partner Central Agents MCP Server ist ein vollständig verwalteter, AWS gehosteter Dienst, der MCP-compatible KI-Clients mit den Partner Central-Agenten verbindet. Er verwendet JSON-RPC 2.0 über HTTPS mit SigV4-Authentifizierung und unterstützt Server-Sent Events (SSE) für Streaming-Antworten in Echtzeit.

Der Server unterstützt Multi-Turn-Konversationen, Dateianhänge für die Dokumentenanalyse und einen integrierten Genehmigungsworkflow, der Ihre ausdrückliche Zustimmung erfordert, bevor ein Schreibvorgang ausgeführt werden kann.

Funktionen der Agenten

- Einblicke in die Pipeline — Erhalten Sie Konversationsinformationen über Ihre Vertriebspipeline, einschließlich risikobehafteter Geschäftschancen, Phasenentwicklung und abgeschlossener Verlustanalysen
- Schaffung von Geschäftschancen — Schaffen Sie neue Geschäftschancen durch Konversation in natürlicher Sprache, durch das Hochladen von Besprechungsnotizen, Vorschlägen oder Anrufprotokollen oder durch das Klonen einer bestehenden Verkaufschance
- Zusammenfassung von Geschäftschancen — Generieren Sie auf einen Blick präzise Zusammenfassungen aller Geschäfte, die sich auf die Phase, die Ausgaben, das Abschlussdatum und mehr beziehen
- Generierung von Vertriebsaktivitäten — Entwickeln Sie maßgeschneiderte Vertriebsstrategien, die Einzelheiten zu Geschäftschancen, Branchenkontext und Lösungsempfehlungen kombinieren AWS
- Erstellung von Kundenprofilen — Generieren Sie Unternehmensprofile anhand öffentlich verfügbarer Informationen zu Branche, Geschäftsmodell, Region und aktuellen Entwicklungen

- Lösungsempfehlung — Cross-reference Ihre registrierten Lösungen im Vergleich zu den Chancenanforderungen, um die beste Lösung zu finden
- Finanzierungsempfehlung — Evaluieren Sie Möglichkeiten anhand verfügbarer AWS Finanzierungsprogramme, schätzen Sie die Beträge ab und erstellen Sie Finanzierungsanträge
- Empfehlungen für den nächsten Schritt — Erstelle priorisierte Maßnahmenpläne, die auf AWS Ko-Selling-Standards und Leitlinien zur schrittweisen Weiterentwicklung basieren
- Weiterentwicklung von Geschäftschancen — Laden Sie unterstützende Dokumente hoch, extrahieren Sie relevante Daten und führen Sie Ihre Geschäftschancen durch die einzelnen Phasen der Pipeline
- AI-assisted Produktliste — Generieren Sie hochwertige AWS Marketplace Produktangebote aus Ihren vorhandenen digitalen Ressourcen, bewerten Sie die Qualität Ihrer Angebote anhand von AWS Marketplace Standards und erhalten Sie Empfehlungen auf Feldebene, um die Auffindbarkeit zu verbessern. Weitere Informationen finden Sie unter [AI-assisted Produktangebot im Verkäuferleitfaden](#).AWS Marketplace

Wichtigste Vorteile

- Konversationszugriff auf Partner Central — Stellen Sie Fragen in natürlicher Sprache, anstatt sich in komplexen Konsolen-Workflows zurechtzufinden
- Mehr Zeit für den Verkauf — Schaffen Sie Geschäftschancen durch ein kurzes Gespräch, anstatt ein mehrstufiges Formular auszufüllen. Dadurch wird die Dateneingabe reduziert und die Vertriebsteams der Partner können mehr Zeit mit dem Verkauf verbringen. Der Mitarbeiter empfiehlt Verbesserungen, sodass Partner qualitativ hochwertigere Verkaufschancen einreichen und die Hygiene in der Pipeline verbessern
- Human-in-the-loop Sicherheit — Alle Schreibvorgänge erfordern vor der Ausführung Ihre ausdrückliche Zustimmung
- Multi-turn Konversationen — Verfeinern Sie Ihre Abfragen innerhalb einer Sitzung, ohne den Kontext zu wiederholen
- Dateianalyse — Hängen Sie Dokumente (PDF, DOCX, XLSX, CSV, Bilder) an, die der Agent zusammen mit Ihren Fragen analysieren kann
- Zentralisiertes Zugriffsmanagement — Steuern Sie den Zugriff mithilfe von IAM-Richtlinien mit fein abgestuften Berechtigungen
- Sandbox-Tests — Testen Sie Workflows in einer isolierten Sandbox-Umgebung, bevor Sie mit Produktionsdaten in Berührung kommen

- Antworten streamen — Holen Sie sich Feedback über SSE, während der Mitarbeiter Ihre Anfrage bearbeitet

Verwendungsbeispiele

Alle AI-generated Erkenntnisse enthalten eine Sitzungs-ID zur Rückverfolgbarkeit. Die Daten werden auf die eigenen Möglichkeiten des angemeldeten Partners beschränkt. Alle Inhalte sind eindeutig gekennzeichnet und unterliegen der Richtlinie [AWS für verantwortungsvolle KI](#).

Schaffung von Chancen

Der Agent erstellt eine neue Opportunity anhand einer Beschreibung in natürlicher Sprache oder eines hochgeladenen Dokuments (PDF, DOCX, XLSX, TXT). Es extrahiert Kundenname, Projektumfang, voraussichtliches Abschlussdatum und andere Pflichtfelder, reichert Kundendetails aus öffentlich zugänglichen Webdaten an, validiert den Entwurf anhand der AWS-Bereitschaftsanforderungen und erstellt die Verkaufschance nach Ihrer Genehmigung über die Partner Central Selling API.

- „Schaffen Sie eine Chance für Acme Corp — sie wollen ihr Data Warehouse auf Redshift migrieren, als Abschlussdatum für Ende des dritten Quartals anvisieren, erwartete monatliche AWS-Ausgaben in Höhe von 40.000\$“
- „Hier sind meine Gesprächsnotizen vom gestrigen Treffen mit GlobalTech — erstellen Sie eine Geschäftschance aus dieser Niederschrift“
- „Nutzen Sie dieses Angebots-PDF, um eine Gelegenheit für die SAP-Migration des Kunden zu schaffen“

Klonen von Geschäftschancen

Der Mitarbeiter erstellt aus einer bestehenden Opportunity eine neue Opportunity, fügt die von Ihnen angegebenen neuen Kunden- oder Projektdetails hinzu und überprüft vor der Einreichung, ob mindestens ein Unterscheidungsfeld geändert wurde, sodass Duplikate nicht von der AWS-Überprüfung zurückgewiesen werden.

- „Verkaufschance O1234567890 für einen neuen Kunden klonen — Globex, gleicher Workload, angestrebter Abschlusstermin Ende des vierten Quartals“
- „Schaffen Sie eine Geschäftschance wie O9876543210, aber für die Produktionsumgebung des Kunden statt für die Sandbox“

Einblicke in die Pipeline

Holen Sie sich Konversationsinformationen über Ihre Vertriebspipeline. Der Mitarbeiter analysiert den Phasenverlauf, Termine, ins Stocken geratene Geschäfte und Muster abgeschlossener Geschäfte, um herauszufinden, was am wichtigsten ist.

- „Welche Gelegenheiten erfordern diese Woche meine Aufmerksamkeit?“
- „Wie viele Möglichkeiten werden nächsten Monat geschlossen?“
- „Was sind die Hauptgründe, warum wir in den letzten 6 Monaten Chancen verpasst haben?“

Zusammenfassung der Geschäftschancen

Der Mitarbeiter fasst den Unternehmensnamen, die Branche, die Phase, die voraussichtlichen monatlichen AWS Ausgaben, das geplante Abschlussdatum und andere wichtige Details zu einer übersichtlichen Zusammenfassung zusammen, ohne dass einzelne Formularfelder gescannt werden müssen.

- „Geben Sie mir eine Zusammenfassung der Geschäftschance O1234567890“
- „Was sind der aktuelle Status und die wichtigsten Details des Acme Corp-Deals?“

Verkäufe spielen bei der Generierung

Der Mitarbeiter entwickelt eine maßgeschneiderte Vertriebsstrategie, indem er die Einzelheiten der Verkaufschance, den Branchenkontext des Kunden und relevante AWS Lösungsempfehlungen zu einem sofort einsatzbereiten Verkaufsspiel kombiniert.

- „Generieren Sie ein Verkaufsspiel für die Opportunity O1234567890“
- „Was ist der beste Ansatz, um diesem Finanzdienstleistungskunden die Cloud-Migration zu verkaufen?“
- „Entwickeln Sie mir eine Vertriebsstrategie für den GlobalTech Datenanalyse-Deal“

Erstellung eines Kundenprofils

Der Kundenberater erstellt anhand öffentlich verfügbarer Informationen ein Unternehmensprofil, das sich auf die Branchenklassifizierung, das Geschäftsmodell (B2B/B2C/hybrid), die geografische Präsenz, die Unternehmensgröße, den Marktfokus und aktuelle Geschäftsentwicklungen

bezieht. Profile tragen die Bezeichnung „Generiert mit öffentlich verfügbaren Daten und AWS KI-Erkenntnissen“.

- „Erstellen Sie ein Kundenprofil für Acme Corp“
- „Was wissen wir über die Branche und das Geschäftsmodell dieses Kunden?“
- „Stellen Sie eine Unternehmensübersicht für mein bevorstehendes Treffen zusammen mit GlobalTech“

Empfehlung für eine Lösung

Der Mitarbeiter vergleicht Ihre registrierten Lösungen mit den Opportunity-Anforderungen und zeigt den Lösungsnamen und die Beschreibung an und gibt an, ob sie bereits mit der Opportunity verknüpft ist.

- „Welche unserer Lösungen passt am besten zur Geschäftschance O1234567890?“
- „Ist unsere Datenanalyzelösung bereits Teil dieses Deals?“
- „Empfehlen Sie Lösungen für Kunden, die ihre SAP-Workloads migrieren möchten“

Empfehlung für eine Finanzierung

Der Makler bewertet jede Gelegenheit anhand der verfügbaren AWS Finanzierungsprogramme auf der Grundlage der Einzelheiten der Möglichkeiten und der Eignungskriterien für das Programm. Wenn eine Übereinstimmung gefunden wird, werden der Programmname, eine Beschreibung und eine ausführliche Begründung angezeigt. Anschließend können Sie Finanzierungsbeträge schätzen, automatisch ausgefüllte Finanzierungsanforderungsentwürfe erstellen oder im Gespräch mehr über Programme erfahren. Die Verfügbarkeit des SCA-Budgets (Strategic Collaboration Agreement) wird gegebenenfalls angezeigt, und bei allen Aktionen werden die IAM-Genehmigungen eingehalten.

- „Habe ich Anspruch auf Förderprogramme im Rahmen der Opportunity O6789012345?“
- „Schätzen Sie den Finanzierungsbetrag für ein PoC mit diesem Kunden“
- „Erstellen Sie einen Antrag auf MAP-Leistung für diese Gelegenheit“

Empfehlungen für den nächsten Schritt

Der Kundenbetreuer bewertet die Geschäftschance anhand AWS der Leitlinien für den Stufenverlauf, vergleicht die aktuellen Daten mit den Kriterien für gut qualifizierte Stellenangebote und ermittelt

genau, welche Informationen Sie noch sammeln müssen. Das Ergebnis ist ein priorisierter Aktionsplan, der auf Co-Selling-Standards basiert.AWS

- „Was muss ich als Nächstes tun, um Opportunity O1234567890 zu nutzen?“
- „Ist diese Gelegenheit bereit für die Einreichung? Welche Felder fehlen?“
- „Was sind die Voraussetzungen, um dieses Angebot von einem potenziellen Kunden auf ein qualifiziertes Angebot umzustellen?“

Weiterentwicklung der Geschäftschance

Der Mitarbeiter akzeptiert unterstützende Dokumente (Sitzungsprotokolle, Gesprächsnotizen, E-Mail-Zusammenfassungen), extrahiert relevante Informationen, ordnet sie den Opportunity-Feldern zu, bewertet die Anforderungen der Phase und aktualisiert die Opportunity. Wenn weiterhin Lücken bestehen, wird eine Aufschlüsselung der erfüllten und der nicht erfüllten Anforderungen zurückgegeben.

- „Hier sind meine Telefonnotizen — Aktualisiere die Gelegenheit O1234567890 mit den relevanten Details“
- „Ich füge das Protokoll der Besprechung bei. Nutzen Sie diese Gelegenheit auf der Grundlage dessen, was wir besprochen haben.“
- „Sehen Sie sich diese E-Mail-Zusammenfassung an und teilen Sie mir mit, welche Opportunity-Felder sie erfüllt“

Agenturerfahrung für das Onboarding von Partnern

Der Mitarbeiter automatisiert das Onboarding von Partnern in Partner Central und bietet geführte Unterstützung bei der Einrichtung von Marketplace-Verkäufern und der Einhaltung von PRM-Vorschriften — alles in natürlicher Sprache.

Funktionen des Agenten:

- Automatisierung von Partnerprofilen — Scannen Sie Ihre Website, füllen Sie Ihr Partnerprofil automatisch aus und verwalten Sie die Sichtbarkeit, den Lead-Kontakt der Allianz, Schulungsdomänen und Kontoverbindungen

- Anleitung zur Einrichtung von Verkäufern — Step-by-step Anleitung zur Marketplace-Verkäuferregistrierung, einschließlich Steuerformulare, Bankwesen, Compliance (KYC/BAV/SU), ESC-Katalog und servicebezogene Rollen
- Richtlinien zur Einhaltung von PRM-Vorschriften — Beurteilen Sie die PRM-Bereitschaft, rufen Sie die Produktcodes für die Umsatzkennzeichnung ab und überprüfen Sie die Verbindungen zwischen Tochterunternehmen, um den Umsatz konsolidiert zuordnen zu können

Automatisierung von Partnerprofilen

Der Mitarbeiter scannt Ihre Unternehmenswebsite, um Ihr Partnerprofil zu extrahieren und automatisch auszufüllen, und führt Sie dann durch alle verbleibenden Lücken. Er kann Profelfelder aktualisieren, die Sichtbarkeit ändern, den Hauptkontakt Ihrer Allianz verwalten, Zertifizierungsdomänen für Schulungen verknüpfen und Einladungen zu Kontoverbindungen bearbeiten.

- „Kannst du dir meine Website ansehen und mein Profil ausfüllen?“
- „Machen Sie unser Profil öffentlich, damit AWS-Kunden uns finden können“
- „Informiere unseren Hauptansprechpartner der Allianz unter [E-Mail] auf [Name]“
- „Connect unser EMEA-Tochterkonto“

Anleitung zur Einrichtung für Verkäufer

Der Mitarbeiter führt die Marketplace-Verkäuferregistrierung von Anfang bis Ende durch, bestimmt das richtige Steuerformular für Ihr Land und Ihren Unternehmenstyp, erklärt die Bank- und Auszahlungsmodalitäten für die einzelnen Regionen und identifiziert die für Sie geltenden Compliance-Anforderungen (KYC, Bankkontoverifizierung, Überprüfung durch sekundäre Nutzer).

- „Ich möchte anfangen, auf Marketplace zu verkaufen — wo fange ich an?“
- „Welches Steuerformular benötige ich? Ich bin ein Unternehmen in Deutschland“
- „Brauche ich KYC? Ich verkaufe an Kunden in Frankreich“
- „Wie richte ich Zahlungen ein, wenn ich mich außerhalb der USA befinde?“

Richtlinien zur Einhaltung von PRM-Vorschriften

Der Mitarbeiter überprüft, ob Ihr Konto verknüpft ist PRM-ready , und stellt sicher, dass Ihr Konto verknüpft ist, dass ein Angebot vorhanden ist und Ihr Produktcode zum Markieren verfügbar ist. Bei Partnern mit Tochterkonten wird überprüft, ob alle Verkäuferkonten für die konsolidierte Umsatzzuweisung ordnungsgemäß verknüpft sind.

- „Bin ich PRM-konform?“
- „Wo finde ich meinen Produktcode für die Kennzeichnung?“
- „Teilen Sie mir alle Schritte mit, die ich ausführen muss, um PRM-konform zu sein“

Erste Schritte

Sind Sie bereit, den MCP-Server für Partner Central Agents einzurichten? Eine schrittweise Anleitung zur Einrichtung finden Sie im Handbuch [Erste Schritte](#).

Erste Schritte mit dem Partner Central Agent MCP Server

Diese Anleitung führt Sie durch die Einrichtung des programmatischen Zugriffs auf den Partner Central Agent MCP Server mithilfe eines benutzerdefinierten MCP-Clients. Der Server verwendet direktes HTTPS mit SigV4-Authentifizierung — kein Proxy oder IDE-Plugin erforderlich.

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie über Folgendes verfügen:

- Ein aktives Partner Central-Konto (auf die AWS Konsole migriert)
- Ein AWS Konto mit IAM-Berechtigungen für Partner Central
- AWS CLI wurde mit Anmeldeinformationen installiert und konfiguriert
- Zugang zur Region US-East-1 (Nord-Virginia)
- HTTPS-Konnektivität zu `partnercentral-agents-mcp.us-east-1.api.aws`
- Unterstützung für TLS 1.2 und höher in Ihrem HTTP-Client
- Ein MCP-compatible Client, der JSON-RPC 2.0- und SigV4-Anforderungssignierung unterstützt

Schritt 1: Richten Sie IAM-Berechtigungen ein

Der Partner Central Agent MCP Server benötigt IAM-Berechtigungen auf zwei Ebenen: Protokollzugriff (zur Kommunikation mit dem MCP-Endpunkt) und Datenzugriff (zur Ausführung von Partner Central-Vorgängen).

IAM-Richtlinien anhängen

So hängen Sie mithilfe der Management Console eine Richtlinie an Ihre IAM-Identität an AWS:

1. Öffnen Sie die [IAM-Konsole](#).
2. Wählen Sie im linken Navigationsbereich Benutzer, Benutzergruppen oder Rollen aus, je nachdem, welcher Identität Sie die Richtlinie zuordnen möchten, und wählen Sie dann den Namen des jeweiligen Benutzers, der Gruppe oder der Rolle aus.
3. Wählen Sie die Registerkarte Berechtigungen.
4. Wählen Sie Richtlinien anhängen (oder Berechtigungen hinzufügen, wenn es das erste Mal ist).
5. Suchen Sie in der Richtlinienliste nach der verwalteten Richtlinie, die Sie anhängen möchten, und wählen Sie sie aus (z. B. eine benutzerdefinierte Richtlinie, die Sie aus den folgenden JSON-Blöcken erstellt haben).
6. Wählen Sie zur Bestätigung Richtlinien anhängen (oder Weiter und dann Berechtigungen hinzufügen).

Die Berechtigungen werden sofort wirksam. Sie können mehrere Richtlinien an dieselbe Identität anhängen.

Empfohlen: Verwenden Sie die verwaltete Richtlinie

Die einfachste Methode, Zugriff auf das MCP-Protokoll zu gewähren, besteht darin, die `AWSMcpServiceActionsFullAccess` verwaltete Richtlinie an Ihre IAM-Identität anzuhängen. Diese Richtlinie umfasst alle Berechtigungen, die für die Interaktion mit dem MCP-Server erforderlich sind.

Für eine detaillierte Steuerung können Sie den `aws:IsMcpServiceAction` Bedingungsschlüssel in Ihren IAM-Richtlinien verwenden, um Berechtigungen speziell für MCP-Serviceaktionen einzuschränken.

Mindestberechtigungen für den Zugriff auf das MCP-Protokoll

Ihre IAM-Identität benötigt mindestens diese Aktion, um mit dem MCP-Server zu interagieren:

Action	Description
<code>partnercentral:UseSession</code>	Erforderlich, um Konversationssitzungen zu erstellen, zu aktualisieren und abzurufen

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "partnercentral:UseSession"
      ],
      "Resource": "*",
      "Condition": {
        "Bool": {
          "aws:IsMcpServiceAction": "true"
        }
      }
    }
  ]
}
```

Berechtigungen für den Datenzugriff

Um tatsächlich Partner Central-Operationen über den Agenten ausführen zu können, benötigen Sie je nach Anwendungsfall zusätzliche Berechtigungen.

Verwaltung von Geschäftschancen:

```
{
  "Effect": "Allow",
  "Action": [
    "partnercentral:List*",
    "partnercentral:Get*",
```

```
    "partnercentral:CreateOpportunity",
    "partnercentral:UpdateOpportunity",
    "partnercentral:SubmitOpportunity",
    "partnercentral:AssignOpportunity",
    "partnercentral:AssociateOpportunity",
    "partnercentral:DisassociateOpportunity"
  ],
  "Resource": "*"
}
```

Förderprogramme:

```
{
  "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": "*"
}
```

Zugang zum Marketplace:

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

Onboarding in 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": "*"
}
```

Fügen Sie für das Marketplace-Verkäufer-Setup außerdem Folgendes hinzu:

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

Für die dienstbezogene Rollenverwaltung (Wiederverkauf Authorizations/CPPO):

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

Richtlinie für vollen Zugriff

Für Entwicklungs- und Testzwecke können Sie alle Berechtigungen in einer einzigen Richtlinie zusammenfassen:

```
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 Richtlinie

Beschränken Sie in Produktionsumgebungen oder in Fällen, in denen nur Lesen verwendet wird, die Berechtigungen auf Lesevorgänge:

```
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": "*"   
      }  
    ]  
  }'
```

```
}  
]  
'
```

Schritt 2: Connect Ihren MCP-Client

Der Partner Central Agent MCP Server verwendet direktes HTTPS mit Sigv4-Anforderungssignierung. Es gibt keine Proxyschicht — Ihr MCP-Client sendet JSON-RPC 2.0-Anfragen direkt an den Endpunkt.

Endpoint

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

Authentifizierung

Alle Anfragen müssen mit [AWS Signature Version 4](#) signiert werden, wobei Folgendes verwendet wird:

- Name des Dienstes: partnercentral-agents-mcp
- Region: us-east-1

Initialisieren Sie die MCP-Verbindung

Senden Sie eine `initialize` Anfrage zur Einrichtung des Protokolls:

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

Erwartete Antwort:

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

Verfügbare Tools auflisten

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

Signieren Sie Ihre Anrufe mit dem MCP-Header

Wenn Sie Anfragen an den MCP der Partner Central-Agenten richten, empfehlen wir, den benutzerdefinierten MCP-Header mithilfe der folgenden Methoden einzubeziehen, um die Quelle der Client-Anwendung zu AWS identifizieren, die Nutzung zu überwachen und die Leistung zu überprüfen. AWS verwendet diesen Header, um den Typ der Client-Anwendung zu unterscheiden, die den Anruf tätigt, und um Erkenntnisse über die Erfolgsquote verschiedener Client-Implementierungen zu gewinnen.

Methode 1: `_meta`-Feld (MCP) programmatic/stateless

Geben Sie für Code, der `tools/call` MCP-Anfragen direkt erstellt, das Feld für Anfragen an `_meta`.

```
{
  "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>"
    }
  }
}
```

Methode 2: ClientInfo (sitzungsbasierte benutzerdefinierte Agenten)

Geben Sie für benutzerdefinierte MCP-Clients, die Sitzungen einrichten, MCP-Header-Informationen in das Feld ein: `clientInfo`

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

Felder in: `clientInfo`

- `integrator`— Firmenname oder „Direkt“ für Partner

Beispiel: AWS

- `sourceProduct`— Product/agent Name

Beispiel: AWS CRM Connector

Methode 3: URL-Parameter (nur gehostetes MCP)

Nur für gehostete MCP-Clients, bei denen der Integrator keine Protokollfelder ändern kann.

Verwenden Sie den URL-Parameter:

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

Schritt 3: Überprüfen Sie Ihr Setup

Senden Sie eine einfache Nachricht, um zu bestätigen, dass alles funktioniert. Verwenden Sie den Sandbox Katalog zum Testen:

```
{
  "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"
    }
  }
}
```

Wenn Sie eine Antwort mit `"status": "complete"` und eine Textantwort vom Agenten erhalten, funktioniert Ihr Setup ordnungsgemäß. Die Antwort wird auch eine `enthaltensessionId` enthalten, die Sie für Follow-up-Nachrichten verwenden können.

Schritt 4: Führen Sie Ihre ersten Aufgaben aus

Fragen Sie nach Ihren Möglichkeiten

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

Prüfen Sie die Förderfähigkeit

```
{
  "jsonrpc": "2.0",
  "id": 5,
  "method": "tools/call",
  "params": {
    "name": "sendMessage",
    "arguments": {
      "sessionId": "session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx",
      "content": [
        {
          "type": "text",
          "text": "Am I eligible for MAP funding for opportunity
01234567890?"
        }
      ],
      "catalog": "AWS"
    }
  }
}
```

```

    }
  }
}

```

Sitzungsverlauf abrufen

```

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

```

Onboarding von Partnern

```

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

```

Weitere Onboarding-Aufgaben, die Sie ausprobieren sollten:

- „Führe mich durch den Prozess des Steuerfragebogens“
- „Können Sie sich meine Website ansehen und mein Partnerprofil ausfüllen?“
- „Was muss ich noch tun, um bereit zu sein, auf Marketplace zu verkaufen?“

Sicherheitsüberlegungen

- Übergeben Sie AWS Anmeldeinformationen nicht über die MCP-Tool-Parameter. Die Authentifizierung erfolgt durch Sigv4-Anforderungssignierung auf der Transportschicht.
- Verwenden Sie den Sandbox-Katalog zum Testen und Entwickeln. Der "Sandbox" Katalog bietet eine isolierte Umgebung, die sich nicht auf die Daten der Produktionspartner auswirkt.
- Wenden Sie in der Produktion IAM-Richtlinien mit den geringsten Rechten an. Verwenden Sie die schreibgeschützte Richtlinie für die Überwachung und Berichterstattung von Anwendungsfällen. Erteilen Sie dem Benutzer nur Schreibberechtigungen, wenn er Opportunities aktualisieren oder Finanzierungsanträge einreichen muss.
- Überprüfen Sie die Schreibvorgänge sorgfältig. Der Server verwendet für alle Schreibvorgänge die Genehmigung durch einen Mitarbeiter. Wenn eine Schreibaktion vorgeschlagen wird, überprüfen Sie die Parameter, bevor Sie sie genehmigen.
- Sitzungsdaten sind vorübergehend. Sitzungen laufen 48 Stunden nach ihrer Erstellung ab. Verlassen Sie sich bei der langfristigen Datenspeicherung nicht auf Sitzungen.
- Datei-Uploads werden in einen kurzlebigen S3-Bucket übertragen. Hochgeladene Dateien werden temporär gespeichert und nicht dauerhaft aufbewahrt. Laden Sie keine Dateien hoch, die Anmeldeinformationen, Geheimnisse oder andere vertrauliche Informationen enthalten.

Nächste Schritte

- [Konfigurationsreferenz](#) — Vollständige Referenz für Endgeräte, IAM-Aktionen, Sitzungsverwaltung und Fehlercodes
- [Tool-Referenz](#) — Detaillierte Dokumentation für sendMessage und getSession Tools

Konfigurationsreferenz

Diese Seite enthält die vollständige Referenz für die Verbindung mit dem Partner Central Agent MCP Server und dessen Konfiguration.

Endpoint

Eigenschaft	Wert
URL	<code>https://partnercentral-agents-mcp.us-east-1.api.aws/mcp</code>
Region	<code>us-east-1</code> Nur (Nord-Virginia)
Protocol (Protokoll)	JSON-RPC 2.0 über HTTPS
Streaming	Server-Sent Ereignisse (SSE)
Authentifizierung	AWS Signatur, Version 4
SigV4-Dienstname	<code>partnercentral-agents-mcp</code>
TLS-Anforderung	TLS 1.2 oder höher

IAM-Berechtigungen

Alle IAM-Aktionen verwenden das `partnercentral:` Präfix, sofern nicht anders angegeben.

Zugriff auf das MCP-Protokoll

Die einfachste Methode, Zugriff auf das MCP-Protokoll zu gewähren, besteht darin, die `AWSMcpServiceActionsFullAccess` verwaltete Richtlinie an Ihre IAM-Identität anzuhängen. Diese Richtlinie umfasst alle Berechtigungen, die für die Interaktion mit dem MCP-Server erforderlich sind. Für eine detaillierte Steuerung können Sie den `aws:IsMcpServiceAction` Bedingungsschlüssel in Ihren IAM-Richtlinien verwenden, um Berechtigungen speziell für MCP-Serviceaktionen einzuschränken.

Action	Description
<code>partnercentral:UseSession</code>	Konversationssitzungen erstellen, aktualisieren und abrufen

Verwaltung von Geschäftschancen

Action	Description
<code>partnercentral:List*</code>	Nennen Sie Möglichkeiten, Lösungen und andere Ressourcen von Partner Central
<code>partnercentral:Get*</code>	Rufen Sie Details zu einzelnen Geschäftschancen und anderen Ressourcen ab
<code>partnercentral:CreateOpportunity</code>	Schaffen Sie eine Geschäftschance
<code>partnercentral:UpdateOpportunity</code>	Ändern Sie die Opportunity-Felder (Phase, Abschlussdatum, Umsatz usw.)
<code>partnercentral:SubmitOpportunity</code>	Reichen Sie eine Opportunity zur AWSÜberprüfung ein
<code>partnercentral:AssignOpportunity</code>	Weisen Sie einem Vertreter eines Partners eine Verkaufschance zu
<code>partnercentral:AssociateOpportunity</code>	Verknüpfen Sie eine Opportunity mit einer anderen Ressource (Lösung usw.)
<code>partnercentral:DisassociateOpportunity</code>	Entfernen Sie eine Verknüpfung zwischen einer Opportunity und einer anderen Ressource
<code>partnercentral:StartEngagementFromOpportunityTask</code>	Reichen Sie Möglichkeiten ein, um ein Engagement von einer Verkaufschance aus zu starten

Förderprogramme

Action	Description
<code>partnercentral:ListBenefitAllocations</code>	Führen Sie die verfügbaren Leistungszuweisungen für Ihr Konto auf

Action	Description
<code>partnercentral:ListBenefitApplications</code>	Führen Sie die eingereichten Leistungsanträge auf
<code>partnercentral:CreateBenefitApplication</code>	Erstellen Sie einen neuen Leistungsantrag (MAP, POC, WMP)
<code>partnercentral:GetBenefitApplication</code>	Rufen Sie Details zu einem bestimmten Leistungsantrag ab
<code>partnercentral:UpdateBenefitApplication</code>	Aktualisieren Sie einen ausstehenden Leistungsantrag
<code>partnercentral:AssociateBenefitApplicationResource</code>	Verknüpfen Sie eine Ressource mit einem Leistungsantrag
<code>partnercentral:DisassociateBenefitApplicationResource</code>	Entfernen Sie einen Ressourcenlink aus einem Leistungsantrag

Zugang zum Marketplace

Action	Description
<code>aws-marketplace:DescribeEntity</code>	Rufen Sie Details einer Marketplace-Entität ab
<code>aws-marketplace:SearchAgreements</code>	Suchen Sie nach Marktplatz-Vereinbarungen
<code>aws-marketplace:ListEntities</code>	Marktplatz-Entitäten auflisten

Onboarding-Agent

Verwaltung von Partnerkonten (**partnercentral**):

Action	Description
<code>partnercentral:ListPartners</code>	Partnerregistrierungen für das Konto auflisten

Action	Description
<code>partnercentral:GetPartner</code>	Rufen Sie die Partnerregistrierungs- und Profildetails ab
<code>partnercentral:GetProfileVisibility</code>	Rufen Sie die aktuelle Profilsichtbarkeit ab (PUBLIC/PRIVATE)
<code>partnercentral:GetAllianceLeadContact</code>	Rufen Sie die Kontaktinformationen des Allianzleiters ab
<code>partnercentral:GetProfileUpdateTask</code>	Überprüfen Sie den Status einer ausstehenden asynchronen Profilaktualisierung
<code>partnercentral:StartProfileUpdateTask</code>	Reichen Sie ein Partnerprofil-Update ein (asynchron)
<code>partnercentral:CancelProfileUpdateTask</code>	Stornieren Sie eine ausstehende Aufgabe zur Profilaktualisierung
<code>partnercentral:PutProfileVisibility</code>	Ändern Sie die Sichtbarkeit des Profils auf ÖFFENTLICH oder PRIVAT
<code>partnercentral:PutAllianceLeadContact</code>	Aktualisieren Sie den Hauptkontakt der Allianz (Name, Titel, E-Mail)
<code>partnercentral:SendEmailVerificationCode</code>	Senden Sie einen Bestätigungscode für E-Mail-basierte Änderungen domain/contact
<code>partnercentral:AssociateAwsTrainingCertificationEmailDomain</code>	Verknüpfen Sie eine E-Mail-Domain für die Nachverfolgung von Schulungszertifikaten
<code>partnercentral:DisassociateAwsTrainingCertificationEmailDomain</code>	Entfernen Sie eine verknüpfte E-Mail-Domain für Schulungszertifikate
<code>partnercentral:GetAccountConnections</code>	Listet aktive Kontoverbindungen auf (z. B. Links zu Tochterunternehmen)

Action	Description
<code>partnercentral:GetConnectionsInvitations</code>	Führt ausstehende gesendete und empfangene Verbindungseinladungen auf
<code>partnercentral:CreateConnectionInvitation</code>	Senden Sie eine Verbindungseinladung an ein anderes AWS Konto
<code>partnercentral:CancelConnectionInvitation</code>	Stornieren Sie eine ausstehende Einladung zu einer ausgehenden Verbindung
<code>partnercentral:RespondConnectionInvitation</code>	Nehmen Sie eine eingehende Verbindungseinladung an oder lehnen Sie sie ab
<code>partnercentral:CancelConnection</code>	Kündigen Sie eine aktive Kontoverbindung
<code>partnercentral:ManageConnectionPreferences</code>	Ruft die Einstellungen für die gemeinsame Nutzung zwischen verbundenen Konten ab oder aktualisiert sie

Einrichtung des Marketplace-Verkäufers (**aws-marketplace**):

Action	Description
<code>aws-marketplace:ListEntities</code>	Marketplace-Entitäten auflisten (z. B. Verkäufer entitäten)
<code>aws-marketplace:DescribeEntity</code>	Vollständige Details einer Marketplace-Entität abrufen
<code>aws-marketplace:DescribeChangeSet</code>	Überprüfen Sie den Status eines eingereichten Änderungssatzes
<code>aws-marketplace:StartChangeSet</code>	Einen Marketplace-Änderungssatz einreichen (z. B. Aktualisierung des Verkäuferprofils)

IAM — serviceverknüpfte Rolle (**iam**):

Action	Description
<code>iam:GetRole</code>	Prüfen Sie, ob die mit dem Service verknüpfte Rolle Marketplace Resale Authorization existiert
<code>iam:CreateServiceLinkedRole</code>	Erstellen Sie die Rolle <code>AWSServiceRoleForMarketplaceResaleAuthorization</code>

Katalogumgebungen

Katalog	Description
"AWS"	Produktionsumgebung. Alle Operationen wirken sich auf Live-Partnerdaten aus.
"Sandbox"	Isolierte Testumgebung. Keine Auswirkungen auf die Produktionsdaten. Für die Entwicklung und Validierung verwenden.

Verwaltung von Sitzungen

Eigenschaft	Wert
Format der Sitzungs-ID	UUID v4 mit <code>session-</code> Präfix (z. B.) <code>session-550e8400-e29b-41d4-a716-446655440000</code>
Erstellung einer Sitzung	Automatisch beim ersten <code>sendMessage</code> Aufruf ohne <code>sessionId</code>
Ablauf der Sitzung	48 Stunden nach der Erstellung der Sitzung (absoluter Ablauf, nicht aufgrund von Inaktivität)

Datei-Upload

Eigenschaft	Wert
Max. Anzahl an Dateien pro Nachricht	3
Größenbeschränkung für Bilder	3,75 MB
Begrenzung der Dokumentgröße	4,5 MB
Zulässige Erweiterungen	doc, docx, pdf, png, jpeg, xlsx, csv, txt
S3-Bucket (Produktion)	aws-partner-central-marketplace-ephemeral-writeonly-Dateien
Pfad hochladen	s3://{bucket}/{aws-account-id}/
S3-URI-Anforderung	Muss einen <code>versionId</code> Abfrageparameter enthalten

Arbeitsablauf hochladen

1. Laden Sie Ihre Datei unter Ihrem AWS Konto-ID-Präfix in den entsprechenden S3-Bucket hoch.
2. Notieren Sie sich die S3-URI einschließlich der vom Upload `versionId` zurückgegebenen.
3. Nehmen Sie die Datei als `document` Inhaltsblock in Ihren `sendMessage` Aufruf auf.

Beispiel für ein S3-URI-Format:

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

Fehlercodes

Code	Name	Description
-32001	AUTHENTICATION_FAILURE	Die Sigv4-Signatur ist ungültig oder die Anmeldeinformationen sind abgelaufen
-31004	TOOL_PERMISSION_DENIED	Die IAM-Identität verfügt nicht über die erforderliche <code>partnercentral:</code> Aktion für diesen Vorgang
-32002	ACCESS_DENIED	Allgemeiner Zugriff verweigert (Konto nicht registriert, Region stimmt nicht überein usw.)
-32004	LIMIT_EXCEEDED	Ratenlimit oder Kontingent überschritten. Versuchen Sie es erneut mit exponentiellem Backoff.
-30001	RESOURCE_NOT_FOUND	Die angeforderte Ressource (Sitzung, Gelegenheit usw.) ist nicht vorhanden
-32600	INVALID_REQUEST	Fehlerhafte JSON-RPC Anfrage oder ungültige Parameter
-32603	INTERNAL_ERROR	Server-side Fehler. Wiederholen Sie die Anforderung.

Ratenbegrenzungen

Der Server erzwingt Ratenbegrenzungen pro Konto:

Operation	Kontinuierliche Rate	Burst
sendMessage	2 Anfragen pro Minute	10
Alle anderen Vorgänge	10 Anfragen pro Minute	20

Wenn Sie diese Grenzwerte überschreiten, gibt der Server den Fehlercode `-32004` (LIMIT_EXCEEDED) zurück. Implementieren Sie exponentielles Backoff mit Jitter in Ihrem Client, um die Ratenbegrenzung problemlos zu handhaben.

Streaming-Ereignistypen (SSE)

Wenn `true` bei einem `sendMessage` Aufruf auf `stream` gesetzt ist, gibt der Server Server-Sent Ereignisse mit den folgenden Ereignistypen zurück:

Ereignistyp	Description
<code>stream_start</code>	Stream-Verbindung hergestellt
<code>assistant-response.start</code>	Der Agent hat mit der Generierung einer Antwort begonnen
<code>assistant-response.delta</code>	Inkrementeller Textteil der Antwort des Agenten
<code>assistant-response.completed</code>	Der Agent hat die Generierung der Antwort abgeschlossen
<code>server-tool-use</code>	Der Agent ruft ein internes Tool auf (Lesevorgang)
<code>server-tool-response</code>	Ergebnis eines internen Tool-Aufrufs
<code>tool_approval_request</code>	Der Agent bittet um die Genehmigung eines menschlichen Benutzers für einen Schreibvorgang
<code>stream_end</code>	Stream-Verbindung wird geschlossen

Ereignistyp	Description
done	Letztes Ereignis, das anzeigt, dass der SSE-Stream abgeschlossen ist

Nächste Schritte

- [Tool-Referenz](#) — Detaillierte Dokumentation für sendMessage und getSession Tools

Referenz zu Tools

Der Partner Central Agent MCP Server stellt zwei MCP-Tools zur Verfügung: sendMessage für alle Agenteninteraktionen und getSession für das Abrufen des Sitzungsstatus. Alle Vorgänge in Partner Central — Anfragen zu Geschäftschancen, Finanzierungsanträge, Dokumentenanalysen — werden in natürlicher Sprache über abgewickelt. sendMessage

Überblick über die Tools

Tool	Description	Kategorie
sendMessage	Senden Sie Nachrichten an den Partner Central AI-Agenten. Unterstützt Text, Dateianhänge und Genehmigungsantworten, die von einem Mitarbeiter bearbeitet werden.	Lesen/Schreiben
getSession	Rufen Sie den Sitzungssatus einschließlich Konversationsverlauf, Ereignissen und Metadaten ab.	Read-only

Nachricht senden

Primäres Tool für alle Interaktionen mit KI-Agenten in Partner Central. Verwenden Sie dieses Tool, um Fragen zu stellen, Aktionen anzufordern, Dokumente zur Analyse anzuhängen und auf Genehmigungsanfragen für Schreibvorgänge zu antworten.

Der Agent behält den Konversationskontext innerhalb einer Sitzung bei, sodass Sie Folgefragen stellen können, ohne den vorherigen Kontext wiederholen zu müssen.

Parameters

- `content(erforderlich)` — Eine Reihe von Inhaltsblöcken. Jeder Block muss ein `type` Feld enthalten, das die Blockstruktur bestimmt. Sie können mehrere Blöcke in eine einzelne Nachricht aufnehmen (z. B. Text und Dokumentanhang).

Typen von Inhaltsblöcken:

Typ	Felder	Description
<code>text</code>	<code>type(erforderlich)</code> , <code>text(erforderlich)</code>	Text der Benutzernachricht, der an den Agenten gesendet wurde
<code>document</code>	<code>type(erforderlich)</code> , <code>filename(erforderlich)</code> , <code>s3Uri(erforderlich)</code>	Dateianhang, den der Agent analysieren soll. Der <code>s3Uri</code> muss einen <code>versionId</code> Parameter enthalten.
<code>tool_approval_response</code>	<code>type(erforderlich)</code> , <code>toolUseId(erforderlich)</code> , <code>decision(erforderlich)</code> , <code>message(optional)</code>	Antwort auf eine Genehmigungsanfrage, die von einem Mitarbeiter bearbeitet wird

- `catalog(erforderlich)` — Zielumgebung für den Vorgang.

- Gültige Werte: "AWS" (Produktion), "Sandbox" (Testen)
- `sessionId(optional)` — UUID v4, die eine bestehende Sitzung identifiziert, um fortzufahren. Lassen Sie es aus, um eine neue Sitzung zu erstellen. Format:`session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx`.

Standard: Eine neue Sitzung wird automatisch erstellt.

- `stream(optional)` — Aktivieren Sie das Streaming von Server-Sent Ereignissen (SSE) für die Bereitstellung von Antworten in Echtzeit.

Zulässige Werte: `true`, `false`

Standard: `false`

Antwort

Die Antwort enthält:

Feld	Description
<code>sessionId</code>	Sitzungs-ID für Folgemeldungen
<code>status</code>	Antwortstatus: <code>"complete"</code> , <code>"requires_approval"</code> , oder <code>"error"</code>
<code>content</code>	Reihe von Antwortinhaltsblöcken vom Agenten

Beispiele

Einfache Textnachricht (neue Sitzung)

Anforderung:

```
{
  "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"
    }
  }
}
```

Antwort:

```
{
  "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. **01234567890** - Acme Corp Cloud Migration - $250,000 - Qualified stage\n2. **01234567891** - GlobalTech Data Analytics - $180,000 - Prospect stage\n..."
      }
    ],
    "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
    "status": "complete"
  }
}
```

Follow-up Nachricht (bestehende Sitzung)

Anforderung:

```
{
  "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 01234567890. Is it ready for
submission?"
    },
    ],
    "catalog": "AWS"
}
}
}

```

Dateianhang

Laden Sie zuerst ein Dokument auf S3 hoch und verweisen Sie dann in der Nachricht darauf:

```

{
  "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"
    }
  }
}

```

Einschränkungen beim Hochladen von Dateien:

- Maximal 3 Dateien pro Nachricht
- Größenbeschränkung für Bilder: 3,75 MB

- Größenbeschränkung für Dokumente: 4,5 MB
- Zulässige Erweiterungen: docdocx,pdf,png,jpeg,xlsx,csv, txt
- Dateien müssen hochgeladen werden auf `s3://{bucket}/{your-aws-account-id}/`
- Die S3-URI muss den `versionId` Abfrageparameter enthalten

Human-in-the-loop Genehmigungsablauf

Wenn der Agent einen Schreibvorgang ausführen muss (z. B. eine Verkaufschance aktualisieren, einen Finanzierungsantrag einreichen), gibt er einen `"requires_approval"` Status mit den Details der vorgeschlagenen Aktion zurück. Sie müssen mit einem `tool_approval_response` Inhaltsblock antworten.

Schritt 1 — Der Agent bittet um Genehmigung:

```
{
  "jsonrpc": "2.0",
  "id": 4,
  "result": {
    "content": [
      {
        "type": "text",
        "text": "I'd like to update opportunity 01234567890 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": "01234567890",
          "targetCloseDate": "2026-03-31",
          "expectedRevenue": 300000,
          "stage": "Qualified"
        }
      }
    ],
    "sessionId": "session-550e8400-e29b-41d4-a716-446655440000",
    "status": "requires_approval"
  }
}
```

```
}
```

Schritt 2 — Genehmigen Sie die Aktion:

```
{
  "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"
    }
  }
}
```

Schritt 2 (Alternative) — Die Aktion ablehnen:

```
{
  "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"
    }
  }
}
```

```
    }  
  }  
}
```

Schritt 2 (Alternative) — Mit benutzerdefinierter Antwort überschreiben:

```
{  
  "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"  
    }  
  }  
}
```

Werte für die Genehmigungsentscheidung:

Entscheidung	Behavior
"approve"	Führen Sie das Tool mit den vorgeschlagenen Parametern aus
"reject"	Führen Sie das Tool nicht aus. Optional message erklärt warum.
"override"	Geben Sie eine benutzerdefinierte Antwort oder geänderte Anweisungen über message

Streamen mit SSE

Aktivieren Sie Streaming, um inkrementelle Antwort-Chunks zu erhalten, während der Agent Ihre Anfrage bearbeitet:

Anforderung:

```
{
  "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
    }
  }
}
```

Der Server antwortet mit einem Stream von SSE-Ereignissen:

```
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. **02345678901** - Close date is past due by 15 days\n"}

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

event: stream_end
data: {}
```

GetSession

Ruft den aktuellen Status einer Konversationssitzung ab, einschließlich des vollständigen Konversationsverlaufs, der Ereignisse und der Metadaten. Verwenden Sie diese Option, um den Sitzungsstatus zu überprüfen, vergangene Interaktionen zu überprüfen oder eine Konversation fortzusetzen.

Parameters

- `sessionId`(erforderlich) — UUID der abzurufenden Sitzung. Format: `session-xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`
- `catalog`(erforderlich) — Umgebung, zu der die Sitzung gehört.

Zulässige Werte: "AWS", "Sandbox"

Antwort

Feld	Typ	Description
<code>sessionId</code>	Zeichenfolge	Sitzungs-ID
<code>createdAt</code>	Zeichenfolge	ISO 8601-Zeitstempel der Sitzungserstellung

Feld	Typ	Description
lastActivity	Zeichenfolge	ISO 8601-Zeitstempel der letzten Aktivität
sequenceNumber	Ganzzahl	Sequenznummer des aktuellen Ereignisses
stateType	Zeichenfolge	Aktueller Sitzungsstatus
events	Array	Vollständiger Konversationsverlauf (Benutzernachrichten, Antworten der Agenten, Verwendung des Tools)
variables	object	Sitzungsvariablen und Metadaten
eventCount	Ganzzahl	Gesamtzahl der Ereignisse in der Sitzung

Beispiel

Anforderung:

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

Antwort:

```
{
  "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\":{\"}}\""}
    ]
  }
}
```

Fehlerbehandlung

Alle Fehler folgen dem JSON-RPC 2.0-Fehlerformat:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "error": {
    "code": -32001,
    "message": "Authentication failed. Verify your SigV4 credentials and ensure they have not expired."
  }
}
```

Die [the section called “Fehlercodes”](#) vollständige Liste der Fehlercodes und ihrer Bedeutung finden Sie unter.

Empfohlene Wiederholungsstrategie

- Für -32004 (LIMIT_EXCEEDED): Wiederholen Sie den Vorgang mit exponentiellem Backoff ab 1 Sekunde
- Für -32603 (INTERNAL_ERROR): Wiederholen Sie den Vorgang bis zu dreimal mit exponentiellem Backoff
- Für -32001 (AUTHENTICATION_FAILURE): Anmeldeinformationen aktualisieren und erneut versuchen

- Für alle anderen Fehler: Versuchen Sie es nicht automatisch erneut — überprüfen Sie die Fehlermeldung und korrigieren Sie die Anfrage

Über den AWS APIs von Partner Central

Die folgenden Abschnitte enthalten allgemeine Informationen zu den APIs.

Themen

- [Verwendung der AWS API für Partner Central](#)
- [Unterstützt AWS Regionen](#)
- [Berechtigungen](#)
- [Kontingente für AWS API für Partner Central](#)
- [Protokollierung für die AWS Partner Central-API](#)
- [Benachrichtigungen für AWS Partner Central-API](#)
- [Testen in einer Sandbox](#)

Verwendung der AWS API für Partner Central

Die folgenden Abschnitte enthalten Informationen zur Verwendung der AWS Partner Central-APIs.

Themen

- [Verwendung der AWS Konto-API für Partner Central](#)
- [Verwendung der AWS Vertriebs-API von Partner Central](#)
- [Verwendung der AWS API für Vorteile von Partner Central](#)
- [Verwendung der AWS Channel-API für Partner Central](#)
- [Verwenden der Revenue Measurement API](#)

Verwendung der AWS Konto-API für Partner Central

Partnerkonto verwalten

Partnerkonten werden registriert und mit einem bestehenden AWS Konto verknüpft, um ein umfassendes IAM-based Erlebnis zu bieten. Bei diesem Ansatz sind keine Anmeldeinformationen auf Benutzerebene erforderlich, sodass sichergestellt wird, dass alle Registrierungen und Vorgänge über die IAM-Entität innerhalb des AWS Kontos verwaltet werden. AWS Partner Central ermöglicht eine nahtlose Integration mit AWS Diensten. Dieses Modell bietet Partnerentitätsverwaltung, unterstützt mehrere Kataloge und richtet ein Berechtigungssystem auf AWS Kontoebene ein.

Partner können die verfügbaren Partnerkonto-APIs verwenden, um ihre Konten zu registrieren, zu verwalten und zu aktualisieren:

API für die Partnerregistrierung

Partner können ihr Konto über ihr AWS Konto für Partner Engagement registrieren. Mithilfe der Create-API geben die Partner die erforderlichen Informationen zum Allianzleiter an, akzeptieren die APN-Bedingungen und geben einen eindeutigen offiziellen Namen an.

Partnerprofil-API

Partner verwalten Profile mit Unternehmensdetails (Name, Beschreibung, Website, Logo) mit public/private Sichtbarkeitssteuerung, asynchroner Validierung und Statusverfolgung, sodass eine Aktualisierung nach der anderen möglich ist.

Domain-Verwaltungs-API

Partner können Geschäftsdomänen per E-Mail-Bestätigung registrieren und verifizieren, um die Unternehmensidentität festzustellen und die Schulungen und Zertifizierungen der Mitarbeiter sicherzustellen.

API für Partnerverbindungen

Partner verknüpfen ihre AWS Konten für verschiedene Zwecke mit anderen AWS Konten, z. B. um gemeinsam mit anderen Partnern zu verkaufen, Daten von AWS Marketplace-Angeboten zwischen Konten auszutauschen, distributors/channel Partner zum distribute/re Verkauf ihrer Produkte zu autorisieren und vieles mehr.

API zur Partnerverifizierung

Die Partnerverifizierung ist eine zwingende Voraussetzung für die Registrierung eines Partnerkontos. Bevor Partner ein Partnerkonto erstellen können, müssen sie die Prozesse zur Geschäfts- und Identitätsprüfung abschließen. Diese Überprüfungen bestätigen, dass das Unternehmen rechtmäßig registriert ist, und bestätigen die Identität der Person, die das AWS Konto registriert.

Themen

- [Mit der Partnerregistrierung arbeiten](#)
- [Mit dem Partnerprofil arbeiten](#)
- [Arbeiten mit Domain Management](#)

- [Mit Partnerkontoverbindungen arbeiten](#)
- [Mit Partner Verification arbeiten](#)

Mit der Partnerregistrierung arbeiten

Registrierung als Partner

Partner können ihr Konto über ihr AWS Konto für Partner Engagement registrieren. Mithilfe der Create-API geben die Partner die erforderlichen Informationen zum Allianzleiter an, akzeptieren die APN-Bedingungen und geben einen eindeutigen offiziellen Namen an.

Während der Registrierung

1. Synchrone Erstellung von Partnerentitäten — Kunden initiieren den Partnerregistrierungsprozess, indem sie die Create API aufrufen, die die Eingaben validiert und die Partnerentität innerhalb derselben Anfrage erstellt.
2. Anforderung von Allianz-Lead-Informationen — Bei der Partnerregistrierung muss der Kunde die Kontaktinformationen des Allianzleiters für die Kommunikation im Zusammenhang mit dem AWS Partnernetzwerk (APN) angeben.
3. Durchsetzung der Allgemeinen Geschäftsbedingungen — Kunden müssen bei der Registrierung die Allgemeinen Geschäftsbedingungen von APN akzeptieren. Die Zustimmung ist erforderlich, um die Registrierung abzuschließen und auf alle Funktionen zuzugreifen. Die Bedingungen gelten für alle Vorteile und Funktionen des APN-Programms.
4. Tag-On-Create Support — Im Rahmen von AWS Consistent Authorization Experience (CAE) und Tag-Based Authorization können Kunden bei der Erstellung der Partnerentität Tags hinzufügen.
5. Rechtliche Name-Based Überprüfung — Durch die Registrierung wird die Eindeutigkeit auf der Grundlage des offiziellen Namens des Partners durchgesetzt, sodass keine doppelten Registrierungen unter derselben Entität erfolgen.

Nach der Registrierung

1. Tag-Management-Support — Nach der Registrierung können Partner Tags für eine bestehende Partnerentität taggen, deren Markierung aufheben und sie auflisten.
2. Partnerentität abrufen — Nachdem eine Partnerentität erstellt wurde, können Kunden die List-API verwenden, um die Partner-ID abzurufen, und dann die Get API verwenden, um die Details einer bestimmten Partnerentität abzurufen.

Übersicht über API-Befehle

1. **CreatePartner API:** Kunden initiieren den Partnerregistrierungsprozess, indem sie die CreatePartner API aufrufen, die die Eingaben validiert und die Partnerentität innerhalb derselben Anfrage erstellt.
2. **ListPartners API:** Nach der Registrierung können Kunden die List-API verwenden, um eine Liste der Partnerentitäten abzurufen.
3. **GetPartner API:** Nach der Registrierung können Kunden die Get API verwenden, um Details zu einer bestimmten Partnerentität abzurufen.
4. **PutAllianceLeadContact API:** Aktualisiert die Kontaktinformationen des primären Allianzleiters. Dieser Vorgang ermöglicht es Partnern, die Bezeichnung des primären Allianzleiters zu übertragen oder die Kontaktdaten zu ändern und gleichzeitig die organisatorische Kontinuität aufrechtzuerhalten.
5. **GetAllianceLeadContact API:** Aktualisiert die Kontaktinformationen des primären Allianzleiters. Dieser Vorgang ermöglicht es Partnern, die Bezeichnung des primären Allianzleiters zu übertragen oder die Kontaktdaten zu ändern und gleichzeitig die organisatorische Kontinuität aufrechtzuerhalten.

Mit dem Partnerprofil arbeiten

Profilverwaltung

Partner verwalten Profile mit Unternehmensdetails (Name, Beschreibung, Website, Logo) mit public/private Sichtbarkeitssteuerung, asynchroner Validierung und Statusverfolgung, sodass eine Aktualisierung nach der anderen möglich ist.

1. **Profilinformationen** — Partnerprofile müssen wichtige Unternehmensdetails wie Anzeigenname, Beschreibung, Website-URL, Logo IndustrySegments und PrimarySolutionType enthalten.
2. **Mehrsprachiger Support** — Partnerprofile unterstützen mehrere Sprachen. Jede lokalisierte Version des Profils enthält ein Attribut für das Gebietsschema.
3. **Kontrolle der Profilsichtbarkeit** — Partner können die Sichtbarkeit ihres Profils (öffentlich oder privat) konfigurieren.
4. **Profilspeicher** — Profilinformationen werden im Objekt des Partnerkontos gespeichert.
5. **Profilstatusverwaltung** — Profilaktualisierungen werden vor der Veröffentlichung asynchron validiert. Partner können den Status der letzten Profilaktualisierung überprüfen (IN_PROGRESS, SUCCEED, FAILED und CANCELLED). Nur genehmigte Profile werden öffentlich sichtbar.

6. Gleichzeitigkeitskontrolle anfordern — Es ist jeweils nur eine Anfrage zur Profilaktualisierung zulässig. Wenn eine Profilaktualisierung bereits im Gange ist, müssen Kunden die laufende Aktualisierung über die `CancelPartnerProfileUpdateTask` API ausdrücklich stornieren, bevor sie eine neue starten.
7. Stornierung von Aktualisierungsaufgaben — Partner können eine laufende Profilaktualisierung stornieren, indem sie die `CancelPartnerProfileUpdateTask` API aufrufen. Eine Stornierung ist nur zulässig, solange sich das Update im Status `IN_PROGRESS` befindet, und muss abgeschlossen sein, bevor ein neues Update gestartet werden kann.

Übersicht über API-Befehle

1. `StartPutProfileTask` API: Diese API akzeptiert Aktualisierungen von Partnerprofilen und führt eine grundlegende synchrone Eingabvalidierung durch.
2. `GetProfileUpdateTask` API: Diese API ermöglicht es Partnern, den aktuellen Status der Profilaktualisierung abzurufen. Sie soll verwendet werden, nachdem der Kunde eine Profilaktualisierung über die `StartPutPartnerProfile` API gestartet hat, sodass Partner überprüfen können, ob ihre Anfrage in Bearbeitung ist, `FEHLGESCHLAGEN`, `ERFOLGREICH ABGESCHLOSSEN` oder `STORNIERT` ist.
3. `CancelProfileTask` API: Ermöglicht es einem Partner, eine Profilaktualisierungsaufgabe, die sich derzeit im Status `IN_PROGRESS` befindet, explizit abubrechen. Dies umfasst sowohl automatische Überprüfungsphasen als auch manuelle Überprüfungsphasen. Durch das Stornieren einer Anfrage kann der Partner eine neue Profilaktualisierung einleiten, ohne auf den Abschluss des aktuellen Überprüfungsprozesses warten zu müssen.
4. `PutProfileVisibility` API: Durch den Aufruf dieser API können Partner die Sichtbarkeit eines Profils unabhängig vom Gebietsschema steuern. Auf diese Weise können Partner Profile öffentlich oder privat machen, ohne den Inhalt zu ändern.
5. `GetProfileVisibility` API: Durch den Aufruf der `UpdatePartnerProfileVisibility` API können Partner die Sichtbarkeit eines Profils unabhängig vom Gebietsschema steuern. Auf diese Weise können Partner Profile öffentlich oder privat machen, ohne den Inhalt zu ändern.

Arbeiten mit Domain Management

Domänenverwaltung

Partner können Geschäftsdomänen per E-Mail-Bestätigung registrieren und verifizieren, um die Unternehmensidentität nachzuweisen und die Schulungen und Zertifizierungen der Mitarbeiter sicherzustellen.

Übersicht über API-Befehle

1. `SendEmailVerificationCode` API: Initiiert den E-Mail-Bestätigungsprozess, indem ein Bestätigungscode an die angegebene E-Mail-Adresse gesendet wird. Wird sowohl für die Erstellung neuer Kontakte als auch für die Aktualisierung von E-Mail-Adressen verwendet.
2. `AssociateAwsTrainingCertificationEmailDomain` API: Ordnet eine E-Mail-Domain der AWS-Schulung und -Zertifizierung für das Partnerkonto zu und ermöglicht so die automatische Überprüfung von Mitarbeiterzertifizierungen.
3. `DisassociateAwsTrainingCertificationEmailDomain` API: Entfernt die Verknüpfung zwischen einer E-Mail-Domain und AWS-Schulung und -Zertifizierung für das Partnerkonto.

Mit Partnerkontoverbindungen arbeiten

Partner können mithilfe der Partner Connection API Verbindungen zwischen ihren eigenen Konten oder Verbindungen mit AWS Konten anderer AWS Partner verwalten. Der Prozess umfasst zwei Phasen:

1. Phase der Verbindungseinladung: Verbindungsanfragen initiieren und beantworten
2. Phase der aktiven Verbindung: Verwaltung etablierter Verbindungen und der damit verbundenen Geschäftsaktivitäten

Eine Verbindungseinladung erstellen und senden

Schritt 1: Erstellen Sie eine Verbindungseinladung

Der erste Schritt beim Herstellen einer Partnerverbindung besteht darin, mithilfe der `CreateConnectionInvitation` Aktion eine Verbindungseinladung zu erstellen und zu senden. Beim Erstellen einer Einladung müssen Sie Folgendes angeben:

- Katalog: Der AWS Katalog (AWS oder die Sandbox), in dem die Verbindung verwaltet wird

- **ConnectionType:** Die Art der Beziehung, die Sie aufbauen möchten. Wählen Sie eine der folgenden Optionen:
 - **OPPORTUNITY_COLLABORATION:** Verwenden Sie diesen Typ, wenn Sie eine Verbindungsanfrage an das Konto eines anderen Partners senden möchten, sodass Sie ihm Einladungen zu Opportunity-Engagements zur Zusammenarbeit senden können
 - **SUBSIDIARY:** Verwenden Sie diesen Typ, wenn Sie mehrere AWS Marketplace-Verkäuferkonten, die Sie besitzen, mit Ihrem „primären“ Konto verbinden möchten, das Sie mit APN verknüpft haben oder bei dem Sie als Partner registriert sind.
- **ReceiverIdentifier:** Die öffentliche Partnerprofil-ID des Empfängers (für den Verbindungstyp Opportunity Collaboration) oder die AWS Konto-ID (für den Verbindungstyp Tochtergesellschaft)
- **E-Mail:** Ihre Kontakt-E-Mail-Adresse für die Kommunikation
- **Name:** Ihr Name als Absender
- **Nachricht:** Eine Beschreibung, die den Zweck der Verbindungsanfrage erklärt

Nach der Erstellung wechselt die Einladung in den Status Ausstehend und wartet auf eine Aktion durch den Empfänger.

 Important

Durch das Senden einer Verbindungseinladung erklären Sie sich damit einverstanden, Ihre AWS Konto-ID an den Empfänger weiterzugeben, falls dieser die Einladung annimmt. Diese Offenlegung ist erforderlich, um die Verbindung auf Kontoebene herzustellen.

Schritt 2: Überwachen Sie Ihre gesendeten Einladungen

Sie können den Status der von Ihnen gesendeten Einladungen mithilfe der `ListConnectionInvitations` Aktion verfolgen, bei der der `ParticipantType` Parameter auf `SENDER` gesetzt ist. Das ermöglicht Ihnen Folgendes:

- Alle ausgehenden Einladungen anzeigen
- Nach Status filtern (PENDING,ACCEPTED,REJECTED,EXPIRED)
- Nach Verbindungstyp filtern
- Schauen Sie sich spezifische Einladungen an andere Partner an

Schritt 3: Eine Einladung stornieren (optional)

Wenn Sie eine ausstehende Einladung zurückziehen müssen, bevor sie akzeptiert wurde, können Sie die `CancelConnectionInvitation` Aktion verwenden. Beachten Sie Folgendes:

- Sie können nur Einladungen stornieren, die den Status Ausstehend haben
- Sobald eine Einladung angenommen und eine Verbindung hergestellt wurde, kann sie nicht mehr storniert werden
- Um eine bestehende Verbindung zu beenden, müssen Sie stattdessen die `CancelConnection` Aktion verwenden

Verbindungseinladungen empfangen und beantworten

Schritt 1: Eingehende Einladungen auflisten

Um Verbindungseinladungen anzuzeigen, die Sie erhalten haben, verwenden Sie die `ListConnectionInvitations` Aktion, bei der der `ParticipantType` Parameter auf `gesetzt` ist `RECEIVER`. Sie können nach folgenden Kriterien filtern:

- Verbindungstyp
- Status (`PENDING`, `ACCEPTED`, `REJECTED`)
- Kennung des Absenders

Schritt 2: Überprüfen Sie die Einladungsdetails

Verwenden Sie die `GetConnectionInvitation` Aktion, um die vollständigen Details einer bestimmten Einladung anzuzeigen, einschließlich:

- Name und Kontakt-E-Mail des Absenders
- Einladungsnachricht mit Erläuterung des Zwecks
- Verbindungstyp wird angefordert
- Ablaufdatum
- Öffentliche Profilinformationen des Absenders

Schritt 3: Nehmen Sie die Einladung an oder lehnen Sie sie ab

Nachdem Sie die Einladungsdetails überprüft haben, haben Sie zwei Möglichkeiten:

Option A: Nehmen Sie die Einladung an

Verwenden Sie die `AcceptConnectionInvitation` Aktion, um die Verbindung herzustellen. Wenn du akzeptierst:

- Im Status Aktiv wird eine neue Verbindung erstellt
- Der Status der Einladung ändert sich in „Angenommen“
- Ihre AWS Konto-ID wird dem Absender angezeigt
- Sie können mit der Durchführung von Geschäftsaktivitäten beginnen, die durch den Verbindungstyp aktiviert werden

Important

Wichtige Zustimmung: Indem Sie eine Verbindungseinladung annehmen, erklären Sie sich damit einverstanden, dem Absender Ihre AWS Konto-ID mitzuteilen. Diese Offenlegung ist erforderlich, um die Verbindung auf Kontoebene herzustellen.

Option B: Lehnen Sie die Einladung ab

Verwenden Sie die `RejectConnectionInvitation` Aktion, wenn Sie die Verbindung nicht herstellen möchten. Sie können optional einen Grund für die Ablehnung angeben. Nach der Ablehnung:

- Der Status der Einladung ändert sich in Abgelehnt
- Es wurde keine Verbindung hergestellt
- Die Einladung wird nicht mehr in Ihrer Liste der ausstehenden Einladungen angezeigt

Automatischer Ablauf

Wenn innerhalb des Ablaufzeitraums keine Maßnahmen ergriffen werden, versetzt das System die Einladung automatisch in den Status Abgelaufen. Abgelaufene Einladungen können nicht angenommen oder abgelehnt werden.

Aktive Verbindungen verwalten

Ihre Verbindungen anzeigen

Sobald eine Verbindung hergestellt ist, können beide Parteien sie mithilfe der folgenden Aktionen anzeigen und verwalten:

ListConnections: Rufen Sie eine Liste Ihrer Verbindungen mit Filteroptionen ab:

- Filtern Sie nach Verbindungstyp und Status (z. B. `OPPORTUNITY_COLLABORATION:ACTIVE`)
- Filtern Sie nach der Kennung eines anderen Teilnehmers
- Übersichtsinformationen für jede Verbindung anzeigen

GetConnection: Ruft vollständige Details einer bestimmten Verbindung ab, einschließlich:

- Alle Verbindungsarten, die mit dieser Partnerbeziehung verknüpft sind
- Status der einzelnen Verbindungstypen (Aktiv oder Beendet)
- Kontaktinformationen aus der ursprünglichen Einladung
- AWS Konto-IDs und öffentliche Profil-IDs beider Parteien
- Zeitstempel für Erstellung und Beendigung für jeden Verbindungstyp

Hinzufügen von Verbindungstypen zu einer bestehenden Verbindung

Wenn Sie bereits eine aktive Verbindung zu einem Partner haben und eine neue Art von Beziehung aufbauen möchten, können Sie eine neue Verbindungseinladung mit einem anderen Verbindungstyp senden. Bei Annahme:

- Der neue Verbindungstyp wird der bestehenden Verbindung hinzugefügt
- Beide Verbindungsarten bleiben unabhängig voneinander verwaltbar
- Für die Beziehung wird dieselbe Verbindungs-ID beibehalten

Note

Es kann immer nur eine aktive Verbindung eines bestimmten Typs zwischen zwei Konten in einem bestimmten Katalog bestehen.

Verbindungstypen beenden

Jede Partei kann einen bestimmten Verbindungstyp mithilfe der `CancelConnection` Aktion beenden. Beim Beenden:

- Geben Sie die Verbindungs-ID und den Verbindungstyp an, der beendet werden soll
- Geben Sie einen Grund für die Kündigung an
- Der Status des Verbindungstyps ändert sich zu `CANCELED`
- Andere Verbindungstypen innerhalb derselben Verbindung bleiben davon unberührt

Vollständiger Verbindungsabbruch

Wenn alle Verbindungstypen innerhalb einer Verbindung beendet wurden, wechselt die Verbindung selbst in den Status `Beendet`. Eine vollständig beendete Verbindung:

- Kann nicht reaktiviert werden, aber Sie können eine neue Verbindungsanfrage senden, um die Geschäfte mit diesem Konto fortzusetzen
- Verbleibt im System zur Aufbewahrung historischer Aufzeichnungen
- Kann über die API eingesehen werden `GetConnection`
- Erfordert eine neue Einladung, um die Beziehung wiederherzustellen

Überwachung von Verbindungsereignissen

Partner sollten Ereignisse im Zusammenhang mit Verbindungen über Amazon verfolgen, um über Folgendes auf dem Laufenden EventBridge zu bleiben:

- Neue eingehende Verbindungseinladungen
- Statusänderungen bei gesendeten Einladungen (`ACCEPTED`, `REJECTED`, `EXPIRED`)
- Verbindungsabbrüche, die von der anderen Partei initiiert wurden

Wenn Sie ein Ereignis empfangen, verwenden Sie die entsprechende Get API-Aktion, um die neuesten Details abzurufen:

- `GetConnectionInvitation` für Einladungsaktualisierungen
- `GetConnection` für Verbindungsaktualisierungen

Mögliche Ereignisse

- **Verbindungseinladung erstellt:** Informiert das Empfängerkonto über eingehende Verbindungseinladungen.
- **Verbindungseinladung storniert:** Benachrichtigt das Empfängerkonto, wenn ein Absender eine eingehende Verbindungseinladung storniert.
- **Verbindungseinladung abgelaufen:** Benachrichtigt sowohl das Absender- als auch das Empfängerkonto, wenn eine Verbindungseinladung untätig abläuft.
- **Verbindungseinladung abgelehnt:** Benachrichtigt das Absenderkonto, wenn die Verbindungseinladung vom Empfänger abgelehnt wird.
- **Verbindungseinladung akzeptiert:** Benachrichtigt das Absenderkonto, wenn seine Verbindungseinladung angenommen und eine Verbindung hergestellt wurde.
- **Verbindung abgebrochen:** Benachrichtigt beide verbundenen Parteien, wenn alle Verbindungen vollständig beendet wurden.

Verbindungsstatus verstehen

Status der Verbindungseinladungen

Status	Description	Kann übergehen zu
PENDING	Ausgangszustand bei der Erstellung; wartet auf die Aktion des Empfängers	ACCEPTED, REJECTED, EXPIRED, CANCELED
ACCEPTED	Empfänger akzeptiert; Verbindung hergestellt	Keine (Terminalstatus)
REJECTED	Empfänger hat abgelehnt ; enthält einen optionalen Ablehnungsgrund	Keine (Endstatus)
EXPIRED	System-expired da innerhalb der Ablaufzeit keine Maßnahmen ergriffen wurden	Keine (Endstatus)

Status	Description	Kann übergehen zu
CANCELED	Der Absender hat die Einladung vor der Annahme zurückgezogen	Keine (Endstatus)

Status des Verbindungstyps

Status	Description
ACCEPTED	Der Verbindungstyp ist betriebsbereit; zugehörige Geschäftsaktivitäten sind aktiviert
CANCELED	Der Verbindungstyp wurde von einer der Parteien beendet

Bewährte Methoden

1. Klare Kommunikation: Geben Sie in Ihren Verbindungseinladungen detaillierte, klare Botschaften an, in denen Sie den Zweck und die erwartete Zusammenarbeit erläutern.
2. Rechtzeitige Antworten: Überwachen Sie Verbindungseinladungen und beantworten Sie sie umgehend, um ein automatisches Ablaufen zu vermeiden.
3. Regelmäßige Überprüfung: Überprüfen Sie regelmäßig Ihre aktiven Verbindungen, um sicherzustellen, dass sie Ihren aktuellen Geschäftsanforderungen entsprechen.
4. Ordnungsgemäße Kündigung: Wenn Sie eine Geschäftsbeziehung beenden, sollten Sie die CancelConnection Maßnahme mit einem klaren Grund ergreifen, um die professionelle Kommunikation aufrechtzuerhalten.
5. Ereignisüberwachung: Richten Sie EventBridge Regeln ein, um automatisch Benachrichtigungen über Änderungen des Verbindungsstatus zu erhalten, damit Sie über wichtige Updates auf dem Laufenden bleiben.
6. Verwaltung von Verbindungstypen: Überlegen Sie sich, welche Verbindungstypen erforderlich sind, bevor Sie Verbindungen herstellen, da jeder Typ unterschiedliche Geschäftsfunktionen ermöglicht.

7. Sicherheitsbewusstsein: Denken Sie daran, dass beim Herstellen einer Verbindung AWS Konto-IDs zwischen den Parteien offengelegt werden. Stellen Sie nur eine Verbindung mit vertrauenswürdigen Partnern her.

Verbindungsarten und ihre Zwecke

Art der Verbindung	Zweck	Anwendungsfall
OPPORTUNITY_COLLABORATION	Ermöglicht die Zusammenarbeit bei Co-Selling-Möglichkeiten mit anderen Partnern	Wenn Sie möchten, dass andere Partner Zugriff auf Ihre Verkaufschance haben oder umgekehrt, verwenden Sie diesen Verbindungstyp. Diese Verbindung ermöglicht es Ihnen, Einladungen zur Teilnahme an Geschäftschancen an verbundene Partner zu senden.
SUBSIDIARY	Stellt Verbindungen zwischen Ihren mehreren AWS Marketplace-Verkäuferkonten und Ihrem primären Partnerkonto her	Verwenden Sie diese Option, wenn Sie mehrere AWS Marketplace-Verkäuferkonten haben, die Sie mit Ihrem Hauptkonto verbinden möchten, in dem Sie als Partner und Verkäufer registriert sind.

Mit Partner Verification arbeiten

Verwaltung der Partnerverifizierung

Die Partnerverifizierung ist eine zwingende Voraussetzung für die Registrierung eines Partnerkontos. Bevor Partner ein Partnerkonto erstellen können, müssen sie die Prozesse zur Geschäfts- und Identitätsprüfung abschließen. Diese Überprüfungen bestätigen, dass das Unternehmen rechtmäßig registriert ist, und bestätigen die Identität der Person, die das AWS Konto registriert.

Partner können die verfügbaren Verifizierungs-APIs verwenden, um Überprüfungsprozesse einzuleiten und zu überwachen:

Während der Überprüfung

1. **Initiierung der Unternehmensverifizierung** — Partner leiten die Geschäftsverifizierung ein, indem sie die `StartVerification` API aufrufen und dabei Einzelheiten zur Unternehmensregistrierung wie Firmenname, Registrierungs-ID, Landesvorwahl und Gerichtsstand der Gründung angeben.
2. **Initiierung der Identitätsprüfung** — Partner leiten die Identitätsprüfung ein, indem sie die `StartVerification` API mit Informationen zum Registranten aufrufen. Die API gibt eine sichere Abschluss-URL zurück, über die der Registrant den Workflow zur Identitätsprüfung mit einem von der Regierung ausgestellten Ausweis abschließt.
3. **Überwachung des Bestätigungsstatus** — Partner verwenden die `GetVerification` API, um den aktuellen Status und Details der Überprüfungsprozesse abzurufen. Die API gibt Überprüfungstyp, Status, Zeitstempel und verifizierungsspezifische Details zurück.
4. **Abschluss der Überprüfung** — Sowohl die Unternehmensverifizierung als auch die Identitätsprüfung müssen `SUCCEEDED` den Status erreicht haben, bevor Partner mit der Erstellung eines Partnerkontos fortfahren können. Fehlgeschlagene oder abgelaufene Überprüfungen müssen vor der Kontoregistrierung geklärt werden.

Nach der Überprüfung

1. **Erstellung eines Partnerkontos** — Nach erfolgreicher Überprüfung erstellen Partner ihr Partnerkonto mithilfe der in der Dokumentation zum Partnerkonto beschriebenen `CreatePartner` API.
2. **Abrufen von Bestätigungsaufzeichnungen** — Partner können die Bestätigungsdetails jederzeit über die `GetVerification` API abrufen, wobei die Bestätigungs-ID bei der Initiierung zurückgegeben wird.

Übersicht über API-Befehle

1. **`StartVerification` API**: Partner initiieren Verifizierungsprozesse, indem sie die `StartVerification` API aufrufen. Zur Unternehmensverifizierung validiert die API die Angaben zur Unternehmensregistrierung. Für die Identitätsprüfung generiert die API eine zeitlich begrenzte Abschluss-URL, über die der Registrant die Identitätsprüfung abschließen kann.

2. **GetVerificationAPI:** Nach dem Start der Überprüfung verwenden Partner die GetVerification API, um den Status und die Details der Überprüfung abzurufen. Die API gibt den aktuellen Status, Zeitstempel und verifizierungsspezifische Informationen zurück.

Verwendung der AWS Vertriebs-API von Partner Central

Diese API-Referenz für AWS Partner Central soll [AWS Partnern](#) helfen, CRM-Systeme (Customer Relationship Management) in Partner Central zu integrieren. Die API automatisiert Interaktionen mit Partner Central, was dazu beiträgt, effektive Interaktionen im Rahmen gemeinsamer Geschäftsaktivitäten sicherzustellen.

Die API bietet AWS Standard-API-Funktionen. Greifen Sie darauf zu, indem Sie entweder [API-Aktionen](#) oder ein AWS SDK verwenden, das auf Ihre Programmiersprache oder Plattform zugeschnitten ist. Weitere Informationen finden Sie unter [Erste Schritte AWS](#) und [Tools, auf denen Sie aufbauen können AWS](#).

Funktionen, die angeboten werden von AWS Partner Central API

1. Chancenmanagement: Erleichtert die Verwaltung von Verkaufschancen AWS mithilfe von API-Aktionen wie `CreateOpportunity`, `UpdateOpportunity`, `ListOpportunities`, `GetOpportunity`, und `AssignOpportunity`.
2. AWS Empfehlungsmanagement: Erleichtert das Empfangen von Empfehlungen, die geteilt werden, AWS indem Aktionen wie `ListEngagementInvitations`, `GetEngagementInvitation`, `StartEngagementByAcceptingInvitation` und verwendet werden. `RejectEngagementInvitation`
3. Zuordnung von Entitäten: Ordnen Sie mithilfe der Aktionen `AssociateOpportunity` und verwandte Entitäten wie AWS Marketplace Produkte, AWS Marketplace Partnerlösungen, Lösungen, Produkte und AWS Marketplace private Angebote Opportunities zu. `DisassociateOpportunity`
4. AWS Opportunity-Details anzeigen: Verwenden Sie die `GetAWSOpportunitySummary` Aktion, um in Echtzeit Zusammenfassungen der AWS Opportunities abzurufen, die mit Ihren Opportunities verknüpft sind.
5. Lösungen auflisten: Bietet Listen-APIs für die Auflistung von Lösungen, die Partner anbieten. `ListSolutions`

6. Event-Abonnement: Partner können Echtzeit-Updates zu Opportunities abonnieren, indem sie sich Events wie „Opportunity Created“, „Opportunity Updated“, „Engagement Invitation Accepted“, „Engagement Invitation Rejected“ und „Engagement Invitation Created via Amazon“ anhören EventBridge.

Unterstützte Regionen und Endpunkte

Die AWS Partner Central API ist in der Region USA Ost (Nord-Virginia) verfügbar.

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

Partner können API-Integrationen in einer sicheren Sandbox-Umgebung testen und validieren. Auf diese Weise können Sie API-Aktionen testen, ohne Live-Daten zu beeinträchtigen. Weitere Informationen finden Sie unter [Testen in einer Sandbox für AWS Vertriebs-API von Partner Central](#).

Verkauf von API-Entitäten

AWS Die Partner Central-Einheiten stellen wichtige Geschäftskomponenten dar, die bei der Zusammenarbeit zwischen Partnern und verwendet werden. AWS Diese Einheiten fassen Informationen zu Geschäftschancen, Lösungen, Produkten und Angeboten zusammen und ermöglichen so eine reibungslose Zusammenarbeit und Verwaltung gemeinsamer Vertriebsaktivitäten. In den folgenden Abschnitten werden die Kerneinheiten der Partner Central Selling API beschrieben.

Gelegenheit

Eine Verkaufschance ist ein potenzieller Verkauf oder ein Geschäft, das ein Unternehmen identifiziert und aktiv verfolgt. Es handelt sich um einen qualifizierten Interessenten oder Interessenten mit einem bestimmten Bedarf, den die Produkte oder Dienstleistungen des Unternehmens erfüllen können. Geschäftschancen werden in der Regel in einer Vertriebspipeline oder einem CRM-System verfolgt und bilden die Grundlage für future Umsätze. Effektives Chancenmanagement beinhaltet die Pflege von Leads während des gesamten Verkaufsprozesses, von der ersten Qualifizierung bis zum Abschluss. Weitere Informationen finden Sie unter [Arbeiten mit Ihren Opportunities](#) und [Datentypen](#)

Partnerlösungen

Stellt eine Partnerlösung dar (in AWS Partner Central als Angebot bezeichnet), bei der es sich um ein Softwareprodukt oder eine Beratungsfirma handelt, die von AWS Partnern entwickelt und bereitgestellt wird. Partnerlösungen helfen Kunden dabei, spezifische geschäftliche Herausforderungen zu bewältigen oder bestimmte Ziele mithilfe von AWS Services zu erreichen. Weitere Informationen finden Sie unter [Was ist eine Lösung?](#)

AWS product

Steht für eine bestimmte AWS Dienstleistung oder ein bestimmtes Produkt. AWS bietet eine breite Palette von Produkten und Dienstleistungen für skalierbare, zuverlässige und kostengünstige Infrastrukturlösungen. Partner können die aktuelle AWS Produktliste [auf der Seite für den Massenimport in Partner Central](#) abrufen (Import starten >AWS Produkte und Lösungen). Weitere Informationen finden Sie unter [AWS Produkte](#) oder in [der Liste aller AWS Produkte](#).

AWS Marketplace privates Angebot

AWS Marketplace Ein privates Angebot ist eine Funktion, mit der AWS Marketplace Verkäufer einzelnen AWS Kunden bestimmte Preise und Konditionen anbieten können. Über private Angebote können Verkäufer individuelle Preise, Zahlungspläne und Lizenzbedingungen für Endbenutzer aushandeln. AWS Kunden können Softwarelösungen erwerben, die ihren spezifischen Anforderungen entsprechen, und profitieren gleichzeitig möglicherweise von günstigeren Konditionen oder Preisen als bei Standardangeboten. Beim privaten Angebotsverfahren erstellt der Verkäufer ein einzigartiges Angebot mit maßgeschneiderten Konditionen, das dann privat an den ausgewählten AWS Kunden weitergegeben wird, damit dieser es prüfen und annehmen kann. Weitere Informationen finden Sie unter [Private Angebote unter AWS Marketplace](#).

Einladung zur Verlobung

Eine Einladung zur Zusammenarbeit bezieht sich auf eine formelle Anfrage von Partnern AWS zur Zusammenarbeit bei einer bestimmten Empfehlung. Auf diese Weise können der Partner AWS und der Partner zusammenarbeiten, um die Geschäftschance voranzutreiben. Die Einladung kann vom Partner angenommen oder abgelehnt werden.

Themen

- [Wir nutzen Ihre Möglichkeiten](#)
- [Arbeiten mit Möglichkeiten von AWS](#)
- [Mit Opportunity-Updates arbeiten](#)

- [Zusammenarbeit mit Möglichkeiten für mehrere Partner](#)
- [Möglichkeiten zuordnen, trennen und zuweisen](#)
- [Arbeiten Sie mit Ihren Leads](#)
- [Arbeiten mit Erkenntnissen zum Geschäftsumfang](#)
- [Bewährte Methoden](#)

Wir nutzen Ihre Möglichkeiten

Was ist eine Opportunity?

In der Anfangsphase des Verkaufsprozesses bewertet ein Vertriebsmitarbeiter, ob eine interessierte Person (Lead genannt) das Potenzial hat, Kunde zu werden. Diese Bewertungs- und Validierungsphase wird als Qualifikation bezeichnet. Sobald ein Lead als qualifiziert eingestuft wird und bei dem davon ausgegangen wird, dass es eine höhere Wahrscheinlichkeit hat, zu einem Kunden zu werden, wird er als Opportunity eingestuft.

Arbeiten Sie mit Ihren Geschäftschancen

Partner können Verkaufschancen verwalten, die in ihren CRM-Systemen erstellt wurden, und sie mithilfe der AWS Partner Central Selling API mit AWS Partner Central synchronisieren. Auf diese Weise können Partner Geschäftschancen von der Initiierung bis zum Abschluss verfolgen und verwalten.

Eine Geschäftschance schaffen

Der erste Schritt bei der Verwaltung von Geschäftschancen besteht darin, mithilfe der `CreateOpportunity` Aktion eine Geschäftschance zu schaffen. Dadurch entsteht eine Geschäftschance mit der `Lifecycle.ReviewStatus` Einstellung von `Pending Submission`. Die Opportunity wurde jedoch noch nicht AWS zur Validierung eingereicht.

Nachdem die Opportunity erstellt wurde, müssen Sie der Opportunity, die die `AssociateOpportunity` Aktion verwendet, mindestens eine AWS Marketplace Partnerlösung, eine Partnerlösung oder ein AWS Marketplace Produkt zuordnen. Durch diese Aktion wird klar definiert, was mit der Opportunity verkauft werden soll. Mithilfe der `ListSolutions` API können Sie die vollständige Liste der verfügbaren Lösungen in Ihrem Konto einsehen und zwischen einer und zehn Lösungen zuordnen.

Optional können Sie mithilfe der `AssociateOpportunity` Aktion auch relevante AWS Produkte mit der Opportunity verknüpfen. Dieser Schritt hilft den AWS Vertriebsteams zu verstehen, welche AWS Produkte voraussichtlich im Zusammenhang mit der Verkaufschance verkauft werden.

Nachdem die erforderliche Lösung oder das Produkt zugeordnet und optional die AWS Produkte verknüpft wurden, können Sie mithilfe der `StartEngagementFromOpportunityTask` Aktion eine Interaktion mit der Verkaufschance starten. In diesem Fall reichen Sie die Möglichkeit ein, ein Engagement zu starten.

Prozess überprüfen

Nachdem das Engagement für die Opportunity mithilfe der `StartEngagementFromOpportunityTask` Aktion gestartet wurde, geht die Opportunity in die AWS Validierungsphase über, und sie `Lifecycle.ReviewStatus` ist auf `Submitted` eingestellt. Solange der Überprüfungsprozess nicht abgeschlossen ist, können keine Änderungen an der Opportunity vorgenommen werden.

Stellt während dieser Überprüfungsphase AWS sicher, dass die Angaben zur Verkaufschance korrekt und vollständig sind. Während der Validierung `Lifecycle.ReviewStatus` ist der auf `eingestelltIn-Review`.

Wenn der Partner Änderungen oder zusätzliche Informationen benötigt, AWS setzt er den Wert `Lifecycle.ReviewStatus` auf `Action Required`, und alle erforderlichen Aktualisierungen werden über das `Lifecycle.ReviewComments` Feld mitgeteilt.

Sobald die Opportunity die Validierung bestanden hat, werden sie `Lifecycle.ReviewStatus` geändert `Approved`, sodass die Opportunity für Co-Selling-Aktivitäten bereit ist.

Partner sollten die Opportunity Updated Veranstaltung über Amazon verfolgen EventBridge. Dadurch werden sie über Statusänderungen oder Feedback von informiert AWS. Nach Erhalt der Veranstaltung können Partner die `GetOpportunity` API-Aktion verwenden, um die neuesten Informationen zur Verkaufschance abzurufen und das `Lifecycle.ReviewStatus` Feld zu verifizieren.

Lösung von Validierungsproblemen

Wenn das auf eingestellt `Lifecycle.ReviewStatus` ist `Action Required`, müssen sich die Partner mit den unter AWS hervorgehobenen Problemen befassen. Um diese zu lösen, können Partner die Opportunity mithilfe der `UpdateOpportunity` API-Aktion aktualisieren.

Während des Action Required Status können nur bestimmte Felder bearbeitet werden. Zu diesen Feldern gehören:

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

Nachdem Sie die erforderlichen Änderungen vorgenommen haben, geht die Opportunity erneut in die Validierungsphase über, und der Vorgang wird wiederholt, bis die Opportunity auf oder gesetzt Lifecycle.ReviewStatus ist. Approved Disqualified

Post-Approval Aktualisierungen

Sobald sich die Gelegenheit bietet Approved, können die Partner die Verkaufschance mithilfe der UpdateOpportunity Aktion bei Bedarf weiter aktualisieren und so reibungslose Co-Selling-Aktivitäten ermöglichen.

Partner sollten die Opportunity Updated Ereignisse weiterhin über Amazon verfolgen EventBridge , um über etwaige Änderungen auf dem Laufenden zu bleiben. Weitere Informationen zur Nachverfolgung von AWS Updates finden Sie im Abschnitt „Mit Opportunity-Updates arbeiten“. Partner können auch ausgewählte Felder auf der Grundlage der Geschäftsvalidierungsregeln aktualisieren.

Arbeiten mit Möglichkeiten von AWS

1. Empfangen der AWS Gelegenheit

Opportunities werden mit Partnern geteilt, wenn ein AWS Vertriebsleiter einen Partner mit einer Opportunity im CRM-System AWS verknüpft. Diese werden als AWS Opportunities bezeichnet und unterscheiden sich von Opportunities, die im eigenen Konto des Partners erstellt wurden.

Wenn einer AWS Opportunity ein Partner angehängt ist, AWS wird eine Einladung zum Engagement erstellt, die eine Teilmenge der Daten aus der AWS Opportunity enthält. Partner erhalten eine Veranstaltung mit dem Titel Engagement Invitation Created.

2. Überprüfung der Einladung zur Verlobung

Die Einladung zur Verlobung enthält wichtige Informationen wie `Project.TitleProject.CustomerUseCase,Lifecycle.Stage,Project.CustomerBusinessProb` und einige zusätzliche Felder. Partner können anhand dieser Daten entscheiden, ob sie die Gelegenheit nutzen möchten.

Die folgenden Felder aus der AWS Opportunity sind jedoch nicht in der Einladung zum Engagement enthalten:

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

3. Bearbeitung der Einladung zur Verlobung

Nach Erhalt einer Veranstaltung, die zu einem Engagement eingeladen wurde, können Partner die `GetEngagementInvitation` Aktion nutzen, um Einzelheiten zu der AWS Verkaufschance abzurufen.

Wenn der Partner beschließt, die Gelegenheit nicht zu nutzen, kann er die Einladung ablehnen, indem er die `RejectEngagementInvitation` Aktion zusammen mit den erforderlichen Aktionen `ablehntRejectionReason`. Nach der Ablehnung geht der Zugriff auf die Verkaufschance verloren.

Um die Einladung anzunehmen und fortzufahren, sollten Partner die `StartEngagementByAcceptingInvitationTask` Aktion nutzen. Dies ist eine asynchrone Aktion, die nacheinander die folgenden Aufgaben ausführt:

1. Nimmt die Einladung zur Verlobung an.
2. Erstellt mithilfe von Daten aus der Opportunity eine neue AWS Opportunity im Konto des Partners.
3. Enthält zusätzliche Details, die zur Identifizierung des Kunden im Konto des Partners erforderlich sind.

Nach Abschluss wird das Ereignis „Opportunity Created“ mit der entsprechenden Opportunity-ID ausgelöst. Partner können diese ID dann in der `GetOpportunity` Aktion verwenden, um alle Details der Opportunity abzurufen.

Wenn der Partner keine Ereignisse verwendet, kann er mit derselben Payload `StartEngagementByAcceptingInvitationTask` erneut anrufen, um den aktuellen Status zu überprüfen.

4. Verwaltung der AWS Gelegenheit Post-Acceptance

Sobald sich die Opportunity im Konto des Partners befindet, kann sie wie jede andere Opportunity im System verwaltet und aktualisiert werden. Partner können beispielsweise Aktionen `nutzenUpdateOpportunity`, um alle erforderlichen Änderungen vorzunehmen.

Weitere Informationen zur Nachverfolgung von AWS Updates und zur Verwaltung von Opportunity-Updates finden Sie im [Mit Opportunity-Updates arbeiten](#) Abschnitt.

Mit Opportunity-Updates arbeiten

Aktualisierung von Geschäftschancen

Partner können die `UpdateOpportunity` Aktion nutzen, um Opportunities zu aktualisieren, wobei spezifische Regeln festlegen, welche Felder aktualisiert werden und wann sie aktualisiert werden:

1. Aktualisierungen können nicht vorgenommen werden, wenn dies `Submitted` oder `Lifecycle.ReviewStatus` ist `In-Review`.
2. Vor der Einreichung können Partner Aktualisierungen vornehmen, verarbeiten AWS diese jedoch nur, wenn die `StartEngagementFromOpportunityTask` Aktion aufgerufen wird.
3. Wenn die Opportunity den In-review Status „SubmittedOr“ hat, werden alle Updates blockiert.

4. Wenn sich die Opportunity im Action Required Status befindet, werden ausgewählte Felder für Updates AWS geöffnet. Zu diesen Feldern gehören:

- `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. Nach dem Überprüfungsprozess (d. h. wann `LifeCycle.ReviewStatus` ist auf `eingestelltApproved`) können die folgenden Felder nicht mehr aktualisiert werden:

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

6. Für alle anderen Felder können Aktualisierungen mithilfe der `UpdateOpportunity` Aktion vorgenommen werden. Es können jedoch zusätzliche Einschränkungen gelten, die auf den Geschäftsregeln für das jeweilige Programm oder die Art der Geschäftschance basieren. Weitere Informationen finden Sie unter Validierungen auf Feldebene.

7. Für alle Aktualisierungen, die sowohl über die Benutzeroberfläche als auch über die API

Verwenden der Verkaufs-API
vorgenommen werden, wird das Ereignis „Opportunity Updated“ generiert.

Empfangen von Updates von AWS über Möglichkeiten

AWS aktualisiert in der Regel AWS Opportunities, und jedes Mal, wenn eine Aktualisierung vorgenommen wird, wird das Ereignis „Opportunity Updated“ generiert.

Um die neuesten Updates abzurufen von AWS, müssen Partner zwei separate Aktionen aufrufen

1. `GetOpportunity` um Einzelheiten zur Geschäftschance des Partners abzurufen.
2. `GetAWSOpportunitySummary` um Zusammenfassungen der AWS Opportunity-Daten in Echtzeit abzurufen.

Die meisten regelmäßigen Updates von AWS werden über die `GetAWSOpportunitySummary` Antwort verfügbar sein. Gelegentlich AWS kann es jedoch vorkommen, dass Attribute im Rahmen der Opportunity des Partners direkt aktualisiert werden.

Um Updates zu konsumieren von AWS

1. Rufen Sie die `GetAWSOpportunitySummary` Aktion auf, um die neuesten Details der AWS Opportunity abzurufen.
2. Wenn Änderungen in der Opportunity des Partners berücksichtigt werden müssen, verwenden Sie die `UpdateOpportunity` Aktion, um die relevanten Daten in die Opportunity des Partners zu kopieren.

Partner können diesen Prozess als direkten Aktualisierungsmechanismus automatisieren oder einen manuellen Überprüfungsprozess zur Validierung und Aktualisierung der Daten implementieren.

Zusammenarbeit mit Möglichkeiten für mehrere Partner

AWS Partner s kann AWS als aktiver Teilnehmer gemeinsam an Möglichkeiten arbeiten.

Themen

- [Engagements und Schnappschüsse](#)
- [Erstellen einer benutzerdefinierten Richtlinie für `ResourceSnapshotJobRole`](#)
- [Partner zu einer Geschäftschance einladen](#)
- [Details zur Einladung zur Interaktion werden abgerufen](#)
- [Auf eine Einladung zur Verlobung antworten](#)

- [Prüfung und Aktualisierung von Möglichkeiten für mehrere Partner](#)
- [Verwenden von Snapshots, um Partner-Updates zu erhalten](#)
- [Engagement-Mitglieder anzeigen](#)
- [Den Status des Ressourcen-Snapshots überwachen](#)

Engagements und Schnappschüsse

Ein Engagement ist eine Ressource, die AWS für die Zusammenarbeit zwischen mehreren AWS Partner Accounts und für eine AWS Kundenchance genutzt wird. Engagements ermöglichen den sicheren Informationsaustausch zwischen Partnern und die Zusammenarbeit, während gleichzeitig die individuelle Eigenverantwortung und Kontrolle über Geschäftschancen gewahrt bleibt. Engagements umfassen Möglichkeiten, die sich AWS sowohl von Partnern als auch von Partnern AWS als aktiver Teilnehmer ergeben. Partner können andere Partner einladen AWS, an einem Engagement teilzunehmen, indem Sie EngagementInvitations Wenn eingeladene Partner zustimmen, erhalten sie innerhalb desselben Engagements ihre eigene Chance.

Engagement-Mitglieder teilen ihre Fortschritte anhand von Schnappschüssen — zu einem bestimmten Zeitpunkt, unveränderlichen Kopien bestimmter Felder aus einer zugrunde liegenden Ressource, z. B. einer Opportunity. Schnappschüsse werden innerhalb des Engagements erstellt und mit Mitgliedern geteilt. Wenn sich die zugrunde liegende Ressource ändert, kann der Eigentümer eine neue Snapshot-Revision erstellen, um Aktualisierungen widerzuspiegeln. AWS stellt Snapshot-Jobs bereit, d. h. Aufträge, die sich im Eigentum des Kunden befinden und bei Änderungen an der Ressource automatisch neue Revisionen erstellen. Sobald sie geteilt wurden, bleiben Snapshots für Engagement-Mitglieder dauerhaft zugänglich.

Erstellen einer benutzerdefinierten Richtlinie für ResourceSnapshotJobRole

Wenn Sie an Geschäftschancen mit mehreren Partnern zusammenarbeiten möchten, müssen Sie im Rahmen eines Engagements Schnappschüsse Ihrer Geschäftschancen mit anderen Partnern teilen. Um weiterhin Zugriff auf die neuesten Opportunity-Details zu haben, müssen Sie eine benutzerdefinierte Rolle mit dem Namen erstellen. ResourceSnapshotJobRole Diese Rolle ermöglicht es dem System, in Ihrem Namen Schnappschüsse Ihrer Opportunities zu erstellen und Schnappschüsse von anderen Partnern im selben Engagement abzurufen. Ohne diese Rolle sind alle Aktualisierungen, die Sie an Ihrer Opportunity vornehmen, für andere Partner im Engagement nicht sichtbar, und sie werden weiterhin veraltete Snapshots Ihrer Opportunity-Daten sehen.

 Note

Sie können einen anderen Namen als `ResourceSnapshotJobRole` verwenden. Wenn Sie einen anderen Namen verwenden, ersetzen Sie alle Instanzen von `ResourceSnapshotJobRole` in den folgenden Richtlinien durch Ihren Richtlinienamen.

Um diese Rolle zu erstellen, gehen Sie zu Ihrer IAM-Konsole und erstellen Sie die `ResourceSnapshotJobRole` Rolle mit der folgenden benutzerdefinierten Vertrauensrichtlinie:

JSON

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

Nachdem Sie diese Rolle erstellt haben, müssen Sie die [AWSPartnerCentralSellingResourceSnapshotJobExecutionRolePolicy](#) AWS verwaltete Richtlinie oder die folgende Berechtigungsrichtlinie anhängen:

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/*"
        ]
    }
]
}

```

Ersetzen Sie {account} durch Ihre AWS-Konto-ID.

Nachdem Sie die ResourceSnapshotJobRole Rolle erstellt und die Berechtigungen angehängt haben, müssen Sie sie ResourceSnapshotJobRole für Ihre Organisation festlegen. Verwenden Sie die [PutSellingSystemSettings](#)Aktion, um die Rolle, die Sie erstellt haben, als ResourceSnapshotJobRole für Ihr Unternehmen festzulegen (identifiziert durch Ihr AWS Konto):

```

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} Ersetzen Sie durch Ihre AWS Konto-ID und ResourceSnapshotJobRole durch den Namen der Rolle, die Sie erstellt haben. Diese Rolle wird implizit von den Resource Snapshot-Jobs übernommen, um auf Ihre Opportunities zuzugreifen und Snapshots zu erstellen, die Sie mit anderen Projektpartnern teilen können.

Partner zu einer Geschäftschance einladen

Partner können gemeinsam an Geschäftschancen arbeiten, die sich von beiden Partnern ergeben, und AWS indem sie eine Einladung zum Engagement initiieren. Sie können maximal neun Partner zu einer Opportunity einladen.

Um Partner zu einer Gelegenheit einzuladen

1. Folgen Sie den Schritten unter Partner Central Sales [Guide unter Partner Central AWS Partner finden und mit](#) ihnen in Kontakt treten. Sie müssen sich mit Partnern in Verbindung setzen, bevor Sie sie zu Opportunities einladen können. Diese Verbindung gewährt gegenseitigen Zugriff auf die Konto-IDs des jeweils anderen und stellt so sicher, dass Einladungen das vorgesehene Partnerkonto erreichen.
2. Verwenden Sie die [StartEngagementFromOpportunityTask](#)Aktion, um ein Engagement zu erstellen und diesem eine Verkaufschance zuzuordnen. Diese asynchrone Aktion führt nacheinander die folgenden Aufgaben aus:
 1. Erzeugt ein Engagement.
 2. Verbindet die Gelegenheit mit dem Engagement.
 3. Ruft die [CreateEngagementInvitation](#)Aktion für auf AWS.
 4. Sendet die Gelegenheit an AWS
 5. Startet das Erstellen ResourceSnapshotJob von Schnappschüssen.
3. Laden Sie andere Partner zum Beitritt ein.
 - a. Um die mit Ihrer Opportunity verknüpfte Engagement-ID abzurufen, verwenden Sie die [ListEngagementFromOpportunityTasks](#)Aktion, gefiltert TaskIdentifier nach der im vorherigen Schritt zurückgegebenen.
 - b. Verwenden Sie die Engagement-ID und die Konto-ID des empfangenden Partners, um eine Einladung mit zu sendenCreateEngagementInvitation.

Bei erfolgreicher Einladung wird ein Ereignis, das mit einer Verlobungseinladung erstellt wurde, an den empfangenden Partner gesendet.

Details zur Einladung zur Interaktion werden abgerufen

Wenn Sie eine mit einer Einladung zur Interaktion erstellte Veranstaltung erhalten, rufen Sie mit der [GetEngagementInvitation](#)Aktion die Einladungsdetails ab. Die Antwort enthält wichtige Informationen über die Kundenchance, die mit dem Engagement verbunden ist.

Überprüfen Sie die Einladungsdetails, insbesondere die `InvitationMessage` `Project.CustomerBusinessProblem` Felder `Project.Title` `Project.CustomerUseCase`, und. Diese Informationen bieten einen Überblick über die Kundenchance und die Erwartungen des einladenden Partners an die Zusammenarbeit.

Verwenden Sie diese Informationen, um zu beurteilen, ob Sie die Gelegenheit nutzen möchten, indem Sie die Einladung zur Zusammenarbeit annehmen oder ablehnen.

Auf eine Einladung zur Verlobung antworten

Wenn ein Partner Ihnen eine Einladung zur Interaktion sendet, erstellt er ein Ereignis, das von einer Einladung zur Interaktion erstellt wurde und Sie über die Einladung informiert. Sie haben die Möglichkeit, die Einladung anzunehmen oder abzulehnen. Diese Entscheidung bestimmt Ihre Teilnahme an der Multipartner-Opportunity.

So lehnen Sie eine Einladung zu einem Engagement ab:

Verwenden Sie die [RejectEngagementInvitation](#)Aktion, um die Einladung abzulehnen. Sie müssen einen `RejectionReason` Parameter angeben, der Ihre Entscheidung erklärt. Nach der Ablehnung verlieren Sie den Zugriff auf die Einladungsdetails, und bei einem Ereignis mit abgelehnter Einladung wird der sendende Partner darüber informiert, dass Sie die Einladung abgelehnt haben.

So nehmen Sie eine Einladung zu einem Engagement an:

Verwenden Sie die [StartEngagementByAcceptingInvitationTask](#)Aktion, um die Einladung anzunehmen und mit der Multipartner-Opportunity fortzufahren. Diese asynchrone Aktion führt nacheinander die folgenden Aufgaben aus:

1. Nimmt die Einladung zur Verlobung an.
2. Erstellt eine neue Opportunity in Ihrem Partnerkonto unter Verwendung von Daten aus der Opportunity des sendenden Partners. Sie erhalten ein Ereignis, das von einer Opportunity erstellt wurde.
3. Enthält zusätzliche Details, die zur Identifizierung des Kunden in Ihrem Konto erforderlich sind.

4. Veröffentlicht eine Veranstaltung, bei der eine Einladung zur Teilnahme angenommen wurde, in der der Partner darüber informiert wird, dass Sie die Einladung angenommen haben und dem Engagement hinzugefügt wurden.

Einladung zur Verlobung abgelaufen

Wenn Sie nicht innerhalb von fünfzehn Tagen auf eine Einladung zu einem Engagement antworten, läuft diese ab und Sie können nicht an der Multipartner-Opportunity teilnehmen. Bei einem Ereignis mit abgelaufener Einladung wird der sendende Partner darüber informiert, dass die Einladung abgelaufen ist.

Note

Wenn Sie weiterhin zusammenarbeiten möchten, muss der sendende Partner eine neue Kontakteinladung initiieren.

Prüfung und Aktualisierung von Möglichkeiten für mehrere Partner

Wenn ein Partner ein Engagement für eine Opportunity initiiert, hängt der Überprüfungsstatus der neu erstellten Opportunity im Konto des empfangenden Partners vom Opportunity-Status des sendenden Partners ab.

Verkaufschance mit dem Status „Bewertung eingereicht“:

Wenn die Opportunity des sendenden Partners auf `Lifecycle.ReviewStatus` eingestellt ist `Submitted`, findet der folgende Vorgang statt:

1. Die neue Opportunity, die im Konto des empfangenden Partners erstellt wurde, wird auf `Lifecycle.ReviewStatus` eingestellt sein `Submitted`.
2. Der Submitted Status bleibt so lange bestehen, bis die Opportunity des sendenden Partners lautet `Approved`.
3. Sobald die Opportunity des sendenden Partners validiert und auf gesetzt `Lifecycle.ReviewStatus` ist `Approved`, übernehmen die Opportunities der anderen Partner automatisch den Approved Status.

Durch diesen Prozess wird gewährleistet, dass alle Opportunity-Details für mehrere Partner einheitlich sind, sodass keine mehrfachen Überprüfungen erforderlich sind.

Sobald der Vorgang abgeschlossen ist, wird das Ereignis „Opportunity Created“ mit der entsprechenden Opportunity-ID ausgelöst. Partner können diese ID zusammen mit der `GetOpportunity` Aktion verwenden, um die vollständigen Opportunity-Details abzurufen.

Verkaufschance mit dem Status „Genehmigt“:

Wenn der sendende Partner ein Engagement für eine Opportunity initiiert, bei der die `Lifecycle.ReviewStatus` Einstellung auf `gesetzt ist Approved`, findet der folgende Prozess statt:

1. Die neue Opportunity, die im Konto des empfangenden Partners erstellt wurde, wird auf `Lifecycle.ReviewStatus Approved` eingestellt sein.
2. Ein Ereignis, das durch eine Opportunity erstellt wurde, wird mit der entsprechenden Opportunity-ID ausgelöst.
3. Partner können die [GetOpportunity](#) Aktion mit der Opportunity-ID verwenden, um die vollständigen Opportunity-Details abzurufen.

Die genehmigte Opportunity enthält jedoch möglicherweise keine partnerspezifischen Details, die nicht aus der Opportunity des sendenden Partners kopiert wurden. Der empfangende Partner sollte diese Daten mithilfe der folgenden [UpdateOpportunity](#) Aktion aktualisieren, wenn die Opportunity-Phase auf `Qualified` oder zu einem späteren Zeitpunkt aktualisiert wird:

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

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

Sobald die Opportunity im Konto des empfangenden Partners erstellt wurde, kann sie wie jede andere Opportunity im System des Partners verwaltet und aktualisiert werden. Partner können die

[UpdateOpportunity](#)Aktion nutzen, um Änderungen vorzunehmen oder weitere Informationen über ihre Beteiligung an der Opportunity bereitzustellen.

Verwenden von Snapshots, um Partner-Updates zu erhalten

Im Rahmen eines Engagements pflegen und aktualisieren die Partner ihre Geschäftschancen unabhängig voneinander. Wenn jemand die Ressourcen eines Engagements überarbeitet, z. B. wenn eine Opportunity hinzugefügt wird, wird ein Ereignis veröffentlicht, das mit einer Momentaufnahme der Engagement-Ressource erstellt wurde.

Um über Änderungen auf dem Laufenden zu bleiben und auf die aktuellsten Informationen zu Opportunities anderer Partner zuzugreifen, können Sie die folgenden Aktionen verwenden:

- [ListEngagementResourceAssociations](#)— Verwenden Sie diese Aktion, um die mit der Opportunity verknüpfte Engagement-ID abzurufen.
- [ListResourceSnapshots](#)— Verwenden Sie diese Aktion, um eine umfassende Liste aller mit dem Engagement verknüpften Opportunity-Snapshots abzurufen, die einen Überblick über die verfügbaren Snapshots aller Partner bietet.
- [GetResourceSnapshot](#)— Verwenden Sie diese Aktion, um in Echtzeit Zusammenfassungen bestimmter Opportunity-Schnappschüsse zu erhalten, sodass Sie die aktuellsten Informationen einsehen können, ohne direkt auf die Opportunity eines anderen Partners zugreifen zu müssen.

Wenn Sie relevante Änderungen oder Aktualisierungen aufgrund der Geschäftschance eines anderen Partners feststellen, die sich in Ihrer eigenen widerspiegeln sollten, verwenden Sie die [UpdateOpportunity](#)Aktion. Diese Aktion erleichtert die selektive Einbindung relevanter Daten in Ihre Geschäftschance und gewährleistet so die Abstimmung und Konsistenz des gesamten Engagements.

Engagement-Mitglieder anzeigen

Ein Engagement im Rahmen einer Opportunity mit mehreren Partnern kann bis zu zehn Partner haben. Immer wenn ein neuer Partner die Einladung zum Engagement annimmt, wird eine Veranstaltung veröffentlicht, die ein Mitglied hinzugefügt hat. Dabei werden alle aktuellen Mitglieder über die Änderung informiert.

Gehen Sie wie folgt vor, um alle Mitglieder zu sehen, die im Rahmen eines Engagements zusammenarbeiten:

1. Verwenden Sie die [ListEngagementResourceAssociations](#)Aktion, um die mit der Opportunity verknüpfte Engagement-ID abzurufen.

2. Geben Sie die ID aus Schritt 1 für die [ListEngagementMembers](#)Aktion ein, um die Partnerdetails der Engagement-Mitglieder abzurufen.

 Note

Nur Mitglieder eines Engagements können die `ListEngagementMembers` Aktion aufrufen.

Den Status des Ressourcen-Snapshots überwachen

Wenn Sie mit Opportunities für mehrere Partner arbeiten, müssen Sie eine Rolle einrichten, die es Ihnen ermöglicht, Snapshots Ihrer Opportunities für andere Partner zu erstellen und Schnappschüsse von anderen Partnern in einem Engagement abzurufen. Ohne diese Rolle stehen alle Aktualisierungen, die Sie an Ihrer Opportunity vornehmen, anderen Partnern des Engagements nicht zur Verfügung, und sie werden weiterhin veraltete Schnappschüsse Ihrer Verkaufschance sehen.

Gehen Sie wie folgt vor, um den Status von Ressourcen-Snapshot-Jobs zu verfolgen, die mit einer Opportunity mit mehreren Partnern in einem Engagement verknüpft sind:

1. Verwenden Sie die [ListEngagementResourceAssociations](#)Aktion, um die mit der Opportunity verknüpfte Engagement-ID abzurufen.
2. Geben Sie die ID aus Schritt 1 bis zur [ListResourceSnapshotJobs](#)Aktion ein, um eine Liste aller Snapshot-Jobs zu erstellen, die dem Anrufer im Engagement gehören. Rufen Sie die Job-ID aus der Antwort ab.
3. Geben Sie die Job-ID für die [GetResourceSnapshotJob](#)Aktion ein, um den Auftragsstatus zu verfolgen und festzustellen, ob sie ausgeführt wird.

Der `ResourceSnapshotJob` Vorgang veröffentlicht Kennzahlen CloudWatch für seine asynchronen Operationen auf Amazon. CloudWatch verarbeitet die Daten in lesbare Metriken, die nahezu in Echtzeit verfügbar sind, sodass Sie die Leistung und den Zustand des Services überwachen können.

Die folgenden Metriken sind verfügbar:

- Fehler — Zählt interne Probleme bei der Auftragsausführung. Verwenden Sie diese Metrik, um die Servicestabilität zu überwachen und Alarme für Fehlerschwellenwerte einzurichten.

- Fehler — Verfolgt kundenbezogene Probleme bei der Auftragsverarbeitung. Diese Kennzahl hilft bei der Identifizierung von Problemen mit Kundeneingaben oder Konfigurationen.
- Aufrufe — Stellt die Gesamtzahl der Job-Aufrufe dar, einschließlich erfolgreicher und fehlgeschlagener Versuche. Verwenden Sie diese Option, um Trends bei der Nutzung von Diensten im Laufe der Zeit zu verfolgen.

Note

Metriken werden CloudWatch nur dann gemeldet, wenn der ResourceSnapshotJob Service aktiv ist. Wenn es für eine bestimmte Metrik keine Jobs oder keine Daten gibt, wird diese Metrik nicht gemeldet.

Um einen reibungslosen Ablauf des ResourceSnapshotJob Vorgangs zu gewährleisten:

- Verwenden Sie die Metriken, um zu überprüfen, ob der Service wie erwartet funktioniert.
- Erstellen Sie CloudWatch Alarme, um Messwerte zu überwachen und Aktionen auszulösen (z. B. das Senden von Benachrichtigungen), wenn die Messwerte akzeptable Bereiche überschreiten.
- Richten Sie eine proaktive Überwachung ein, um Probleme zu identifizieren und zu lösen.

Weitere Informationen zur Verwendung von CloudWatch Metriken finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

Möglichkeiten zuordnen, trennen und zuweisen

Opportunities können in jeder Phase des Opportunity-Lebenszyklus mit Partnerlösungen, AWS Produkten, AWS Marketplace Lösungen, AWS Marketplace Produkten, AWS Marketplace Angeboten und AWS Marketplace Angebotspaketen verknüpft oder getrennt werden.

Verknüpfung von Opportunities mit anderen Entitäten

Die zugehörigen Entitäten werden aus der GetOpportunity Methode innerhalb des RelatedEntityIdentifiers Objekts abgerufen. Die RelatedEntityIdentifiers können aktualisiert werden mit AssociateOpportunity. Beachten Sie, dass dieses Feld nicht mit der CreateOpportunity Methode UpdateOpportunity oder aktualisiert werden kann.

Lösungen

Bevor eine Zusammenarbeit mit der `StartEngagementFromOpportunityTask` Aktion gestartet wird, müssen mindestens eine und bis zu zehn Partnerlösungen verknüpft werden. Eine AWS Empfehlung kann eine Partnerlösung beinhalten oder auch nicht.

Verwenden Sie die `ListSolutions` Aktion, um sich Ihre bestehenden Lösungen anzusehen.

Partner können ihre Lösungen im Build Abschnitt in [AWS Partner Central](#) erstellen, aktualisieren und verwalten.

Important

Um aus der Opportunity-Phase „Zugesagt“ oder „Markteinführung“ zu werden, müssen Sie der Verkaufschance eine gültige Lösung oder ein gültiges AWS Marketplace Produktangebot zuordnen.

Note

Ältere Partnerlösungen (S-XXXXIdentifikatoren), die den `Solutions` Entitätstyp verwenden, werden weiterhin unterstützt. Wir empfehlen jedoch, den Opportunities den `AwsMarketplaceSolutions` Entitätstyp zuzuordnen, da der `Solution` Entitätstyp in future veraltet sein wird.

AWS Produkte

Bis zu 20 AWS Produkte können mit einer Verkaufschance verknüpft werden. Um eine Liste der verfügbaren AWS Produkte einzusehen, verwenden Sie die Liste der [AWS Produkte, auf denen gehostet wird GitHub](#). Die Zuordnung zu AWS Produkten erfolgt ausschließlich über die `AssociateOpportunity` Aktion. Verwenden Sie die `DisassociateOpportunity` Aktion, um ein Produkt zu ersetzen oder zu entfernen.

Lösungen

Opportunities können mit Lösungen aus Ihrem Verkäuferkonto verknüpft werden. Verwenden Sie die `AssociateOpportunity` Aktion mit der `RelatedEntityType` Einstellung `aufAwsMarketplaceSolutions`.

Die Entität muss den Status „Öffentlich“ oder „Eingeschränkt“ haben. Der Dienst lehnt Entwürfe oder inaktive Entitäten ab.

Für die Zuordnung einer Lösung ist ein ARN erforderlich. Das folgende Beispiel zeigt das ARN-Format für eine Lösung:

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

Sie können auch Lösungen von einem AWS Marketplace Tochterkonto verknüpfen, das über Partner Account Connection mit Ihrem Hauptkonto verbunden ist. Geben Sie den vollständigen ARN ein, einschließlich der Konto-ID der Tochtergesellschaft.

Die `ListSolutions` Aktion wird `AwsMarketplaceSolutionArn` in jeder Lösungszusammenfassung angezeigt und unterstützt das Filtern nach ARN.

AWS Marketplace Produkte

Verkaufschancen können mit AWS Marketplace Produkten aus Ihrem Verkäuferkonto verknüpft werden. Verwenden Sie die `AssociateOpportunity` Aktion mit der `RelatedEntityType` Einstellung auf `AwsMarketplaceProducts`.

Die Entität muss den Status „Öffentlich“ oder „Eingeschränkt“ haben. Der Dienst lehnt Entwürfe oder inaktive Entitäten ab.

Für die Zuordnung eines Produkts ist ein ARN erforderlich. Der ARN beinhaltet den Produkttyp (z. B. `AmiProduct`, `SaaSProduct`, `ContainerProduct`). Das folgende Beispiel zeigt das ARN-Format für ein AMI-Produkt:

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

Sie können auch Produkte von einem AWS Marketplace Tochterkonto verknüpfen, das über Partner Account Connection mit Ihrem Hauptkonto verbunden ist. Geben Sie den vollständigen ARN ein, einschließlich der Konto-ID der Tochtergesellschaft.

AWS Marketplace Angebote und Angebotssets

AWS Marketplace Angebote

Gelegenheiten können mit einem AWS Marketplace privaten Angebot verknüpft werden. Verwenden Sie die `API ListEntities` aus dem AWS Marketplace Katalog, um sich die verfügbaren Angebote

anzusehen. Derzeit können Sie nur Angebote aus dem AWS Marketplace Verkäuferkonto verknüpfen, das mit AWS Partner Central verknüpft ist.

Für die Verknüpfung eines privaten Angebots ist ein ARN erforderlich. Beispiel:

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

AWS Marketplace Sets anbieten

Opportunities können auch mit AWS Marketplace Angebotspaketen verknüpft werden. Angebotspakete ermöglichen die Gruppierung mehrerer verwandter Marktangebote zu einem umfassenden Lösungspaket. Um verfügbare Angebotssets anzuzeigen, verwenden Sie die ListEntities aus der AWS Marketplace Katalog-API.

Für die Zuordnung eines Angebotssets ist ein ARN erforderlich. Beispiel:

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

Wichtig: Eine Opportunity kann entweder einem Angebot ODER einem Angebotspaket zugeordnet werden, jedoch nicht mit beiden. Einer Opportunity kann jeweils nur einer dieser Entitätstypen zugeordnet werden.

Trennung von Opportunities und anderen Entitäten

Verwenden Sie die DisassociateOpportunity Aktion, um die Verknüpfung der Opportunity mit den folgenden Entitätstypen aufzuheben:

- Lösungen
- AWS Produkte
- AWS Marketplace Lösungen
- AWS Marketplace Produkte
- AWS Marketplace Angebote
- AWS Marketplace Bieten Sie Sets an

Je nach Status der Verkaufschance gelten unterschiedliche Validierungsregeln, wenn Sie die Verknüpfung dieser verwandten Objekte aufheben.

Zuweisen von Opportunities

Wird verwendet `AssignOpportunity`, um den Besitzer der Opportunity zu ändern. Sie können jeden Ihrer Partner Central-Benutzer zum Eigentümer der Opportunity machen.

Arbeiten Sie mit Ihren Leads

Was ist ein Lead?

Im B2B-Vertrieb (B2B) steht ein Lead für einen potenziellen Kunden, der Interesse an den Produkten oder Dienstleistungen eines Unternehmens bekundet hat, aber noch nicht in vollem Umfang als Verkaufschance qualifiziert wurde. Leads sind die Anfangsphase der Vertriebspipeline und müssen gepflegt und qualifiziert werden, bevor sie in Möglichkeiten für einen aktiven Co-Selling umgewandelt werden können. AWS

Leads, die an Partner AWS weitergegeben werden, basieren auf Kundenbindungssignalen wie der Teilnahme an Webinaren, der Teilnahme an Kampagnen oder Anfragen zu Partnerlösungen. Diese Leads enthalten wertvollen Kontext wie Unternehmensinformationen zu Kunden, Geschäftsproblemen und Details zur Kundenbindung, sodass Partner potenzielle Geschäftschancen priorisieren und qualifizieren können.

Mit Einladungen für Leads arbeiten

Partner erhalten Lead-Einladungen AWS, wenn potenzielle Kunden Interesse an Partnerlösungen oder -dienstleistungen bekunden. Das Einladungsverfahren ermöglicht es Partnern, Leads zu bewerten, bevor sie sich zu einem Engagement verpflichten, wodurch eine effiziente Ressourcenzuweisung und höhere Konversionsraten gewährleistet werden.

Empfang von Lead-Einladungen

AWS erstellt Lead-Einladungen und gibt sie über die Verkaufs-API an Partner weiter. Wenn eine neue Lead-Einladung verfügbar ist, AWS sendet eine Engagement-Einladung mit dem Lead-Kontext an berechnigte Partner.

Partner sollten die Engagement Invitation Created Veranstaltung über Amazon verfolgen EventBridge. Diese Veranstaltungsbenachrichtigung enthält das `payloadType` Feld, das auf gesetzt ist `LeadInvitation`, sodass Partner zwischen Einladungen für Leads und Einladungen zu Geschäftschancen unterscheiden können. Nach Erhalt der Veranstaltung können Partner die Einladungsdetails abrufen, um den Lead zu bewerten.

Einladungen für Leads auflisten

Partner können alle ihre Lead-Einladungen mithilfe der `ListEngagementInvitations` API-Aktion einsehen. Diese Aktion ruft Einladungen ab, bei denen der aufrufende Principal entweder ein Absender oder ein Empfänger ist.

Um speziell nach Lead-Einladungen zu filtern, sollten Partner den `payloadType` Filter mit dem Wert `LeadInvitation` verwenden. Dieser Filter stellt sicher, dass nur Lead-Einladungen zurückgegeben werden, wobei Opportunity-Einladungen nicht in den Ergebnissen enthalten sind.

Einladungen für Leads können sich in den folgenden Zuständen befinden:

- **Ausstehend:** Wartet auf Annahme oder Ablehnung durch den Partner
- **Angenommen:** Der Partner hat die Einladung angenommen und Zugriff auf alle Lead-Details erhalten
- **Abgelehnt:** Der Partner hat die Einladung abgelehnt
- **Abgelaufen:** Die Einladung hat ihr Ablaufdatum ohne Aktion überschritten

Bewertung von Einladungen für Leads

Bevor Partner eine Lead-Einladung annehmen, sollten sie mithilfe der `GetEngagementInvitation` API-Aktion detaillierte Einladungsinformationen abrufen. Dies bietet einen wichtigen Kontext, um fundierte Entscheidungen über Annahme oder Ablehnung zu treffen.

Die Payload für Lead-Einladungen umfasst:

Kundeninformationen (vor der Annahme verfügbar):

- Firmenname, Branche und Land
- Website-URL
- Marktsegment (Unternehmen, Groß, Mittel, Klein, Kleinstunternehmen)
- AWS Reifegrad (Bewertung, Single-Account, Multi-Account)

Interaktionen mit Kunden:

- Informationen zur Quelle und zur Kampagne
- Maßnahmen des Kunden (Ausfüllen des Formulars, Teilnahme am Webinar usw.)

- Vom Kunden bereitgestellte Beschreibung des Geschäftsproblems
- Kategorie Anwendungsfall
- Titel der Kontaktperson (stellt den behördlichen Kontext für die BANT-Qualifikation bereit)

 Important

Die Kontaktinformationen des Kunden (Name, E-Mail-Adresse, Telefonnummer) sind in der Einladungsphase nicht sichtbar. Die Kontaktdaten werden erst verfügbar, nachdem Sie die Einladung angenommen haben, wodurch der Datenschutz der Kunden gewährleistet und das Verhalten von Lead-Farmern verhindert wird.

Annahme von Lead-Einladungen

Wenn ein Partner beschließt, einem Lead nachzugehen, muss er die Einladung mithilfe der `AcceptEngagementInvitation` API-Aktion annehmen. Diese Aktion fügt den Partner dem Engagement hinzu und gewährt Zugriff auf die vollständigen Lead-Details, einschließlich der Kontaktinformationen der Kunden.

Nach der Annahme:

- Der Partner wird als Mitglied des Engagements hinzugefügt
- Die Kontaktinformationen des Kunden werden im Kontext des Engagement-Leads sichtbar
- Der Lead wird in Systemen als partnerqualifizierter Lead (PQL) eingestuft AWS
- Der Lead erscheint in der Liste der aktiven Leads des Partners
- Partner können damit beginnen, den Lead zu pflegen und Kundenkontakt herzustellen

Partner können mehrere Lead-Einladungen programmatisch annehmen, indem sie `AcceptEngagementInvitation` für jede Einladung aufrufen, was eine effiziente Massenverarbeitung hochwertiger Leads ermöglicht.

Lead-Einladungen ablehnen

Wenn ein Lead nicht den Fähigkeiten, Kapazitäten oder Geschäftsschwerpunkten des Partners entspricht, können Partner die Einladung mithilfe der `RejectEngagementInvitation` API-Aktion ablehnen. Bei der Ablehnung einer Lead-Einladung sollten Partner einen Ablehnungsgrund

angeben, um die Weiterleitung und den Abgleich von Leads zu AWS verbessern. Zu den häufigsten Ablehnungsgründen gehören:

- Mangelnde geografische Abdeckung
- Unzureichendes technisches Fachwissen für den Anwendungsfall
- Derzeitige Kapazitätsengpässe
- Diskrepanz zwischen Kundenbranche und Branche
- Fehlausrichtung des Budgets

Nach der Ablehnung:

- Der Lead wird aus der Warteschlange für ausstehende Einladungen des Partners entfernt
- Kontaktinformationen des Kunden werden niemals an den Partner weitergegeben
- Der Lead wird in Systemen als Partner Rejected Lead (PRL) eingestuft AWS
- Die Einladung bleibt als Referenz in der Liste der abgelehnten Einladungen des Partners sichtbar

Abgelehnte Leads können möglicherweise anderen Partnern zugewiesen werden, sofern die Zustimmung des Kunden erfüllt ist.

Verwaltung akzeptierter Leads

Nach der Annahme einer Lead-Einladung können Partner auf vollständige Lead-Informationen zugreifen, Kundenbeziehungen pflegen und den Lead durch die Qualifizierungsphasen verfolgen.

Lead-Details anzeigen

Partner greifen über die `GetEngagement` API-Aktion auf vollständige Lead-Details zu. Das Engagement enthält einen Lead-Kontext mit umfassenden Informationen über den Kunden und seine Interaktionen mit AWS.

Der Lead-Kontext umfasst:

Vollständiges Kundenprofil:

- Alle Unternehmensinformationen aus der ursprünglichen Einladung
- Vollständige Kontaktdaten für alle Kundeninteraktionen (Name, E-Mail, Telefon, Geschäftstitel)
- AWS Reifegrad und Marktsegment

Verlauf der Interaktion:

- Mehrere Kontaktpunkte mit Kunden wurden zu einem einzigen Kundenkontakt zusammengefasst
- Jede Interaktion umfasst Quellinformationen, Kundenaktionen, Geschäftsprobleme und Kontaktinformationen
- Die Interaktionen werden chronologisch aufgelistet, um einen vollständigen Überblick über die Kundenreise zu erhalten

Qualifikationsstatus:

- Aktueller Status der Lead-Qualifikation (Status „Unqualifiziert“, „Qualifiziert“, „Disqualifiziert“ oder „Benutzerdefiniert“)
- Partner können diesen Status aktualisieren, während sie ihren Qualifizierungsprozess durchlaufen

Diese umfassende Ansicht ermöglicht es Partnern, die gesamte Kundeninteraktionsgeschichte über mehrere AWS Kampagnen und Kontaktpunkte hinweg nachzuvollziehen, anstatt jede Interaktion als separaten Lead zu behandeln.

Aktive Leads auflisten

Partner können alle ihre aktiven Leads mithilfe der `ListEngagements` API-Aktion abrufen, bei der der `contextType` Filter auf `Lead` eingestellt ist. Dadurch werden Interaktionen zurückgegeben, bei denen der Partner Mitglied ist und die Interaktion einen Lead-Kontext enthält.

Die Antwort auf die Liste enthält wichtige Informationen wie:

- ID und Titel des Engagements
- Firmenname des Kunden
- Aktueller Engagement-Score
- Qualifizierungsstatus
- Anzahl der Interaktionen
- Zeitstempel für Erstellung und Änderung

Partner können diese Liste verwenden, um Dashboards zu erstellen, den Zustand der Lead-Pipeline zu verfolgen und Leads zu identifizieren, die Aufmerksamkeit erfordern.

Anreicherung von Leads mit Erkenntnissen zur Kundenakquise

Partner können akzeptierte Leads mit AWS Erkenntnissen anreichern, um sie besser zu priorisieren und zu qualifizieren, bevor sie in Öffentlichkeitsarbeit investieren. Die Anreicherung erfolgt asynchron durch Akquise-Aufgaben, die das Engagement von Leads um abgeleitete Signale zur Unterstützung von Qualifizierungsentscheidungen ergänzen. AWS

Eine Aufgabe zur Kundenakquise starten

Partner initiieren die Anreicherung mithilfe der `StartProspectingFromEngagementTask` API-Aktion. Diese Aktion akzeptiert bis zu 100 Engagement-Identifikatoren in einer einzigen Anfrage, sodass Partner Leads stapelweise anreichern können. Die Aufgabe wird asynchron ausgeführt und gibt sofort ein `TaskId` und `taskArn` zurück, das Partner verwenden, um den Fortschritt zu verfolgen. Die initiale `TaskStatus` ist eine von `PENDING`, `IN_PROGRESS`, `COMPLETED`, oder `FAILED`.

Partner können optional `a` angeben, um die Aufgabe `TaskName` zu kennzeichnen, und `aClientToken`, um Idempotenz bei Wiederholungsversuchen sicherzustellen.

Umfrage nach Ergebnissen

Da die Kundenakquise asynchron erfolgt, fragen die Partner mithilfe der `GetProspectingFromEngagementTask` API-Aktion nach Abschluss ab, wobei beim Start das Ergebnis `TaskId` zurückgegeben wird. In der Antwort werden der Gesamtstatus der Aufgabe und ein Ergebnis pro Engagement für jede übermittelte Kennung angegeben.

Jeder Auftrag wird unabhängig bearbeitet, sodass einzelne Engagements erfolgreich sein oder scheitern können, ohne dass sich dies auf die anderen an derselben Aufgabe beteiligten Personen auswirkt. Für jedes Engagement umfasst das Ergebnis:

- Interaktions-ID: Die Interaktion, die verarbeitet wurde.
- Status: `PENDING`, `IN_PROGRESS`, `COMPLETED`, oder `FAILED`.
- Kontext-ID für die Kundenakquise: Wird eingegeben, wenn das Engagement erfolgreich bearbeitet wurde. Diese Kennung bezieht sich auf den erweiterten Kontext der Kundenakquise, der dem Engagement hinzugefügt wurde, und kann in nachfolgenden Vorgängen verwendet werden.
- Ursachencode und Meldung: Wird nur angezeigt, wenn ein Projekt fehlschlägt, und enthält einen aufgezählten Fehlercode und eine für Menschen lesbare Beschreibung mit empfohlenen Wiederherstellungsschritten.

Überwachung mehrerer Aufgaben

Um die Anreicherung vieler Leads nachzuverfolgen, verwenden Partner die `ListProspectingFromEngagementTasks` API-Aktion. Diese Aktion listet alle vom Konto des Anrufers initiierten Aufgaben zur Kundenakquise auf und unterstützt optionale Filter nach Aufgaben-ID, Aufgabenname oder Startzeitbereich mit konfigurierbarer Sortierung. Die Antwort ist paginiert. Verwenden Sie den `NextToken` Wert jeder Antwort, um nachfolgende Seiten abzurufen.

Wenn die Anreicherung erfolgreich abgeschlossen wurde, werden die AWS Erkenntnisse dem Engagement als Kontext für die Kundenakquise hinzugefügt. Partner können die angereicherten Details abrufen `GetEngagement` und sie verwenden, um den Qualifikationsstatus des Leads festzulegen und zu entscheiden, welche Leads zur Konversion übergehen sollen.

Aktualisierung der Lead-Informationen

Wenn Partner Leads pflegen und zusätzliche Informationen sammeln, können sie die Lead-Details mithilfe der `UpdateEngagementContext` API-Aktion aktualisieren. Diese Aktion ermöglicht es Partnern, den Lead-Kontext innerhalb des Engagements zu ändern.

Partner können Folgendes aktualisieren:

Qualifikationsstatus: Partner sollten den Qualifikationsstatus aktualisieren, während sie ihren internen Qualifizierungsprozess durchlaufen:

- **Unqualifiziert:** Standardstatus nach Annahme eines Leads; gibt an, dass der Lead einer Bewertung bedarf
- **Qualifiziert:** Der Lead erfüllt die Qualifikationskriterien und weist ein hohes Potenzial zur Umwandlung in eine Verkaufschance auf
- **Disqualifiziert:** Lead erfüllt die Kriterien nicht oder passt nicht zu den Lösungen des Partners
- **Benutzerdefinierter Status:** Partner können ihren eigenen Qualifikationsstatus definieren, der ihren internen Prozessen entspricht

Lead-Metadaten: Partner können zusätzliche Informationen hinzufügen oder aktualisieren, die für ihre Qualifizierungs- und Förderprozesse relevant sind.

Die Aktualisierung des Qualifikationsstatus hilft Partnern dabei, die Lead-Entwicklung zu verfolgen, genaue Pipeline-Berichte zu erstellen und Leads zu identifizieren, die für die Umwandlung in Geschäftschancen bereit sind.

Überwachen AWS Aktualisierungen

AWS kann Lead-Informationen auf der Grundlage neuer Kundenbindungssignale oder einer verfeinerten Bewertung der Kundenbindung aktualisieren. Wenn das Engagement AWS aktualisiert wird, erhalten Partner eine Engagement Updated Veranstaltung über Amazon EventBridge.

Diese Updates können Folgendes beinhalten:

- Verfeinerte Interaktionswerte auf der Grundlage neuer Kundenaktivitäten
- Zusätzliche Kundeninteraktionen durch neue Kampagnen oder Kontaktpunkte
- Aktualisierte Kundeninformationen

Partner sollten diese Ereignisse im Auge behalten und GetEngagement sich telefonisch über die neuesten Lead-Details informieren. Dadurch wird sichergestellt, dass die Partner über die aktuellsten Informationen zur Priorisierung und Pflege von Leads verfügen.

Die Engagement Updated Veranstaltung umfasst das contextTypes Feld, sodass Partner gezielt nach Leads filtern können (Interaktionen mit Lead-Kontext).

Umwandlung eines Leads in eine Verkaufschance

Wenn ein Lead qualifiziert wurde und eine ernsthafte Kaufabsicht zeigt, können Partner dies in eine Gelegenheit für eine formelle Co-Selling-Zusammenarbeit umwandeln. AWS Diese Umstellung markiert den Übergang von der Leadpflege (in erster Linie durch Partner) zum Chancenmanagement (in Zusammenarbeit mit). AWS

Empfohlener Ansatz: Convenience API

Partner können den Prozess zur Erstellung und Verknüpfung von Opportunities mithilfe der StartOpportunityFromEngagementTask praktischen API-Aktion optimieren. Diese Task-API orchestriert mehrere Aktionen automatisch:

1. Erstellt einen Opportunity-Entwurf mit vorläufigen Informationen aus dem Lead-Kontext
2. Verbindet die Opportunity über einen Ressourcen-Snapshot mit dem ursprünglichen Engagement
3. Fügt dem Engagement den CustomerProject Kontext hinzu

Diese praktische Methode reduziert die Anzahl der erforderlichen API-Aufrufe und gewährleistet eine korrekte Nachverfolgung der Zuordnung. Partner, die diesen Ansatz verwenden, müssen dennoch:

- Vollständige Angaben zur Geschäftschance unter `UpdateOpportunity`
- Associate Partner Solutions verwenden `AssociateOpportunity`
- Reichen Sie die Verkaufschance ein, `StartEngagementFromOpportunityTask` wenn Sie bereit sind

Die praktische API ist besonders nützlich für Partner mit automatisierten Lead-to-Popportunity-Workflows, die die Komplexität der Integration minimieren möchten.

Alternativer Ansatz: API Step-by-step

Einen Opportunity-Entwurf erstellen

Der erste Schritt bei der Konvertierung eines Leads besteht darin, mithilfe der `CreateOpportunity` API-Aktion einen Opportunity-Entwurf zu erstellen. Dadurch entsteht eine Opportunity mit der `Lifecycle.ReviewStatus` Einstellung auf `Pending Submission`. Zu diesem Zeitpunkt wurde die Opportunity noch nicht AWS zur Validierung eingereicht.

Die Partner sollten die Gelegenheit mit Informationen füllen, die sie im Rahmen der Lead-Nurturing gesammelt haben, darunter:

- Angaben zum Kundenkonto
- Projektinformationen und Geschäftsproblem
- Erwartete Kundenausgaben
- Zielabschlussdatum
- Alle anderen relevanten Daten, die während der Qualifikation gesammelt wurden

Der Entwurf ermöglicht es den Partnern, vollständige Informationen vorzubereiten, bevor sie sie einreichen AWS, wodurch höhere Genehmigungsraten und eine schnellere Validierung gewährleistet werden.

Verknüpfung von Geschäftschance und Kundenbindung

Nachdem der Opportunity-Entwurf erstellt wurde, müssen die Partner ihn mithilfe der `CreateResourceSnapshot` API-Aktion mit dem ursprünglichen Lead-Engagement verknüpfen. Dieser Schritt ist entscheidend für:

- **Attribution-Tracking:** Legt die Herkunftskette von der Lead-Einladung bis zum Abschluss von Geschäftschancen fest und ermöglicht so eine genaue Berechnung des ROI für Marketingkampagnen und Lead-Quellen.
- **Konversionsmetriken:** Ermöglicht es AWS und Partnern, die Konversionsraten von Leads zu Geschäftschancen zu messen und erfolgreiche Lead-Quellen zu identifizieren.
- **Kontexterhaltung:** Behält den gesamten Verlauf der Kundenreise bei, einschließlich aller ursprünglichen Interaktionen und Interaktionssignale.

Bei der Erstellung des Ressourcen-Snapshots geben die Partner Folgendes an:

- Die Engagement-ID (Quelle: Lead-Engagement)
- Die Opportunity-ID
- Der Ressourcentyp (Opportunity)

Diese Aktion fügt dem Engagement automatisch einen CustomerProject Kontext hinzu und signalisiert, dass der Lead in eine aktive Opportunity umgewandelt wurde. Das Engagement enthält jetzt sowohl den ursprünglichen Lead-Kontext (unter Beibehaltung der Historie) als auch den neuen CustomerProject Kontext (was auf eine aktive Opportunity hinweist).

Details zur Verkaufschance vervollständigen

Sobald die Opportunity erstellt und verlinkt ist, müssen Partner vor der Einreichung zusätzliche Opportunity-Anforderungen erfüllen. Einzelheiten finden Sie in der AssociateOpportunity API-Aktion.

Aktualisierung der Projektdetails: Partner können die UpdateOpportunity API-Aktion verwenden, um Projektinformationen, Geschäftsprobleme von Kunden, erwartete Ausgaben und andere relevante Details, die während der Lead-Qualifizierung gesammelt wurden, zu verfeinern.

Einreichung der Geschäftschance

Sobald alle erforderlichen Informationen vollständig sind, reichen die Partner die Möglichkeit zur AWS Validierung mithilfe der StartEngagementFromOpportunityTask praktischen API ein. Dadurch wird die Opportunity vom Status des Entwurfs in den AWSÜberprüfungsprozess überführt.

Validierungsanforderung: Das Engagement muss vor der Einreichung einen CustomerProject Kontext haben. Dieser Kontext wird automatisch erstellt, wenn er verwendet

wird `CreateResourceSnapshot`, um die Opportunity mit dem Engagement zu verknüpfen. Fehlt dieser Kontext, schlägt die Einreichung fehl.

Nach der Einreichung:

- Die Gelegenheit `Lifecycle.ReviewStatus` ändert sich zu `Submitted`
- Der Lead wird in Systemen als qualifizierter Partner Sales Lead (PSQL) eingestuft AWS
- AWS beginnt mit der Validierung, um sicherzustellen, dass die Angaben zur Verkaufschance korrekt und vollständig sind
- Es können keine Änderungen an der Opportunity vorgenommen werden, bis der Überprüfungsprozess abgeschlossen ist

Die Opportunity folgt dann dem standardmäßigen Validierungsablauf, der in der Dokumentation „Arbeiten mit Ihren Opportunities“ beschrieben ist, und durchläuft Status wie „Maßnahme erforderlich“, „In-Review“, „Genehmigt“ oder „Disqualifiziert“.

Arbeiten mit Erkenntnissen zum Geschäftsumfang

Was ist Deal Sizing?

Deal Sizing ist eine Funktion innerhalb der APN Customer Engagements (ACE) Opportunities in AWS Partner Central, die KI nutzt, um automatisierte Schätzungen der Geschäftsgröße und AWS Serviceempfehlungen bereitzustellen, wenn Partner Opportunities erstellen oder aktualisieren.

KI prognostizierte MRR-Schätzungen

Einblicke in die Geschäftsgröße liefern den Median der prognostizierten Bandbreite der prognostizierten monatlichen wiederkehrenden Umsätze (MRR) auf der Grundlage der bisherigen AWS Geschäftschancen, Anwendungsfälle und Geschäftsbeschreibungen der Partner. Partner sollten die gesamte MRR überprüfen und aktualisieren, sobald die Partner mehr Informationen über das Geschäft und den Verlauf des Verkaufszyklus sammeln.

AI-Recommended AWS Produkte

AI-recommended Produkte werden auf der Grundlage von Problembeschreibungen des Kunden und Einzelheiten zu Geschäftschancen identifiziert. Die KI empfiehlt relevante AWS Produkte, die auf die technischen Anforderungen und Anwendungsfälle abgestimmt sind. Partner sollten diese Vorschläge überprüfen und die Produktauswahl an die spezifischen Bedürfnisse des Kunden anpassen.

Verbesserte Einblicke mit Preisrechner-URLs

Partner können auch AWS Preisrechner-URLs importieren, um Serviceauswahlen automatisch auszufüllen und erweiterte Einblicke zu erhalten, darunter Zulassungsindikatoren des Migration Acceleration Program (MAP), Optimierungsempfehlungen und Analysen potenzieller Kosteneinsparungen.

Daten verstehen Source-Separated

Deal Sizing bietet Einblicke aus zwei verschiedenen Quellen, um Ihnen einen umfassenden Einblick in den Wert von Geschäftschancen zu geben:

- AWS Quelle: AI-recommended Produkte, die auf Mustern basieren, die sich aus ähnlichen Geschäftschancen in der Vergangenheit ergeben
- Partnerquelle: Ihre eigenen Schätzungen wurden aus dem AWS Preisrechner importiert

Umgang mit Abweichungen zwischen Quellen

Wenn die vom Partner bereitgestellten Schätzungen erheblich von den AWS Empfehlungen abweichen, sollten die Partner:

1. Überprüfen Sie die Einzelheiten der Geschäftschance, um sicherzustellen, dass alle relevanten Informationen erfasst wurden
2. Stellen Sie sicher, dass die URL-Konfigurationen des Preisrechners den tatsächlichen Anforderungen entsprechen
3. Betrachten Sie AWS Empfehlungen als Datenpunkte, die auf ähnlichen historischen Geschäftschancen basieren
4. Treffen Sie fundierte Entscheidungen auf der Grundlage des spezifischen Kundenkontextes und der spezifischen Anforderungen

Zugriff auf Einblicke in die Geschäftsgröße

Partner greifen über die `GetAwsOpportunitySummary` API-Aktion auf Deal-Sizing-Daten zu. Dabei wird ein erweitertes `AwsOpportunitySummary` Objekt zurückgegeben, das Prognosen zur Geschäftsgröße, Produktempfehlungen und Optimierungsinformationen enthält.

Wenn Daten zur Angebotsgröße verfügbar werden

Erkenntnisse zur Geschäftsgröße werden verfügbar, sobald eine Verkaufschance mit ausreichenden Kunden- und Projektdetails erstellt wurde. Das System analysiert die Chancenattribute und generiert:

1. Von KI prognostizierte MRR: Wird automatisch auf der Grundlage von Chancenmerkmalen berechnet
2. AWS Produktempfehlungen: Dienstleistungen, die für das Geschäftsproblem und die technischen Anforderungen des Kunden relevant sind
3. Bessere Einblicke (wenn die URL des Preisrechners angegeben wird): MAP-Eignung, Optimierungsempfehlungen und Analyse der Kosteneinsparungen

Note

Einblicke in die Geschäftsgröße sind nicht für alle Opportunities verfügbar. Die Eignung hängt von Faktoren wie der Partnerhistorie und der Vollständigkeit der bereitgestellten Opportunity-Details ab.

Bereitstellung von Eingaben zur Geschäftsgröße

Partner stellen über die `CreateOpportunity` und `UpdateOpportunity` API-Aktionen Eingaben für Deal Sizing bereit. Das System verwendet Opportunity-Attribute, um Empfehlungen zu generieren.

Bereitstellung des Gesamtauftragswerts (TCV)

Partner, die den Gesamtvertragswert und nicht den monatlichen wiederkehrenden Umsatz schätzen, können ihren Geschäftswert direkt angeben. AWS konvertiert TCV mithilfe von AI-powered Konvertierungsmodellen automatisch in AWS MRR.

Für die Größe von Deals in TCV geben Sie `ExpectedContractDuration as Months` (gültige Werte liegen zwischen 1 und 144 Monaten) und `TargetCompany as` und `as Self` an. `Frequency` `None`

```
{
  "Project": {
    "ExpectedContractDuration": { "Term": "Months", "Value": "36" },
    "ExpectedCustomerSpend": [
```

```
{
  "Amount": "1200000.00",
  "CurrencyCode": "USD",
  "Frequency": "None",
  "TargetCompany": "Self"
}
]
```

AWS leitet die AWS-MRR-Schätzung ab und gibt beide Einträge zurück auf: 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"
      }
    ]
  }
}
```

Im obigen Beispiel enthält das ExpectedCustomerSpend erste Array die AWS-MRR-Schätzung (TargetCompany: "AWS",Frequency: "Monthly") und das zweite Array ist der Gesamtvertragswert des Partners (TargetCompany: "Self",Frequency: "None").

TCV-Validierungsregeln

Alle TCV-Validierungsfehler geben den Fehlercode INVALID_VALUE mit zurück. Reason: "BUSINESS_VALIDATION_FAILED" Der Unterscheidungsfaktor ist das fieldName und: Message

Bedingung	Feld	Fehlermeldung
Selbsteinstieg ohne ExpectedContractDuration	<code>project.expectedContractDuration</code>	ExpectedContractDuration ist erforderlich, wenn ein Eigeneintrag vorhanden ist
ExpectedContractDuration ohne Selbsteintrag (nur Create)	<code>project.expectedCustomerSpend</code>	Eine Selbsteingabe ist erforderlich, wenn sie vorhanden ExpectedContractDuration ist
Mehrere Self-Einträge	<code>project.expectedCustomerSpend</code>	Nur ein Self-Eintrag ist zulässig
Ungültig TargetCompany , wenn eine Dauer vorhanden ist	<code>project.expectedCustomerSpend.targetCompany</code>	TargetCompany muss 'AWS' oder 'Self' sein
Währungsinkongruenz zwischen AWS und Self	<code>project.expectedCustomerSpend.currencyCode</code>	CurrencyCode muss zwischen den Einträgen übereinstimmen
Selbsteingabe + EstimationUrl zusammen	<code>project.expectedCustomerSpend.estimationUrl</code>	Selbsteintritt und EstimationUrl kann nicht koexistieren

Note

Wenn ein Eintrag ohne Eigeneintrag vorhanden ExpectedContractDuration istUpdateOpportunity, der Wert des Partnergeschäfts jedoch im Speicher vorhanden ist, rekonstruiert das System den Eigeneintrag automatisch — es wird kein Fehler ausgegeben.

Umschalten zwischen TCV und manueller MRR

Um vom TCV-Modus zurück zum manuellen MRR zu wechseln, setzen Sie die Einstellung explizit ExpectedContractDuration auf null und geben Sie nur einen Eintrag ein: AWS

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

Note

Das Auslassen `ExpectedContractDuration` aus der Nutzlast (ohne das Feld zu senden) bedeutet „keine Änderung“ — die vorhandenen TCV-Daten bleiben erhalten. Explizites Senden `null` bedeutet „TCV-Modus löschen“.

URLs für TCV und Preisrechner

Wenn ein Partner sowohl einen `Self` Eintrag (TCV-Modus) als auch einen `angibtEstimationUrl`, gibt die API einen Validierungsfehler zurück. Selbsteingabe und `EstimationUrl` kann nicht gleichzeitig in derselben Anfrage existieren.

Bereitstellung manueller MRR-Schätzungen

Partner können MRR-Schätzungen mithilfe des `Project.ExpectedCustomerSpend[].Amount` folgenden Felds manuell eingeben:

Note

Das `CurrencyCode` Feld akzeptiert `USD` und `EUR`. Wenn die AWS-Partition `aws-eusc` (AWS European Sovereign Cloud) lautet, muss der Währungscode `EUR` lauten.

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

```

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

```

Dieses Feld steht allen Partnern zur Verfügung und kann verwendet werden, um KI-Prognosen zu akzeptieren, indem derselbe Wert festgelegt wird, oder um es mit Partner-provided Werten zu überschreiben.

Bereitstellung von URLs für Preisrechner

Partner können optional AWS Preisrechner-URLs über das `Project.ExpectedCustomerSpend[].EstimationUrl` Feld angeben, um Servicekonfigurationen automatisch zu importieren und erweiterte Einblicke zu erhalten, darunter MAP-Eignungsindikatoren, Optimierungsempfehlungen und Kosteneinsparungsanalysen.

```

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

```

Wenn eine gültige URL für den Preisrechner angegeben wird, berechnet das System automatisch den MRR-Betrag aus der Rechnerkonfiguration und füllt sowohl das Feld Betrag als auch detaillierte Ausgabendaten auf Produktebene in der Partnerquelle von aus. `AwsProductsSpendInsightsBySource` Partner sollten bei der Angabe eines Betrags den Wert „Betrag“ auf Null setzen `EstimationUrl` — das System berechnet ihn automatisch.

URLs in ungültigen Formaten und URLs, die nicht aufgelöst werden können, werden mit einem `ValidationException` zurückgewiesen.

Wie viel und wie `EstimationUrl` funktionieren

Sie sind flexibel, wenn es darum geht, Schätzungen der monatlichen wiederkehrenden Umsätze (MRR) für Opportunities abzugeben. Die API verarbeitet auf intelligente Weise verschiedene Kombinationen aus manuellen Beträgen und AWS Preisrechner-URLs, um sicherzustellen, dass Ihre Geschäftschancen immer erfolgreich erstellt werden können.

Verwenden Sie nur einen manuellen Betrag

Wenn Sie nur das Feld `Betrag` angeben, speichert die API Ihre manuelle Schätzung und setzt `EstimationUrl` auf Null. Dieser Ansatz ist nützlich, wenn Sie aufgrund von Kundengesprächen oder Ihren eigenen Berechnungen eine bestimmte MRR-Schätzung haben.

Beispielanforderung:

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

Verwenden Sie nur ein AWS URL für den Preisrechner

Wenn Sie nur das `EstimationUrl` Feld angeben:

- Gültige URL — Die API berechnet den MRR-Betrag anhand Ihrer Rechnerkonfiguration und speichert sowohl die URL als auch den berechneten Betrag.
- Ungültige URL — Die API gibt eine `ValidationException` mit Details darüber zurück, warum die URL ungültig ist.

Beispielanforderung:

```
{
```

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

Beispiel für eine Fehlerantwort (ungültige URL):

```
{
  "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"
}
```

Angabe von Betrag und EstimationUrl

Wenn Sie beide Felder angeben, legt die API Wert darauf, dass Ihre Opportunity erfolgreich erstellt wird:

1. Wenn beide Werte unverändert sind, wird für diese Felder keine Aktualisierung durchgeführt
2. URL ist ungültig — Die API speichert Ihren angegebenen Betrag und setzt ihn EstimationUrl auf Null, sodass Ihre Opportunity-Erstellung fortgesetzt werden kann.
3. Die URL ist gültig und der berechnete Betrag entspricht Ihrem angegebenen Betrag. Beide Werte werden gespeichert.
4. Die URL ist gültig, aber der berechnete Betrag entspricht nicht Ihrem angegebenen Betrag. Die API speichert Ihren angegebenen Betrag und setzt ihn EstimationUrl auf Null, ohne dass ein Fehler zurückgegeben wird.

 Note

Wenn beide Werte nicht mit den vorhandenen Werten übereinstimmen, versucht die API zunächst, den Betrag anhand der URL zu berechnen. Ihr manuell eingegebener Betrag hat immer Vorrang, wenn es eine Nichtübereinstimmung oder ein Validierungsproblem gibt.

Wichtiger Vorteil: Ungültige URLs blockieren niemals die Erstellung von Opportunities

Dieser Ansatz stellt sicher, dass ungültige oder unzugängliche URLs für den AWS Preisrechner Sie niemals daran hindern, Opportunities zu erstellen oder zu aktualisieren. Ihr manuell eingegebener Betrag hat immer Vorrang, wenn ein Konflikt oder ein Validierungsproblem auftritt, sodass Sie die volle Kontrolle über Ihre Opportunity-Daten haben.

Das erweiterte `AwsOpportunitySummary` Datenmodell verstehen

In diesem Abschnitt wird die von der `GetAwsOpportunitySummary` API-Aktion zurückgegebene Antwortstruktur beschrieben.

 Note

AI-forecasted MRR wird zurückgegeben in `Project.ExpectedCustomerSpend[] .Amount`

Die `GetAwsOpportunitySummary` API-Aktion liefert dank der neuen `Insights.AwsProductsSpendInsightsBySource` Struktur, die eine umfassende Ausgabenanalyse mit Quellenzuweisung bietet, Einblicke in die Geschäftsgröße.

Trennung der Quellen: AWS im Vergleich zu Partnerdaten

Um Transparenz zu schaffen und Doppelzählungen zu vermeiden, werden die Erkenntnisse bei der Transaktionsgröße nach Quellen getrennt:

- AWS Quelle: KI-Produktempfehlungen von AWS
- Partnerquelle: Ausgabendaten und Produktdetails wurden aus den vom Partner bereitgestellten Preisrechner-URLs importiert

Diese Trennung ermöglicht Partnern:

1. Ihre Schätzungen mit den AWS Empfehlungen vergleichen
2. Bestätigen Sie ihre Größenannahmen
3. Vermeiden Sie es, die Gesamtwerte zu erhöhen, indem Sie versehentlich beide Quellen kombinieren
4. Behalten Sie den Überblick über die Herkunft der Daten

Überblick über die Struktur

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

Feldreferenz

AwsProductsSpendInsightsBySource Landkarte

Source-separated Ausgabeninformationen, die unabhängige Analysen für AWS Empfehlungen und Schätzungen der Partner ermöglichen.

Das `Insights.AwsProductsSpendInsightsBySource` Feld ist eine Map mit bis zu zwei Schlüsseln:

Key (Schlüssel)	Typ	Description
AWS	AwsProductsSpendInsights	AI-generated Einblicke, einschließlich empfohlener Produkte von AWS
Partner	AwsProductsSpendInsights	Partner-sourced Erkenntnisse aus den URLs des Preisrechners, einschließlich detaillierter Servicekosten und Optimierungen

Note

Nur Quellen mit verfügbaren Daten sind in der Antwort enthalten. Neue Geschäftschancen beginnen in der Regel damit, dass nur die AWS Quelle angegeben wird.

AwsProductInsights Objekt

Umfassende Ausgabenanalyse für eine einzige Quelle (AWS oder einen Partner), einschließlich Gesamtbeträgen, Optimierungseinsparungen, Aufschlüsselung der Programmkategorien und detaillierter Einblicke auf Produktebene.

Jedes Quellobjekt enthält die folgenden Felder:

Feld	Typ	Description
CurrencyCode	Zeichenfolge	Währungscode nach ISO 4217. Unterstützte Werte sind „USD“ und „EUR“.
Frequency (Frequenz)	Zeichenfolge	Gibt an, wie oft der Kunde den voraussichtlichen Betrag voraussichtlich ausgeben wird. Für das Feld Häufigkeit ist nur der Wert Monatlich zulässig,

Feld	Typ	Description
		der wiederkehrende monatliche Ausgaben darstellt.
TotalAmount	Zeichenfolge	Geschätzte Gesamtausgaben für diese Quelle vor Optimierungen
TotalOptimizedAmount	Zeichenfolge	Geschätzte Gesamtausgaben nach Anwendung der empfohlenen Optimierungen
TotalPotentialSavingsAmount	Zeichenfolge	Quantifizierte Einsparungen, die durch die Implementierung von Optimierungen erzielt werden können
TotalAmountByCategory	object	Die Ausgabenbeträge sind Programmen und Modernisierungspfaden AWS zugeordnet
AwsProducts	Array	Product-level Einzelheiten, einschließlich Kosten und Optimierungsempfehlungen

Beschränkung für den aktuellen Status: Für die AWS Quelle TotalPotentialSavingsAmount können TotalAmount, TotalOptimizedAmount, und weggelassen werden, wenn keine Ausgabenempfehlungen pro Produkt verfügbar sind. Partner sollten in diesen `Project.ExpectedCustomerSpend[].Amount` Fällen die gesamte AWS MRR-Schätzung als Referenz angeben.

TotalAmountByCategory Objekt

Maps gibt Beträge für AWS Programme und strategische Initiativen aus:

Schlüssel zur Kategorie	Description
MAP-fähig	Ausgaben für Dienstleistungen, die für eine Finanzierung durch das Migration Acceleration Program in Frage kommen
CAT-fähig	Dienstleistungen zur Unterstützung der Kostenzuweisung und -verfolgung
Pathway — Wechseln Sie zu Cloud Native	Services, die auf die cloudnative Modernisierung ausgerichtet sind
Pathway — Umstellung auf Managed Services	Dienste, die die Einführung von Managed Services unterstützen
Pathway — Umstellung auf Open Source	Alternativen zu Open-Source-Diensten
Pathway — Wechseln Sie zu Containern	Container-based Serviceoptionen
Pathway — Umstellung auf Serverless	Serverlose Computerdienste
Pathway — Umstellung auf Datenbanken	Dienstleistungen zur Modernisierung von Datenbanken
Pathway — Wechseln Sie zu Analytics	Dienstleistungen im Bereich Analytik und Datenverarbeitung

Note

Die Gesamtanzahl der Kategorien kann höher sein TotalAmount , da einzelne Produkte mehreren Kategorien angehören können.

AwsProducts Reihe

Liste der AWS Dienstleistungen mit Indikatoren für die Eignung des Programms (MAP, Modernisierungspfade), Kostenschätzungen und Optimierungsempfehlungen.

Jedes Produktobjekt enthält:

Feld	Typ	Erforderlich	Beschreibung
ProductCode	Zeichenfolge	Ja	AWS Die Produktkennzeichnung von Partner Central wird für die Zuordnung von Geschäftschancen verwendet
ServiceCode	Zeichenfolge	Nein	Servicecode für den Preisrechner (Links zur ursprünglichen URL des Rechners)
Kategorien	Array	Ja	Liste der Programm- und Pathway-Kategorien, für die dieses Produkt in Frage kommt
Betrag	Zeichenfolge	Nein	Basiskosten für den Service vor Optimierungen (bei Empfehlungen AWS aus eigener Quelle können sie Null sein)
OptimizedAmount	Zeichenfolge	Nein	Servicekosten nach der Anwendung von Optimierungen (können bei Empfehlungen aus anderen Quellen Null sein)AWS
PotentialSavingsAmount	Zeichenfolge	Nein	Service-specific Kostensenkung durch Optimierungen (kann

Feld	Typ	Erforderlich	Beschreibung
			bei Empfehlungen aus anderen Quellen Null sein)AWS
Optimierungen	Array	Nein	Liste der spezifischen Optimierungsempfehlungen für dieses Produkt

Behandlung mit Nullwerten: Bei Produkten AWS mit Herkunft können Betrag und PotentialSavingsAmount Felder den Wert Null haben OptimizedAmount, wenn keine Ausgabenempfehlungen pro Produkt verfügbar sind. Das System stellt stattdessen die gesamte MRR bereit. `Project.ExpectedCustomerSpend[0].Amount`

Optimierungsobjekt

Jede Optimierungsempfehlung enthält:

Feld	Typ	Description
Description	Zeichenfolge	Human-readable Erläuterung der Optimierungsstrategie
SavingsAmount	Zeichenfolge	Quantifizierte Kosteneinsparungen, die durch die Implementierung dieser Optimierung erzielt werden können

Beispiele

Beispiel 1:AWS Nur Empfehlungen (aktueller Stand)

Dieses Beispiel zeigt die typische Reaktion, wenn nur KI-Empfehlungen verfügbar sind und Ausgabenbeträge pro Produkt noch nicht empfohlen werden können.

```

{
  "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"
    ]
}
}

```

Wichtige Punkte: Die AWS Quelle zeigt empfohlene Produkte mit Kategorien, aber ohne Beträge pro Produkt. Die gesamte MRR-Schätzung ist verfügbar unter. `Project.ExpectedCustomerSpend[].Amount`

Beispiel 2: Import des Partner-Preisrechners

Dieses Beispiel zeigt verbesserte Einblicke, wenn ein Partner eine gültige Preisrechner-URL angegeben hat.

```

{
  "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": []
}
}

```

Wichtige Punkte: Die Partnerquelle enthält vollständige Ausgabedetails mit Beträgen pro Produkt. Optimierungsempfehlungen bieten umsetzbare Strategien zur Kostensenkung. Aufschlüsselungen nach Kategorien belegen die MAP-Eignung (22.500\$ von insgesamt 32.500\$). Zu den Produkten gehört ein ServiceCode Link zurück zur Konfiguration des Preisrechners.

Beispiel 3: Beide Quellen sind vorhanden (Vergleichsszenario)

Dieses Beispiel zeigt, wie sowohl AWS Empfehlungen als auch Schätzungen der Partner zusammen erscheinen, was einen Vergleich ermöglicht.

```

{
  "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"
  ]
}
}

```

Wichtigste Punkte: Beide Quellen enthalten: AWS Empfehlungen (insgesamt 28.000\$) und Partnerrechner (insgesamt 32.5.000\$). AWS Empfehlungen zeigen 4 Produkte; Partnerrechner zeigt 4 Produkte (3 überlappen sich). Partner können AWS Empfehlungen mit ihren detaillierten Schätzungen vergleichen. AWS insgesamt verfügbar über `Project.ExpectedCustomerSpend[].Amount`.

Bewährte Methoden

Umgang mit Nullwerten

1. EstimationUrl Feld: Erwarte für EstimationUrl die meisten Partner einen Wert von Null. Dies ist ein Standardverhalten und sollte nicht als Fehler behandelt werden. Zeigt manuelle MRR-Eingabeoptionen als primären Pfad an.
2. Fehlende Kategorien: Wenn diese Angabe für die AWS Quelle nicht vorhanden `TotalAmountByCategory` ist, bedeutet dies, dass keine Produktempfehlungen verfügbar sind.

Aufschlüsselungen nach Kategorien sind für Partner-sourced Daten verfügbar, wenn Preisrechner-URLs angegeben werden.

Empfehlungen zur Optimierung

1. Umsetzbare Erkenntnisse anzeigen: Zeigen Sie Optimierungsbeschreibungen und Sparbeträge gut sichtbar an, damit Partner die Möglichkeiten zur Kostensenkung besser verstehen.
2. Gesamteinsparungen: Summe aller PotentialSavingsAmount Produkte, um das gesamte Optimierungspotenzial aufzuzeigen.
3. Priorisieren Sie nach Wirkung: Sortieren Sie Optimierungen nach, SavingsAmount damit sich die Partner auf die Empfehlungen mit der größten Wirkung konzentrieren können.

Überwachung auf Aktualisierungen

1. Regelmäßige Umfrage: Führen Sie während der Workflows zur Opportunity-Erstellung `GetAwsOpportunitySummary` regelmäßige Umfragen durch (empfohlen: alle 5 Minuten).
2. Asynchrone Generierung handhaben: Erkenntnisse zur Geschäftsgröße sind möglicherweise nicht sofort nach der Erstellung von Geschäftschancen verfügbar. Implementieren Sie geeignete Ladestatus und Wiederholungslogik.

API-Integrationsmuster

1. Nutzen Sie bestehende Integrationen: Partner, die bereits aufrufen, müssen `GetAwsOpportunitySummary` nur zusätzliche Felder in der Antwort verwenden — es sind keine neuen API-Aufrufe erforderlich.
2. Graceful Degradation: Entwerfen Sie Benutzeroberflächen für Szenarien, in denen Deal-Sizing-Daten noch nicht verfügbar oder unvollständig sind.

Datenqualität und Genauigkeit

1. Überprüfen Sie die URLs des Preisrechners: Stellen Sie vor dem Absenden sicher, dass die URLs gültig sind, um `EstimationUrl` Nullantworten zu vermeiden.
2. Geben Sie ausführliche Informationen zu Geschäftschancen an: Umfassendere Informationen zu Geschäftschancen (Kundendetails, Projektbeschreibung, technische Anforderungen) führen zu genaueren KI-Empfehlungen.

3. Varianz überprüfen: Wenn die Schätzungen der Partner erheblich von den AWS Empfehlungen abweichen, überprüfen Sie die Opportunity-Details, um sicherzustellen, dass der gesamte relevante Kontext erfasst wird.

Bewährte Methoden

Auf Ereignisse reagieren

Stellen Sie bei der Verarbeitung von Ereignissen über die AWS Partner Central API sicher, dass Ihre Verarbeitungslogik für die Verarbeitung doppelter Ereignisse geeignet ist. Anstatt für jedes Ereignis sofort [GetOpportunity](#)-Aufrufe zu tätigen, sollten Sie erwägen, Details auf der Grundlage der Anforderungen Ihrer Anwendung stapelweise oder selektiv abzurufen. [Achten Sie für einen unterbrechungsfreien Betrieb auf Kontingente.](#)

Implementierung von optimistischem Sperren

Optimistisches Sperren verhindert unbeabsichtigte Datenüberschreibungen bei gleichzeitigen Aktualisierungen. Hier ist ein typisches Szenario:

1. Der Partner ruft eine Opportunity aus seinem CRM-System ab.
2. Der Benutzer A aktualisiert die Opportunity in AWS Partner Central.
3. Der Benutzer B aktualisiert dieselbe Opportunity gleichzeitig über die CRM-Integration.
4. Wenn sich die Daten ändern, versucht das CRM-System, die Daten hochzuladen, gibt aber eine `zurückConflictException` zurück.
5. Der Benutzer überprüft den Fehler und behebt die widersprüchlichen Daten manuell.

Um dieses Szenario zu vermeiden, müssen alle [UpdateOpportunity](#)-Anfragen den `LastModifiedDate` Parameter enthalten, den Sie aus vorherigen Aktionen [CreateOpportunity](#), [UpdateOpportunity](#), und [GetOpportunity](#) abrufen können. Das Update ist nur erfolgreich, wenn es zu unserem System `LastModifiedDate` passt. Wenn dies nicht der Fall ist, müssen Sie die neueste Version `LastModifiedDate` mit dem Update abrufen [GetOpportunity](#) und es erneut versuchen.

Synchronisieren von Daten zwischen CRM und AWS Partner Central

Es ist wichtig, dass Ihr System stets mit den neuesten Daten von Partner Central synchronisiert wird. Mit den folgenden zwei Strategien können Sie sicherstellen, dass Ihr System die neuesten Daten wiedergibt:

Ereignisse verwenden (empfohlen)

1. Daten laden mit [ListOpportunities](#).
2. Abonnieren Sie Opportunity-Events.
3. Reagieren Sie auf neue Möglichkeiten oder Veränderungen.
 - Wenn Sie Engagement Invitation Accepted Ereignisse erhalten Opportunity Created/Opportunity Updated, verwenden Sie diese, GetOpportunity um die neuesten Daten abzurufen.
 - Wenn Sie Engagement Invitation Rejected Ereignisse erhalten, entfernen Sie die entsprechenden Opportunities.

Verwenden Sie ListOpportunities Umfragen

1. Daten laden mit [ListOpportunities](#).
2. Wählen Sie eine Abfragefrequenz und achten Sie darauf, dass diese nicht zu häufig erfolgt, um zu vermeiden, dass Ihr tägliches [Lesekontingent](#) ausgeschöpft wird.
3. Identifizieren Sie die neuesten Daten LastModifiedDate aus Ihren gespeicherten Daten und stellen Sie sicher, dass sie von ihnen stammen.AWS
4. Verwenden Sie beim Aufrufen den Zeitstempel im AfterLastModifiedDate Filter.

[ListOpportunities](#)

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

5. AWS gibt Opportunities zurück, die nach dem auf dem Zeitstempel angegebenen Wert erstellt oder aktualisiert wurden.
6. Iterieren Sie mit Hilfe von NextToken allen zurückgegebenen Seiten und aktualisieren Sie die Daten Ihres Systems mit. [GetOpportunity](#)

```
{
```

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

Verwendung der AWS API für Vorteile von Partner Central

Die AWS Partner Central Benefits API optimiert die Art und Weise, wie Partner Leistungen in allen AWS Partnerprogrammen finden, beantragen und verwalten. Durch die Zentralisierung der Leistungsverwaltung in einer einzigen, intuitiven Oberfläche schafft der Service eine transparente Meritokratie, die Partner belohnt und gleichzeitig die Abläufe vereinfacht.

Die API bietet Standard-API-Funktionen. AWS Greifen Sie darauf zu, indem Sie entweder [API-Aktionen](#) oder ein AWS SDK verwenden, das auf Ihre Programmiersprache oder Plattform zugeschnitten ist. Weitere Informationen finden Sie unter [Erste Schritte AWS](#) und [Tools, auf denen Sie aufbauen können AWS](#).

Was ist ein Vorteil?

Im AWS Partnernetzwerk (APN) stellt ein Vorteil einen Vorteil oder eine Fähigkeit dar, die ein Partner nutzen kann, um sein Geschäftswachstum und seinen Kundenerfolg AWS zu unterstützen. Die Vorteile sollen Partner auf der Grundlage ihrer Qualifikationen und Beiträge belohnen und gleichzeitig einen konkreten Mehrwert bieten, der den Partnern hilft, ihre AWS Geschäftspraktiken zu skalieren.

Vorteile können viele Formen annehmen, darunter:

- **Finanzielle Anreize** — Kredite, Barauszahlungen und Finanzierungsprogramme wie das Migration Acceleration Program (MAP) oder Marketing Development Funds (MDF)
- **Access-Vorteile** — Vorab-Zugriff auf neue Services, Tools oder spezielle Ressourcen
- **Anerkennung** — Digitale Ausweise, Zertifizierungen, Kompetenzbezeichnungen und Status der Partnerstufe
- **Technische Ressourcen** — Unterstützung bei der Lösungsarchitektur, technischer Support und Kundenbetreuung ProServe

- Marketingunterstützung — Co-marketing Möglichkeiten, ausgewählter Partnerstatus und Veröffentlichungen zu Erfolgsgeschichten

Entdecken Sie die verfügbaren Vorteile

Partner beginnen ihre Reise mit den Vorteilen, indem sie herausfinden, welche Leistungen ihnen zur Verfügung stehen, und sich mit den Anspruchsvoraussetzungen vertraut machen. Die Benefits API bietet über die Benefit Discovery APIs umfassende Erkennungsfunktionen.

Liste der verfügbaren Vorteile

Partner können alle verfügbaren Vorteile mithilfe der `ListBenefits` API-Aktion einsehen. Mit dieser Aktion wird ein Leistungskatalog mit optionalen Filterfunktionen abgerufen, damit Partner schnell relevante Geschäftschancen finden können.

Um Vorteile effizient zu ermitteln, können Partner nach folgenden Kriterien filtern:

- Programm — Filtern Sie nach bestimmten AWS Partnerprogrammen wie MAP (Migration Acceleration Program), MDF (Marketing Development Funds), ISV Workload Migration, Innovation Sandbox, Proof of Concept oder Finanzierung von Partnerinitiativen
- Art der Auftragsabwicklung — Filtern Sie danach, wie die Leistungen erbracht werden: `CREDITS`, `CASH`, `DISCOUNT`, `ACCESS`, oder `RECOGNITION RESOURCE`
- Status — Nach Leistungsstatus filtern: `ACTIVE` oder `DEPRECATED`

Die `ListBenefits` API gibt Leistungsübersichten zurück, die wichtige Informationen wie Leistungs-ID, Name, Beschreibung, zugehörige Programme und Erfüllungsarten enthalten. Diese Übersichtsansicht ermöglicht es Partnern, schnell verfügbare Leistungen zu durchsuchen und diejenigen zu identifizieren, die für ihre Geschäftsziele am relevantesten sind.

Beispiel für einen Filteransatz:

Partner, die sich auf finanzielle Anreize konzentrieren, könnten nach Erfüllungsarten `CREDITS` filtern `CASH`, um Finanzierungsmöglichkeiten zu finden. Partner, die technische Unterstützung suchen, können nach der Art der Auftragsabwicklung filtern `RESOURCE` oder `ACCESS` nach Vorteilen des technischen Supports und dem Zugriff auf Tools suchen.

Detaillierte Informationen zu den Zusatzleistungen anzeigen

Sobald ein Partner einen Vorteil identifiziert hat, der für ihn von Interesse ist, kann er mithilfe der `GetBenefit` API-Aktion vollständige Informationen abrufen. Dies bietet umfassende Informationen, darunter:

- Zulassungskriterien — Detaillierte Anforderungen, die Partner erfüllen müssen, um die Leistung in Anspruch nehmen zu können
- Schema für die Beantragung von Leistungen — Die spezifischen Informationen und Unterlagen, die die Partner bei der Beantragung vorlegen müssen
- Einzelheiten des Zuschusses — Was erhalten die Partner nach der Genehmigung
- Verbundene Programme — Welche AWS Partnerprogramme werden durch den Zuschuss unterstützt

Das Schema für Leistungsanträge ist besonders wichtig, da es die Struktur und die erforderlichen Felder für Leistungsanträge definiert. Partner sollten dieses Schema sorgfältig prüfen, bevor sie einen Leistungsantrag stellen, um sicherzustellen, dass sie alle erforderlichen Informationen im Voraus sammeln.

Important

Die Entscheidung über die Eignung der einzelnen Leistungen erfolgt durch den Kunden oder die Benutzeroberflächenebene. Bei Leistungen im Zusammenhang mit der Finanzierung hängt die Anspruchsberechtigung in der Regel von der Scorecard-Leistung des Partners und der Teilnahme am Programm ab.

Topics

- [Mit Leistungsanträgen arbeiten](#)
- [Arbeiten mit Leistungszuweisungen](#)

Mit Leistungsanträgen arbeiten

Ein Leistungsantrag modelliert die Anfrage eines Partners nach einer bestimmten Leistung. Darin werden alle Informationen erfasst, die für die Bewertung und Bearbeitung des Antrags gemäß den leistungsspezifischen Bedingungen erforderlich sind. Der Lebenszyklus des Leistungsantrags

durchläuft mehrere Phasen, von der Erstellung des Entwurfs bis zur endgültigen Genehmigung oder Ablehnung.

Leistungsanträge erstellen

Partner initiieren den Prozess zur Beantragung von Leistungen, indem sie mithilfe der `CreateBenefitApplication` API-Aktion einen Leistungsantrag erstellen. Bei der Erstellung geht der Antrag in den `PENDING_SUBMISSION` Status über, sodass die Partner vollständige Informationen vorbereiten können, bevor sie ihn zur Prüfung einreichen.

Bei der Erstellung eines Leistungsantrags müssen die Partner Folgendes angeben:

- **Leistungs-ID** — Die ID oder ARN der beantragten Leistung
- **Erfüllungsarten** — Wie der Partner die Leistung voraussichtlich erhalten wird (`CREDITCASH`, `DISCOUNT`, `ACCESS`, `RECOGNITION`, oder `RESOURCE`)
- **Einzelheiten des Leistungsantrags** — Ein JSON-Dokument mit leistungsspezifischen Informationen, wie sie im Antragsschema der Leistung definiert sind
- **Kunden-Token** — Ein einzigartiges Idempotenz-Token, um doppelte Einreichungen zu verhindern

Partner können optional Folgendes bereitstellen:

- **Name und Beschreibung** — Human-readable Identifikatoren für die Anwendung
- **Partnerkontakte** — Kontaktinformationen der Person, die diesen Leistungsantrag bearbeitet (maximal 1 Ansprechpartner)
- **Dateianhänge** — Unterstützende Unterlagen wie Projektpläne, Leistungsbescheinigungen, Kostennachweise oder Umfragen zur Kundenzufriedenheit (maximal 10 Dateien)
- **Assoziierte Ressourcen** — Links zu verwandten Möglichkeiten oder bestehenden Leistungszuweisungen (maximal 10 Ressourcen)
- **Stichwörter** — Key-value Paare für die Organisation und Nachverfolgung von Ressourcen (maximal 200 Stichwörter)

Die `CreateBenefitApplication` API führt eine sanfte Validierung durch und überprüft nur grundlegende Feldtypen und Muster. Während der Einreichung erfolgt eine vollständige Validierung der Geschäftslogik, sodass Partner unvollständige Anträge speichern und später zurückkehren können, um sie auszufüllen.

Bewährtes Verfahren: Partner sollten das Schema für Leistungsanträge von `verwendenGetBenefit`, um genau zu verstehen, welche Informationen vor der Erstellung eines Antrags erforderlich sind. Dadurch wird die Wahrscheinlichkeit verringert, dass die Einreichung aufgrund fehlender oder ungültiger Daten fehlschlägt.

Aktualisierung der Leistungsanträge

Partner können Entwürfe von Leistungsanträgen mithilfe der `UpdateBenefitApplication` API-Aktion ändern. Auf diese Weise können Partner Informationen verfeinern, Unterlagen hinzufügen oder Fehler vor der Einreichung korrigieren.

Bei der Aktualisierung eines Leistungsantrags müssen die Partner Folgendes angeben:

- Kennung — Die ID oder ARN des Leistungsantrags, der aktualisiert werden soll
- Revision — Die aktuelle Revisionsnummer für optimistisches Sperren
- Einzelheiten zum Antrag auf Leistungen — Das vollständige, aktualisierte JSON-Dokument

Partner müssen den vollständigen Leistungsantrag einreichen, auch wenn sich nur bestimmte Felder ändern. Es empfiehlt sich, zunächst die neuesten Antragsdetails abzurufen `GetBenefitApplication`, indem Sie die erforderlichen Felder ändern und dann die vollständige aktualisierte Payload an `UpdateBenefitApplication` senden.

Optimistisches Sperren: Das Revisionsfeld stellt sicher, dass Aktualisierungen nur angewendet werden, wenn sich die Anwendung seit dem letzten Abruf nicht geändert hat. Wenn die Revision nicht mit dem aktuellen Datenbankwert übereinstimmt, wird das Update mit einem Konfliktfehler abgelehnt. Dadurch wird verhindert, dass Partner versehentlich Änderungen überschreiben, die von anderen Prozessen oder Systemen vorgenommen wurden.

Aktualisierungen können nur durchgeführt werden, solange sich die Anwendung im `PENDING_SUBMISSION` Status befindet. Sobald die Anträge eingereicht wurden, werden sie geprüft und können nicht mehr über diese API aktualisiert werden.

Ressourcen mit Leistungsanträgen verknüpfen

Partner können mithilfe der `AssociateBenefitApplicationResource` API-Aktion Leistungsanträge mit verwandten Ressourcen verknüpfen. Dadurch entstehen wertvolle Verbindungen zwischen Leistungen und dem Geschäftskontext, in dem sie genutzt werden.

Unterstützte Ressourcentypen sind u. a.:

- **GESCHÄFTSMÖGLICHKEIT** — Verknüpfen Sie den Antrag auf Leistungen mit einer bestimmten Kundenchance in. Dies ist besonders nützlich bei chancenspezifischen Leistungen wie MAP-Finanzierung oder POC-Gutschriften, die an Kundeninteraktionen gebunden sind.
- **BENEFIT_ALLOCATION** — Link zu bestehenden Leistungszuweisungen, sodass Partner Leistungen miteinander verknüpfen oder nachweisen können, wie frühere Leistungen genutzt werden

Eine Zuordnung der Ressourcen kann nur vor der Einreichung erfolgen. Sobald ein Leistungsantrag eingereicht wurde (der Status ändert sich auf `IN_REVIEW`), sind keine weiteren Verknüpfungen mehr zulässig. Partner können jedem Leistungsantrag bis zu 10 Ressourcen zuordnen.

Um die Zuordnung einer Ressource aufzuheben, verwenden Partner die `DisassociateBenefitApplicationResource` API-Aktion. Wie die Zuordnung kann auch die Trennung nur im `PENDING_SUBMISSION` Status erfolgen.

 Important

Partner müssen über die entsprechenden Berechtigungen verfügen, um sowohl auf den Leistungsantrag zugreifen als auch die spezifische Ressource lesen zu können, der sie zugeordnet sind. Die API validiert diese Berechtigungen während des Zuordnungsvorgangs.

Einreichung von Leistungsanträgen

Wenn ein Antrag auf Leistungen vollständig und zur AWS Prüfung bereit ist, reichen die Partner ihn über die `SubmitBenefitApplication` API-Aktion ein. Die Einreichung löst einen Statusübergang von `PENDING_SUBMISSION` zu `IN_REVIEW` und leitet den leistungsspezifischen Genehmigungsworkflow ein.

Nach der Einreichung führt die API eine umfassende Validierung durch, die Folgendes umfasst:

- **Überprüfung der erforderlichen Felder** — Stellt sicher, dass alle Pflichtfelder im Leistungsantrag vorhanden sind
- **Überprüfung des Feldformats** — Überprüft, ob die Feldwerte den erwarteten Mustern und Datentypen entsprechen
- **Validierung von Geschäftsregeln** — Wendet nutzenspezifische Geschäftslogik und Einschränkungen an
- **Ressourcenvalidierung** — Bestätigt, dass alle zugehörigen Ressourcen gültig und zugänglich sind

- **Dateiüberprüfung** — Überprüft, ob die Verarbeitung aller Dateianhänge erfolgreich abgeschlossen wurde

Schlägt die Überprüfung fehl, gibt die API eine `ValidationException` mit detaillierten Fehlercodes und Meldungen zurück, die angeben, welche Felder korrigiert werden müssen. Partner müssen diese Probleme mithilfe von Internet lösen, `UpdateBenefitApplication` bevor sie erneut versuchen, die Daten einzureichen.

Anforderung zur Dateiverarbeitung: Leistungsanträge können nicht eingereicht werden, wenn angehängte Dateien noch im `PENDING` Status sind. Partner müssen warten, bis die Dateiverarbeitung abgeschlossen ist (der Status ändert sich auf `SUCCEEDED`), bevor sie sie einreichen. Wenn Dateien nicht verarbeitet werden können (`StatusFAILED`), sollten Partner die korrigierten Versionen hochladen.

Nach erfolgreicher Einreichung:

- Der Status der Bewerbung ändert sich auf `IN_REVIEW`
- Das Team des Leistungsempfängers wird über die neue Einreichung informiert
- Partner können Anwendungsdetails nicht mehr über Standard-Update-APIs ändern
- Für die Anwendung gilt ein definierter Genehmigungsablauf, der die Phasen der Geschäftsgenehmigung, der technischen Genehmigung und der finanziellen Genehmigung umfassen kann

Partner können den Fortschritt der Einreichung verfolgen, indem sie den Antrag aufrufen und das Feld `Phase` überwachen, das die aktuelle Genehmigungsphase angibt (z. B. „Geschäftsgenehmigung“, „Technische Genehmigung“, „Finanzielle Genehmigung“).

Verwaltung der eingereichten Anträge

Nach der Einreichung haben Partner nur begrenzte Möglichkeiten, Leistungsanträge zu verwalten. Die API bietet jedoch spezifische Aktionen für gängige Szenarien.

Rückruf von Leistungsanträgen

Wenn ein Partner nach der Einreichung einen Fehler entdeckt oder Änderungen vornehmen muss, kann er den Antrag mithilfe der `RecallBenefitApplication` API-Aktion zurückrufen. Durch Rückruf wird der Antrag wieder in den `PENDING_SUBMISSION` Status versetzt, sodass Aktualisierungen und eine erneute Einreichung möglich sind.

Beim Rückruf einer Bewerbung sollten die Partner Folgendes angeben:

- Kennung — Die ID oder ARN des Leistungsantrags, der zurückgerufen werden soll
- Grund — Eine optionale Erklärung für den Rückruf (maximal 1000 Zeichen)

Das Feld „Grund“ ermöglicht eine bessere Rückverfolgbarkeit und hilft dabei, häufig auftretende Probleme zu AWS verstehen, die zu Rückrufen führen, sodass future Verbesserungen des Verfahrens zur Beantragung von Leistungen berücksichtigt werden können.

Nach dem Rückruf können die Partner die erforderlichen `UpdateBenefitApplication` Änderungen vornehmen und sie dann erneut einreichen, wenn sie bereit sind. `SubmitBenefitApplication`

 Important

Der Rückruf ist in der Regel nur in frühen Überprüfungsphasen möglich. Anträge, die sich in einer späteren Genehmigungsphase befinden oder genehmigt wurden, können möglicherweise nicht zurückgerufen werden.

Änderung von Leistungsanträgen

Bei geringfügigen Korrekturen nach der Einreichung können Partner die `AmendBenefitApplication` API-Aktion verwenden, um bestimmte Felder zu aktualisieren, ohne die gesamte Anwendung erneut aufrufen zu müssen. Dies ist besonders nützlich, wenn AWS Gutachter während des Überprüfungsprozesses Klarstellungen oder Korrekturen anfordern.

Änderungsanträge verwenden einen Patch-style JSON-Ansatz, bei dem die Partner Folgendes angeben:

- Path — Ein `JsonPath`-Ausdruck, der das zu aktualisierende Feld identifiziert (z. B. `$.CreditDisbursementDetails.AwsAccountIdForCredits`)
- Wert — Der neue Wert für das Feld
- Operation — Der auszuführende Vorgang (`REPLACE` wird derzeit nur unterstützt)

Partner können in einem einzigen API-Aufruf bis zu 10 Änderungen einreichen. Für optimistisches Sperren muss jede Änderung eine aktualisierte Revisionsnummer enthalten.

Änderungsanträge sollten für geringfügige Korrekturen verwendet werden. Bei wesentlichen Änderungen sollten die Partner den Antrag `RecallBenefitApplication` in den Entwurfsstatus zurückversetzen, um ihn umfassend zu aktualisieren.

Stornierung von Leistungsanträgen

Wenn ein Partner eine Leistung nicht mehr benötigt oder seinen Antrag zurückziehen möchte, kann er ihn mithilfe der `CancelBenefitApplication` API-Aktion stornieren. Durch die Stornierung wird der Antrag in den `CANCELED` Status versetzt, wodurch der Bewerbungsprozess dauerhaft beendet wird.

Bei der Stornierung einer Bewerbung sollten die Partner Folgendes angeben:

- Kennung — Die ID oder ARN des zu stornierenden Leistungsantrags
- Grund — Eine optionale Erklärung für die Kündigung (maximal 1000 Zeichen)

Einmal stornierte Anwendungen können nicht reaktiviert werden. Wenn der Partner später entscheidet, dass er die Leistung benötigt, muss er einen neuen Leistungsantrag stellen.

Zu den häufigsten Kündigungsgründen gehören:

- Die Kundenchance wurde verpasst oder verspätet
- Die Kapazitätsbeschränkungen der Partner haben sich geändert
- Die Prioritäten der Unternehmen haben sich verschoben
- Die Leistung wird für den vorgesehenen Zweck nicht mehr benötigt

Einzelheiten zum Leistungsantrag anzeigen

Partner können mithilfe der `GetBenefitApplication` API-Aktion vollständige Informationen zu einem Leistungsantrag abrufen. Dies bietet einen umfassenden Überblick über die Anwendung, einschließlich:

Metadaten der Anwendung:

- Eindeutige Kennung (ID) und Amazon-Ressourcenname (ARN)
- Kennung der zugehörigen Leistung
- Name und Beschreibung der Anwendung

- Aktueller Status (PENDING_SUBMISSIONIN_REVIEW,ACTION_REQUIRED,APPROVED,REJECTED, oderCANCELED)
- Aktuelle Verarbeitungsphase

Informationen zum Status:

- Statusgrund — Human-readable Erläuterung des aktuellen Status
- Status-Ursachencodes — Strukturierte Codes, die auf bestimmte Probleme oder Anforderungen hinweisen (z. B. „Checkliste unvollständig“, „Projektplan fehlt“, „Design Win fehlt“)

Inhalt der Bewerbung:

- Vollständige Angaben zum Leistungsantrag (JSON-Dokument)
- Kontaktinformationen des Partners
- Dateianhänge mit Bearbeitungsstatus
- Dazugehörige Ressourcen (Opportunities oder Allokationen)
- Ressourcen-Tags

Informationen zur Prüfung:

- Zeitstempel der Erstellung
- Zeitstempel der letzten Änderung
- Aktuelle Revisionsnummer

Die Codes für die Statusursache sind besonders nützlich, wenn sich eine Anwendung im ACTION_REQUIRED Status befindet. Diese Codes bieten konkrete, umsetzbare Hinweise darauf, was Partner beachten müssen, damit der Antrag bearbeitet werden kann.

Auflistung von Leistungsanträgen

Partner können mithilfe der ListBenefitApplications API-Aktion alle ihre Leistungsanträge einsehen. Dadurch wird eine paginierte Liste von Anwendungsübersichten mit leistungsstarken Filterfunktionen zurückgegeben.

Partner können Anwendungen nach folgenden Kriterien filtern:

- Programm — Anwendungen für bestimmte Programme anzeigen (MAP, MDF, Sandbox, POC usw.)
- Versandart — Nach Versandart filtern (CREDITS, CASHACCESS, usw.)
- Benefit Identifier — Anträge für eine bestimmte Leistung anzeigen
- Status — Nach Antragsstatus filtern (PENDING_SUBMISSION, IN_REVIEW, APPROVED, REJECTED, CANCELED)
- Phase — Nach Genehmigungsphase filtern (Geschäftsgenehmigung, technische Genehmigung, Finanzgenehmigung)
- ARNs für zugehörige Ressourcen — Suchen Sie nach Bewerbungen, die mit bestimmten Stellenangeboten oder Zuweisungen verknüpft sind

Die Antwort auf die Liste enthält wichtige zusammenfassende Informationen wie:

- Anwendungs-ID, ARN und Name
- ID der zugehörigen Leistung
- Programme und Arten der Auftragsabwicklung
- Aktueller Status und Stand
- Zeitstempel für Erstellung und Änderung
- Zugeordnete Ressourcen
- Aktuelle Revisionsnummer
- Ausgewählte Felder aus den Angaben zum Leistungsantrag (wie vom Leistungsempfänger definiert)

Partner können die Ergebnissortierung mithilfe des Sortierparameters konfigurieren. Dabei stehen Optionen zur Verfügung, um die Ergebnisse nach Erstellungsdatum, Status, Programm oder Leistungs-ID in aufsteigender oder absteigender Reihenfolge zu sortieren.

Erstellung von Dashboards: Die `ListBenefitApplications` API wurde entwickelt, um die Erstellung von Partner-Dashboards zu unterstützen. Durch das Filtern und Sortieren von Anwendungen können Partner Ansichten wie die folgenden erstellen:

- Anwendungen, für die Maßnahmen erforderlich sind (ACTION_REQUIREDStatus)
- Kürzlich eingereichte Bewerbungen (sortiert nach Erstellungsdatum, IN_REVIEW Status)
- Genehmigte Anträge, deren Zuteilung noch aussteht (APPROVEDStatus)

- Bewerbungen nach Programmen (nach bestimmten Programmen gefiltert)

Arbeiten mit Leistungszuweisungen

Eine Leistungszuweisung stellt eine Leistung dar, die einem Partner gewährt wurde. Wenn ein Leistungsantrag genehmigt wird, werden eine oder mehrere Leistungszuweisungen AWS erstellt, die den Partnern die tatsächlichen Gutschriften, die Finanzierung, den Zugang oder die Erfüllung anderer Leistungen ermöglichen.

Grundlegendes zu Leistungszuweisungen

Bei der Zuweisung von Leistungen kann es sich um verschiedene Arten der Leistungserfüllung handeln:

- Finanzielle Auszahlungen (CASH) — Direkte Finanzauszahlungen an Partner mit Auszahlungsverfolgung
- AWS Gutschriften (CREDITS) — Kreditcodes, die für Konten mit Nutzungsverfolgung gelten AWS
- Zuteilung verbrauchbarer Ressourcen — Pre-approved Finanzierungsbeträge oder Wallets mit Nutzungsverfolgung
- Access Grants (ACCESS) — Zugriff auf Systeme, Tools oder Ressourcen
- Anerkennung (ANERKENNUNG) — Digitale Ausweise, Zertifizierungen oder Statusbezeichnungen
- Ressourcen (RESOURCE) — Technischer Support, Beratungsdienste oder Aufträge ProServe

Jede Leistungszuweisung hat einen Lebenszyklus, der über das entsprechende Statusfeld verfolgt wird:

- AKTIV — Die Zuteilung kann verwendet werden
- INAKTIV — Die Zuordnung ist vorübergehend nicht verfügbar
- ERFÜLLT — Das Kontingent wurde vollständig genutzt oder zugestellt

Die Leistungszuweisungen enthalten auch Informationen zu bestimmten Zeitpunkten, in denen festgelegt wird, wann sie wirksam werden (StartsAt) und wann sie ablaufen (ExpiresAt), sodass die Partner den Zeitrahmen für die Inanspruchnahme der Leistungen nachvollziehen können.

Auflisten der Leistungszuweisungen

Partner können mithilfe der `ListBenefitAllocations` API-Aktion alle ihre Leistungszuweisungen einsehen. Dadurch wird eine paginierte Liste von Zuweisungsübersichten mit umfassenden Filterfunktionen zurückgegeben.

Partner können Allokationen nach folgenden Kriterien filtern:

- Benefit Identifier — Zuweisungen für eine bestimmte Leistung anzeigen
- Kennung des Leistungsantrags — Zeigt die Zuweisungen an, die sich aus einem bestimmten Antrag ergeben
- Erfüllungsart — Filtern Sie nach Zuweisungsart (`CREDITCASHACCESS`, usw.)
- Status — Nach Zuordnungsstatus filtern (`ACTIVE`, `INACTIVE`, `FULFILLED`)

Die Antwort auf die Liste enthält Zusammenfassungen der Allokationen, die Folgendes enthalten:

- Zuordnungs-ID, ARN und Name
- Zugeordnete Leistungs-ID und Leistungsantrags-ID (falls zutreffend)
- Art der Erfüllung
- Aktueller Status
- Zeitstempel der Erstellung
- Startdatum

Diese Listenansicht ermöglicht es Partnern, Dashboards zur Allokationsverfolgung zu erstellen und Folgendes zu identifizieren:

- Aktive Zuweisungen stehen zur sofortigen Verwendung zur Verfügung
- Zuteilungen, die bald ablaufen
- Erfüllte Zuweisungen für die historische Nachverfolgung
- Gesamtwert der Zuteilung, aufgeschlüsselt nach Programm oder Erfüllungsart

Detaillierte Zuordnungsinformationen anzeigen

Partner können mithilfe der `GetBenefitAllocation` API-Aktion vollständige Zuweisungsdetails abrufen. Auf diese Weise erhalten Sie umfassende Informationen über die Zuteilung, einschließlich abwicklungsspezifischer Details, die je nach Zuweisungstyp variieren.

Wichtige Informationen zur Allokation:

- Eindeutige Kennung (ID) und Amazon-Ressourcenname (ARN)
- Name und Status der Zuordnung
- Grund für den Status (Erläuterung des aktuellen Status)
- Zugeordnete Leistungs-ID und Leistungsantrags-ID
- Art der Erfüllung
- Zeitstempel für Erstellung, Aktualisierung, Start und Ablauf

Einzelheiten zur Auftragsabwicklung (Type-Specific):

Das FulfillmentDetail Feld enthält je nach Versandart unterschiedliche Informationsstrukturen:

Einzelheiten zur Barauszahlung (Art CASH)

Bei direkten Barzahlungen beinhalten die Einzelheiten zur Erfüllung:

- Ausbezahlter Betrag — Der Gesamtbetrag der ausgezahlten Finanzierung, einschließlich Betrag, Wert und Währungscode
- Emissionsdetails (falls zutreffend) — Informationen zu Bestellungen oder Emissionsereignissen:
 - Ausstellungs-ID — Eindeutige Kennung für die Auszahlung
 - Emissionsbetrag — Betrag in Landeswährung
 - Ausgestellt am — Zeitpunkt der Auszahlung

Diese Struktur unterstützt sowohl Einzelauszahlungen als auch vorab genehmigte Finanzierungsszenarien, bei denen mehrere Emissionen gegen eine einzige Zuteilung erfolgen können.

Kreditdetails (Art CREDITS)

Bei AWS Gutschriften beinhalten die Erfüllungsdetails:

- Zugewiesener Betrag — Gesamter zugewiesener Kreditbetrag
- Ausgegebener Betrag — Gesamtbetrag des ausgegebenen Kredits (möglicherweise geringer als der für die schrittweise Emission zugewiesene Betrag)
- Kreditcodes — Eine Reihe von individuellen Kreditcode-Details:
 - AWS Konto-ID — Konto, auf das der Kredit gutgeschrieben wird

- Wert — Der monetäre Wert des Kreditcodes
- AWS Kreditcode — Die tatsächliche Kreditcode-Zeichenfolge
- Status — Status des Kreditcodes (ACTIVE, INACTIVE, FULFILLED)
- Ausgestellt am — Zeitpunkt, an dem der Kreditcode ausgestellt wurde
- Läuft ab am — Wenn der Guthabencode abläuft

Diese detaillierte Kreditverfolgung ermöglicht es Partnern, die Kreditauslastung über mehrere Konten hinweg zu überwachen und nachzuvollziehen, welche Credits verfügbar, teilweise oder vollständig verbraucht sind.

Einzelheiten zur Zuordnung von Verbrauchsgütern () Pre-Approved Amounts/Wallets

Für vorab genehmigte Finanzierungspools beinhalten die Einzelheiten zur Auftragsabwicklung:

- Zugewiesener Betrag — Gesamtbetrag in der Zuweisung
- Verbleibender Betrag — Ungenutzter Betrag ist noch verfügbar
- Verwendete Menge — Bereits verbrauchte Menge
- Ausstellungsdetails (falls zutreffend) — Bestellinformationen für die Zuteilung

Durch die Zuteilung von Verbrauchsgütern können Partner bei Bedarf Mittel für förderfähige Aktivitäten abrufen und dabei den verbleibenden Restbetrag in Echtzeit verfolgen.

Zugangsdetails (ACCESS-Typ)

Bei Zugangsberechtigungen beinhalten die Erfüllungsdetails:

- Beschreibung — Ausführliche Erläuterung des gewährten Zugriffs und seiner Nutzung

Zugriffszuweisungen können den Zugriff auf Vorschaudienste, spezielle Tools oder eingeschränkte Ressourcen ermöglichen. Das Beschreibungsfeld enthält Anweisungen dazu, wie Partner den gewährten Zugriff aktivieren und nutzen können.

Zuweisungen auf neue Leistungsanträge anwenden

Das `ApplicableBenefitIds` Feld Leistungszuweisungen identifiziert weitere Leistungen, die diese Zuweisung nutzen können. Dies ermöglicht eine Verkettung von Leistungen, bei der Partner bestehende Zuweisungen nutzen können, um zusätzliche Leistungen zu beantragen.

Ein Partner könnte zum Beispiel:

1. Sie erhalten eine vorab genehmigte MAP-Mittelzuweisung
2. Erstellen Sie Leistungsanträge für individuelle Kundenmigrationsprojekte
3. Ordnen Sie diese Anwendungen der vorab genehmigten Zuteilung zu
4. Ziehen Sie Mittel aus der Mittelzuweisung in Anspruch, sobald die Projekte genehmigt sind

Dieser Ansatz rationalisiert das Verfahren zur Beantragung von Leistungen für Partner mit vorab bewilligten Mitteln und verkürzt so den Überprüfungszyklus für einzelne Projektanträge.

Verwendung der AWS Channel-API für Partner Central

AWS Mit den Partner Central Channel-APIs können Sie Ihr autorisiertes Wiederverkaufsgeschäft programmgesteuert verwalten. AWS Diese APIs ermöglichen AWS Lösungsanbietern, AWS Vertriebshändlern und AWS Vertriebsverkäufern:

- Für Wiederverkaufsprogramme verwendete AWS Verwaltungskonten an Partner Central melden
- Senden Sie Einladungen an AWS Konten, um Partnervereinigungen anzunehmen
- Stellen Sie Beziehungen her, die für das Onboarding weiterverkaufter AWS Endkundenkonten und die Gewährung von Partnerrabatten erforderlich sind

Mit den Channel-APIs von AWS Partner Central können Sie benutzerdefinierte Automatisierungen einrichten, um:

- Integrieren Sie neue AWS Konten, die sich im Besitz eines Partners befinden, in APN-Wiederverkaufsprogramme
- Integrieren Sie neue Endkundenkonten in Wiederverkaufsprogramme
- Beschleunigen Sie das Kunden-Onboarding und die Rabattqualifizierung

Arbeiten mit Program Management Accounts (PMAs)

Ein Programmverwaltungs-konto (PMA) verknüpft Ihr AWS Verwaltungskonto mit Ihrer Channel-Programm-Registrierung. Programmverwaltungs-konten werden per Self-Service mithilfe von Channel-Handshakes aktiviert. Wenn Sie eine PMA erstellen und aktivieren, werden die Vorteile und Rabatte des Channel-Programms auf alle Rechnungen angerechnet, die dem zugehörigen

Verwaltungskonto gutgeschrieben werden. AWS Verwenden Sie diese APIs, um AWS Konten für Wiederverkäufe und Channel-Rabatte zu melden und zu verwalten:

- `CreateProgramManagementAccount`: Fügen Sie neue Konten für die Verwaltung der Kundenabrechnungen hinzu
- `UpdateProgramManagementAccount`: Ändern Sie bestehende Konten
- `DeleteProgramManagementAccount`: Konten aus Wiederverkaufsprogrammen entfernen
- `ListProgramManagementAccounts`: Sehen Sie sich bestehende Konten und deren Status an

Arbeitet mit Kanal-Handshakes

Channel-Handshakes ermöglichen eine sichere, gegenseitige Zustimmung für kritische Channel-Management-Operationen. AWS Partner Central verwendet Channel-Handshakes, um die Zustimmung der AWS Verwaltungskonten für drei verschiedene Zwecke zu überprüfen: Aktivierung von Programmverwaltungskonten (PMAs), Einrichtung von Serviceperioden und vorzeitige Beendigung von Serviceperioden. Es gibt drei Typen:

- `PROGRAM_MANAGEMENT_ACCOUNT`: Bei der Aktivierung von PMAs zur Inanspruchnahme der Vorteile des Channel-Programms wird mit Handshakes für das Programmverwaltungskonto die Zustimmung bestätigt
- `START_SERVICE_PERIOD`: Mit Handshakes für die Festlegung des Leistungszeitraums werden wechselseitige Vereinbarungen über Abrechnungs- und Transferverpflichtungen mit Mindestkündigungsfristen oder festen Laufzeiten getroffen
- `REVOKE_SERVICE_PERIOD`: Handshakes zur Kündigung des Dienstzeitraums ermöglichen die vorzeitige Beendigung laufender Dienstzeiten, sofern beide Parteien zustimmen. Alle Handshake-Typen müssen vom AWS Zielverwaltungskonto akzeptiert werden, um wirksam zu werden.

Verwenden Sie diese APIs, um Channel-Handshakes für die PMA-Aktivierung und die Verwaltung des Servicezeitraums zu verwalten:

- `CreateChannelHandshake`: Erstellen Sie Handshakes für PMA-Einladungen, die Festlegung des Servicezeitraums oder die Kündigung des Servicezeitraums
- `AcceptChannelHandshake`: Akzeptiert Handshakes von AWS Zielkonten (kann je nach Handshake-Typ von Partnern oder Kunden verwendet werden)
- `RejectChannelHandshake`: Lehnen Sie Handshakes von Zielkonten ab AWS

- `CancelChannelHandshake`: Stornieren Sie ausstehende Handshakes, bevor sie akzeptiert, abgelehnt oder abgelaufen sind
- `ListChannelHandshakes`: Verfolge und zeige Handshakes mit Filterung nach Typ und Status

Mit Kanalbeziehungen arbeiten

Beziehungen ermöglichen es Ihnen, Endkunden, interne Organisationen oder nachgelagerte Verkäufer zu verwalten. Jede Beziehung beinhaltet:

- Das AWS Verwaltungskonto der zugehörigen AWS Organisation
- Erforderliche AWS Partnernetzwerk-Metadaten für die Qualifizierung von Kanalrabatten

Wenn Sie eine Beziehung einrichten, erhalten alle Nutzungsdaten der zugehörigen AWS Organisation die entsprechenden Vorteile des Channel-Programms. Verwenden Sie diese APIs, um Verkäufer und Endkunden zu melden:

- `CreateRelationship`: Integrieren Sie neue Kunden bei Endkunden, Vertriebsverkäufern oder internen Kunden AWS
- `GetRelationship`: Spezifische Beziehungsdetails anzeigen
- `ListRelationships`: Alle Beziehungen anzeigen
- `UpdateRelationship`: Ändern Sie bestehende Beziehungen
- `DeleteRelationship`: Beziehungen entfernen

Verwenden der Revenue Measurement API

Topics

- [Arbeiten Sie mit Ihren Umsatzzuordnungs-IDs](#)
- [Arbeiten Sie mit Ihren Marketplace Revenue Shares](#)

Arbeiten Sie mit Ihren Umsatzzuordnungs-IDs

Eine Umsatzzuordnungs-ID ist eine Kennung auf Angebotsebene, die Ihren AWS Marketplace-Produktumsatz bestimmten AWS Marketplace-Angeboten and/or AWS Partner Central-Opportunities zuordnet. Es baut auf der Implementierung von Partner Revenue Measurement (PRM) auf Produktebene auf: PRM erfasst den gesamten zugeschriebenen Umsatz für ein Marketplace-Produkt,

und eine Umsatzzuweisungs-ID ordnet diesen Umsatz bestimmten Deals entsprechend den von Ihnen angegebenen monatlichen Kostenzuweisungsprozentsätzen zu.

Der Revenue Attribution Service stellt APIs zur Verfügung, um Umsatzzuordnungs-IDs zu erstellen, abzurufen, aufzulisten und zu aktualisieren und ihre monatlichen Kostenzuordnungseinträge asynchron zu verwalten.

Was ist eine Umsatzzuordnungs-ID?

Eine Umsatzzuordnungs-ID besteht aus zwei Ebenen:

- Die Zuordnung einer vom Partner benannten, vom Partner beschriebenen Ressource, die durch eine `ra-` Präfix-ID (z. B.) und einen ARN identifiziert wird. `ra-aabbccdde001` Jede Zuordnung ist auf ein Konto `Catalog` (AWS für die Produktion, für Tests) und das Konto des Partners `Sandbox` beschränkt.
- Ein oder mehrere monatliche Kostenzuordnungseinträge, wobei jeder Eintrag eine einzelne Kombination (Angebots-ID oder Opportunity-ID, Abrechnungsmonat) einem Kostenzuweisungsprozentsatz zuordnet. Die Einträge werden stapelweise asynchron über ein Zuweisungsaufgabenmuster verwaltet.

Eine Umsatzzuordnungs-ID kann auf zwei Arten verwendet werden:

- Als Overlay auf Angebotsebene zu Ihrer bestehenden PRM-Implementierung auf Produktebene. Erstellen Sie eine Umsatzzuordnungs-ID, ordnen Sie Ihre entsprechenden Marketplace-Angebote and/or ACE-Opportunities zu und AWS ordnen Sie Ihren bereits gemessenen Produktumsatz diesen spezifischen Deals zu — es sind keine Änderungen an Ihren vorhandenen Ressourcen-Tags oder User-Agent-Strings erforderlich.
- Als eigenständige Kennung, die direkt als Ressourcen-Tag-Wert (`aws-apn-id=<RA ID>`) oder in einer User-Agent-Zeichenfolge (`APN_1.1/pc_<RA ID>$`) verwendet wird, um den AWS Verbrauch von Anfang an einem bestimmten Angebot zuzuordnen.

Arbeiten Sie mit Ihren Umsatzzuordnungs-IDs

Partner können Umsatzzuordnungs-IDs und ihre monatlichen Kostenzuordnungseinträge über den Revenue Attribution Service verwalten. Der Lebenszyklus durchläuft zwei unabhängige Abläufe: den Ablauf der Zuordnung von Datensätzen (Erstellen, Aktualisieren, Abrufen, Auflisten) und den Zuweisungsablauf (eine Batch-Aufgabe starten, Ergebnisse abfragen, einzelne Einträge abrufen, Einträge für eine Zuordnung auflisten).

Eine Umsatzzuordnungs-ID erstellen

Der erste Schritt besteht darin, mithilfe der `CreateRevenueAttribution` API-Aktion eine Umsatzzuordnungs-ID zu erstellen. Der zurückgegebene `Id` Wert `Arn` kann sofort als Resource-Tag-Wert oder in einer User-Agent-Zeichenfolge verwendet werden, um den Verbrauch zuzuordnen. AWS

Bei der Erstellung einer Umsatzzuordnungs-ID müssen Partner Folgendes angeben:

- **Catalog**— AWS für die Produktion oder Sandbox zum Testen.
- **Name**— ein für Menschen lesbarer Name für die Zuordnung. Muss innerhalb des Katalogs und im Konto des Partners eindeutig sein. Maximal 128 Zeichen.

Partner können optional Folgendes angeben:

- **Description**— Freitext-Beschreibung der Zuordnung. Maximal 1024 Zeichen.
- **MarketplaceProduct**— das AWS Marketplace-Produkt, das mit dieser Zuordnung verknüpft werden soll. Geben Sie `ProductIdentifier` (die 25-stellige Marketplace-Produkt-ID) und `TenancyModel` (`MULTI_TENANT` oder `SINGLE_TENANT`) an. Wenn bei der Erstellung nicht angegeben, gilt die Zuordnung für den gesamten Verbrauch, der unter der Umsatzzuordnungs-ID gemessen wird, unabhängig davon, welches Marketplace-Produkt der Kunde gekauft hat. Dies ist nützlich, wenn sich eine einzelne Bereitstellung über mehr als ein Marketplace-Angebot erstreckt.
- **Tags**— bis zu 200 Schlüssel-Wert-Paare für die Ressourcenorganisation. Tag-Schlüssel müssen innerhalb der Anfrage eindeutig sein.

Best Practice: Geben Sie `anMarketplaceProduct.ProductIdentifier`, wann immer Ihr Angebot in einem bestimmten Marketplace-Angebot verankert ist. Auf diese Weise kann AWS überprüft werden, ob das Produkt dem Konto des anrufenden Partners oder einem Tochterkonto gehört, das über die Partnerkontoverbindung (PAC) verbunden ist, und das gelöste `ProductCode` und `ProductType` (z. B., `SaaS`, `ML`) wird in der Antwort angezeigt. Wenn das Produkt keinem autorisierten Konto gehört, kehrt die `API ValidationException` mit gutem Grund zurück `PRODUCT_NOT_FOUND_OR_NOT_OWNED`.

Die Antwort gibt die neuen Attribute `Id` (Format `tra-[a-z0-9]{13}`) `Arn`, die aufgelösten `MarketplaceProduct` Attribute und die `Version` Anfangszahl (die bei 1 beginnt und bei jeder nachfolgenden Aktualisierung erhöht wird) zurück.

Hinzufügen von Einträgen zur monatlichen Kostenzuweisung

Sobald eine Umsatzzuordnungs-ID erstellt wurde, ordnen Partner ihr AWS Marketplace Offers and/or ACE Opportunities zu, indem sie über die `StartRevenueAttributionAllocationsTask` API-Aktion einen Stapel monatlicher Kostenzuordnungseinträge einreichen. Dies ist ein asynchroner Vorgang, der bis zu 250 Zuordnungsänderungen (CREATE and/or UPDATE) pro Aufgabe akzeptiert.

Jeder Zuordnungseintrag gibt Folgendes an:

- Angebots-ID oder Opportunity-ID — die eindeutige Kennung des verknüpften Deals. Wenn ein Marketplace-Angebot bereits mit einer ACE-Opportunity verknüpft ist, müssen Partner nur die Angebots-ID angeben. Der Umsatz AWS wird automatisch der verknüpften Opportunity zugewiesen.
- Abrechnungsmonat — der Kalendermonat, für den die Kostenzuweisung gilt (z. B. 2026-04).
- Kostenzuweisung in% — der Teil des AWS Gesamtverbrauchs des Produkts, der auf dieses Geschäft in diesem Abrechnungsmonat entfällt. Erforderlich für mehrinstanzenfähige SaaS-Produkte, einschließlich von Partnern gehosteter Komponenten für Hybridbereitstellungen, bei denen sich Kunden die Infrastruktur teilen.
- AWS Kundenkonto-ID — das AWS Konto des Kunden, der das Produkt konsumiert. Erforderlich für SaaS-Produkte mit mehreren Mandanten.

Die Gesamtkostenzuweisung in% aller Zuordnungseinträge für eine bestimmte Umsatzzuordnungs-ID und denselben Abrechnungsmonat darf 100% nicht überschreiten. Wenn die Summe 100% übersteigt, wird der fehlerhafte Eintrag während der asynchronen Geschäftsvalidierung mit einem Fehlercode pro Datensatz zurückgewiesen.

Synchrone Validierung: `StartRevenueAttributionAllocationsTask` Führt synchron die grundlegende Formvalidierung durch (Feldtypen, Muster, Chargengröße). Wenn die Standardvalidierung erfolgreich ist, gibt die API den Status `TaskId with` zurück. `IN_PROGRESS` Die Geschäftsvalidierung (Grenzwertprüfungen, Unveränderlichkeitsregeln, Abhängigkeitsabfragen anhand von Marketplace und ACE) wird asynchron ausgeführt, und die Ergebnisse pro Datensatz werden ermittelt. `GetRevenueAttributionAllocationsTask`

Um eine eingereichte Aufgabe zu überwachen, fragen Partner `GetRevenueAttributionAllocationsTask` anhand der Revenue Attribution Resource Id oder ARN ab. Die Antwort geht von `IN_PROGRESS` bis weiter `COMPLETED` (unabhängig davon, ob einzelne Datensätze erfolgreich waren oder nicht) und gibt Folgendes zurück:

- **TaskStatus**— IN_PROGRESSCOMPLETED, oder FAILED (Fehler bei der Aufgabe der obersten Ebene).
- **RecordResults**— für jeden Eingabedatensatz: der zugewiesene AllocationId (falls erfolgreich), der Status pro Datensatz (SUCCEEDEDoderFAILED) und ein strukturierter Fehlercode und eine Meldung, falls der Datensatz fehlgeschlagen ist.

Die Partner sollten die Aufgabenergebnisse umgehend abrufen. Nach Abschluss der Aufgabe können die Ergebnisse pro Datensatz abgeglichen werden. Fehlgeschlagene Eingaben können korrigiert und in einer neuen Aufgabe erneut eingereicht werden.

Bearbeiten von Einträgen zur monatlichen Kostenzuweisung

Einträge zur Kostenzuweisung werden pro Abrechnungsmonat nach folgenden Regeln verwaltet:

- Partner können jederzeit einen neuen monatlichen Eintrag für den aktuellen oder einen beliebigen future Abrechnungsmonat hinzufügen.
- Partner können einen monatlichen Eintrag für den aktuellen oder einen beliebigen future Abrechnungsmonat jederzeit aktualisieren.
- Partner können den Eintrag des letzten Monats bis zum 7. des aktuellen Monats aktualisieren. In diesem Fenster können Partner die tatsächliche Nutzung im AWS Cost Explorer überprüfen, bevor sie die Zuteilung für den Vormonat abschließen.

Nach dem 7. des aktuellen Monats kann die Zuordnung für den vorherigen Abrechnungsmonat nicht mehr geändert werden. Aktualisierungen des Eintrags eines aktuellen oder future Monats gelten ab dem nächsten monatlichen Abrechnungszeitraum. Die historische Zuordnung für frühere Monate wird nicht rückwirkend neu berechnet.

Aktualisierungen der Zuordnungseinträge werden über dieselbe StartRevenueAttributionAllocationsTask API-Aktion übermittelt, wobei für jeden betroffenen Eintrag die Operation Einstellung auf UPDATE gesetzt ist. Das asynchrone Aufgabenmuster verarbeitet Aktualisierungen und Kreationen einheitlich.

Aktualisierung einer Umsatzzuordnungs-ID

Partner können eine bestehende Umsatzzuordnungs-ID mithilfe der UpdateRevenueAttribution API-Aktion aktualisieren. Description Der Attributionsdatensatz selbst ist ansonsten unveränderlich — NameMarketplaceProduct, und Tag-on-Create-Werte können nach der

Erstellung nicht mehr geändert werden. Um die Ressource erneut zu taggen, verwenden Sie die AWS Standard-Tagging-Operationen für den ARN der Zuordnung.

Bei der Aktualisierung einer Umsatzzuordnungs-ID müssen Partner Folgendes angeben:

- **Catalog**— AWS oder Sandbox.
- **Identifier**— die ra- ID der zu aktualisierenden Zuordnung.
- **Version**— die aktuelle Version der Zuordnung für optimistisches Sperren.

Partner können optional Folgendes bereitstellen:

- **Description**— die aktualisierte Freitext-Beschreibung. Maximal 1024 Zeichen.
- **ClientToken**— ein Idempotenz-Token für das Update.

Optimistisches Sperren: Das `Version` Feld stellt sicher, dass Aktualisierungen nur gelten, wenn sich die Zuordnung seit dem letzten Abruf nicht geändert hat. Wenn die eingereichte `Version` Datei nicht mit der aktuellen Version der Ressource übereinstimmt, wird das Update mit einer Meldung, die die übermittelte `ConflictException` und die aktuelle Versionsnummer enthält, abgelehnt. Es empfiehlt sich, `GetRevenueAttribution` vor jedem Update die neueste Version mit abzurufen.

Details zur Umsatzzuordnungs-ID anzeigen

Partner können mithilfe der `GetRevenueAttribution` API-Aktion vollständige Informationen für eine einzelne Umsatzzuordnungs-ID abrufen. Das Ergebnis ist:

Metadaten der Ressource:

- Eindeutiger Bezeichner (`Id`) und Amazon-Ressourcenname (`Arn`).
- Der `Catalog` in dem die Zuordnung existiert.
- `Name` und `Description`.
- Die abgerufenen `Version` und die `LatestVersion` (verwenden Sie die neueste Version für nachfolgende Aktualisierungen).

Marketplace-Produktattribute (falls `MarketplaceProduct` bei der Erstellung festgelegt):

- `ProductId`— die vom Partner bereitgestellte Marketplace-Produkt-ID.

- **ProductCode**— der AWS Marketplace-Produktcode wurde von aufgelöst **ProductId**.
- **ProductType**— SaaS, AMI, ContainerML, Data, oder Professional Services.
- **TenancyModel**— MULTI_TENANT oder SINGLE_TENANT.

Informationen zur Prüfung:

- **CreatedDate** und **LastModifiedDate**.

Partner können **Version** in der Anfrage optional eine spezifische Angabe machen, um eine historische Version der Zuordnung abzurufen. Lassen Sie es aus **Version**, die neueste Version zurückzugeben.

Um einen bestimmten Eintrag für die monatliche Kostenzuweisung abzurufen, verwenden Partner die **GetRevenueAttributionAllocation** API-Aktion mit den Einträgen. **AllocationId** Dadurch werden die Angebots- oder Opportunity-ID, der Abrechnungsmonat, die Kostenzuweisung in%, die AWS Kundenkonto-ID sowie die Status- und Prüfungsinformationen des Eintrags zurückgegeben.

IDs zur Umsatzzuweisung werden aufgelistet

Partner können mithilfe der **ListRevenueAttributions** API-Aktion alle Umsatzzuordnungs-IDs in ihrem Konto einsehen. Dadurch wird eine paginierte Liste von Attributionszusammenfassungen mit Filter- und Sortierfunktionen zurückgegeben.

Partner können Ergebnisse nach folgenden Kriterien filtern:

- **Catalog**— AWS oder Sandbox (erforderlich).
- **Identifiers**— eine Liste mit bis zu 100 spezifischen **ra-** IDs, die abgerufen werden sollen.
- **CreatedAfter/CreatedBefore**— nach Zeitstempelbereich der Erstellung filtern (einschließlich).

Partner können die Ergebnissortierung mit dem folgenden **Sort** Parameter konfigurieren:

- **SortBy**— **CreatedDate** oder **LastModifiedDate**.
- **SortOrder**— **ASCENDING** oder **DESCENDING**.

Die Antwort enthält eine Zusammenfassung für jede Zuordnung: **Arn**, **IdCatalog**, **Name** aufgelöste **MarketplaceProduct** **Attribute**, **CreatedDate** **LastModifiedDate**, die letzten

und `TotalRevenueAttributionAssociationCount` (die Anzahl der aktiven monatlichen Kostenzuordnungseinträge `Version`, die mit der Zuordnung verknüpft sind).

Verwenden Sie `MaxResults` (Standard 25, maximal 100) und `NextToken` um große Ergebnismengen zu paginieren. Die Antwort beinhaltet, `NextToken` ob weitere Seiten verfügbar sind.

Auflistung der Einträge zur monatlichen Kostenzuweisung

Partner können mithilfe der `ListRevenueAttributionAllocations` API-Aktion die monatlichen Kostenzuordnungseinträge für eine bestimmte Umsatzzuordnungs-ID auflisten. Dadurch wird eine paginierte Liste von Zuordnungsübersichten mit Filterfunktionen zurückgegeben.

Partner können Ergebnisse nach folgenden Kriterien filtern:

- **MarketplaceOfferId**— listet nur Einträge auf, die mit einem bestimmten Marketplace-Angebot verknüpft sind.
- **AwsPartnerCentralOpportunityId**— listet nur Einträge auf, die mit einer bestimmten Partner Central-Opportunity verknüpft sind.
- **BillingMonth**— listet nur Einträge für einen bestimmten Kalendermonat auf.

Die Antwort enthält zusammenfassende Informationen für jeden Eintrag: die `AllocationId`, die zugehörige Angebots- oder Opportunity-ID, die AWS Kundenkonto-ID, den Abrechnungsmonat, die Kostenzuweisung in%, den Namen und Status des Eintrags sowie Zeitstempel für die Prüfung.

Erstellung von Dashboards: Die Kombination aus `ListRevenueAttributions` und `ListRevenueAttributionAllocations` soll die Erstellung von Partner-Dashboards unterstützen. Partner können Ansichten wie die folgenden erstellen:

- Alle Umsatzzuordnungs-IDs, die in den letzten 30 Tagen erstellt wurden, sortiert nach Erstellungsdatum.
- Alle monatlichen Kostenzuordnungseinträge für ein bestimmtes Marketplace-Angebot über mehrere Umsatzzuweisungs-IDs.
- Alle Kostenzuordnungseinträge für den aktuellen Abrechnungsmonat, um zu überprüfen, ob die Gesamtwerte pro Zuordnung 100% nicht überschreiten.
- Alle Umsatzzuordnungs-IDs, für die mindestens ein aktiver Umlageeintrag () vorhanden ist.
`TotalRevenueAttributionAssociationCount > 0`

Arbeiten Sie mit Ihren Marketplace Revenue Shares

Eine Marketplace-Umsatzbeteiligung ist eine Eingabe für ein AWS Marketplace-Produkt, die den Teil des Gesamtumsatzes des Produkts angibt, der über AWS Marketplace Marketplace.AWS verwendet diese Eingabe, um den Umsatz eines Partners mit anderen Marketplace-collected Umsatzsignalen zu korrelieren.

Der Marketplace Revenue Share Service stellt APIs zur Verfügung, um Marketplace Revenue Share Ressourcen zu erstellen, abzurufen und aufzulisten und deren Zuweisungen zu verwalten (die vom Partner angegebenen Umsatzbeteiligungsprozentsätze und Gültigkeitszeitfenster). Standardmäßige AWS Tagging-Operationen werden auf der Marketplace-Ressource Revenue Share unterstützt.

Was ist eine Marketplace-Umsatzbeteiligung?

Eine Marketplace-Umsatzbeteiligung besteht aus zwei Ebenen:

- Die Aktie — ein Input für ein einzelnes AWS Marketplace-Produkt, das vom Marketplace identifiziert wurde `ProductId`. Pro Produkt gibt es genau eine Marketplace-Umsatzbeteiligung.
- Eine oder mehrere Zuweisungen — für jede Zuteilung werden ein `RevenueSharePercent` und ein Gültigkeitsdatum angegeben (`EffectiveDate` optional `ExpirationDate`). Einer Aktie können im Laufe der Zeit viele Zuteilungen zugewiesen werden, aber zu jedem Zeitpunkt kann nur eine ACTIVE Zuteilung gelten.

Allokationen haben die folgenden wichtigsten Eigenschaften:

- Status — ACTIVE oder INACTIVE. Bei der Erstellung ist der Status standardmäßig auf ACTIVE. Bei Umsatzberechnungen und Überschneidungsprüfungen werden nur ACTIVE Umlagen berücksichtigt.
- Überschneidungsinvariant — innerhalb einer einzelnen Aktie dürfen sich die Bandbreiten nicht bei zwei ACTIVE Zuweisungen überschneiden. [`EffectiveDate`, `ExpirationDate`] Pro Aktie ist zu einem beliebigen Zeitpunkt höchstens eine unbefristete ACTIVE Zuteilung (kein `ExpirationDate`) zulässig.

Arbeiten Sie mit Ihren Marketplace Revenue Shares

Partner verwalten Marketplace Revenue Shares und deren Zuweisungen über den Marketplace Revenue Share Service. Der Lebenszyklus durchläuft zwei Abläufe: den Ablauf zum Teilen (Erstellen,

Abrufen, Auflisten, Markieren) und den Zuweisungsfluss (Erstellen, Aktualisieren, Löschen, Abrufen, Auflisten).

Schaffung einer Marktplatz-Umsatzbeteiligung

Der erste Schritt besteht darin, mithilfe der `CreateMarketplaceRevenueShare` API-Aktion eine Marketplace-Umsatzbeteiligung für ein AWS Marketplace-Produkt zu erstellen. Die Aktie wird vom Marketplace identifiziert `ProductId` und dient als Container für alle nachfolgenden Allokationen.

Bei der Erstellung einer Marketplace-Umsatzbeteiligung müssen Partner Folgendes angeben:

- **Catalog**— AWS für die Produktion oder Sandbox zum Testen.
- **ProductId**— die 25-stellige Marketplace-Produkt-ID. Muster: `[a-z0-9]{25}`.

Partner können optional Folgendes bereitstellen:

- **Tags**— bis zu 50 Schlüssel-Wert-Paare für die Organisation der Ressourcen zum Zeitpunkt der Erstellung. Wird verwendet `TagResource`, um Tags nach der Erstellung hinzuzufügen.
- **ClientToken**— ein Idempotenz-Token, um doppelte Kreationen beim erneuten Versuch zu verhindern. Maximal 64 Zeichen. SDKs generieren dieses Token automatisch.

Produktvalidierung: Bei der Erstellung verifiziert der Service bei AWS Marketplace, ob das Produkt dem Konto des Anrufers oder einem Tochterkonto gehört, das über Partner Account Connection (PAC) mit dem Anrufer verbunden ist. Wenn das Produkt nicht existiert oder das Eigentümerkonto für den Anrufer nicht autorisiert ist, kehrt die API mit gutem Grund zurück. `ValidationException` `PRODUCT_NOT_FOUND_OR_NOT_OWNED`

Eine Aktie pro Produkt: Ein zweiter `CreateMarketplaceRevenueShare` Aufruf für dasselbe Ergebnis führt `ProductId` zu denselben Catalog Ergebnissen. `ConflictException`

Die Antwort gibt die Aktie Arn (Formatarn: `{partition}:partnercentral:{region}:{account-id}:catalog/{catalog}/marketplace-revenue-share/{productId}`), dieCatalog, die ursprüngliche Version Zahl (die bei 1 beginnt und bei jeder Zuordnungsänderung inkrementiert wird) und alle bei der Erstellung verwendeten Tags zurück. `ProductId`

Zuordnungen werden hinzugefügt

Sobald eine Marketplace-Umsatzbeteiligung erstellt wurde, geben die Partner einen Prozentsatz der Umsatzbeteiligung und ein Gültigkeitsdatum an, indem sie über die `CreateMarketplaceRevenueShareAllocation` API-Aktion eine Zuordnung erstellen.

Bei der Erstellung einer Zuteilung müssen die Partner Folgendes angeben:

- **Catalog**— AWS oderSandbox.
- **MarketplaceRevenueShareIdentifier**— die Kennung der Mutteraktie.
- **RevenueSharePercent**— der Prozentsatz des über Marketplace erzielten Umsatzes des Produkts, angegeben als Dezimalzahl (z. B. 0.45 für 45%).
- **EffectiveDate**— das Kalenderdatum, an dem diese Zuordnung wirksam wird. Format:YYYY-MM-DD.

Die Antwort gibt den Status der Allokation Id (Format`mrsa-[A-Za-z0-9]{13}`), den Zuordnungsstatus (ACTIVEbei der Erstellung) und den Wert der übergeordneten Aktie zurück. `Version` (eine einzige Ganzzahl pro Aktie wird bei jeder Allokationsmutation erhöht, was für optimistische Sperren bei nachfolgenden Allokationsaktualisierungen verwendet wird).

Allokationen werden aktualisiert

Partner können eine Zuordnung mithilfe der `UpdateMarketplaceRevenueShareAllocation` API-Aktion aktualisieren. Die aktualisierbaren Felder hängen davon ab, ob es sich bei der Zuordnung um eine zukünftige, aktuell gültige oder vergangene Zuordnung handelt.

Bei der Aktualisierung einer Zuordnung müssen die Partner Folgendes angeben:

- **Catalog**— AWS oderSandbox.
- **MarketplaceRevenueShareIdentifier**— die Kennung der Mutteraktie.
- **AllocationId**— die `mrsa`- ID der zu aktualisierenden Zuordnung.
- **MarketplaceRevenueShareVersion**— die aktuelle Version der Mutteraktie für eine optimistische Festlegung der Allokation. Bump-Konflikte kehren zurück`ConflictException`.

Partner können optional Folgendes bereitstellen:

- **RevenueSharePercent**— den aktualisierten Prozentsatz der Umsatzbeteiligung. Nur für zukünftige Zuweisungen bearbeitbar.

- **EffectiveDate**— das aktualisierte Datum des Inkrafttretens. Nur für zukünftige Zuweisungen bearbeitbar.
- **ExpirationDate**— das aktualisierte Ablaufdatum. Bearbeitbar für zukünftige Zuweisungen (beliebiger Wert > EffectiveDate) und für aktuell gültige oder unbefristete Zuweisungen (beliebiger Wert). $\geq$ today
- **Status**— ACTIVE oder. INACTIVE Die Einstellung auf „INACTIVEVorläufiges Löschen“ löscht die Zuordnung. Nur für zukünftige Zuweisungen bearbeitbar.
- **ClientToken**— ein Idempotenz-Token.

Details zur Umsatzbeteiligung auf Marketplace anzeigen

Partner können mithilfe der GetMarketplaceRevenueShare API-Aktion vollständige Informationen für eine einzelne Marketplace-Umsatzbeteiligung abrufen. Dadurch werden die Arn,, CatalogProductId, Version (die aktuelle Version der übergeordneten Aktie),, CreatedDateLastModifiedDate, und alle auf die Aktie angewendeten Tags zurückgegeben.

Partner können eine einzelne Zuordnung mithilfe der GetMarketplaceRevenueShareAllocation API-Aktion mit der Zuordnung abrufenId. Das Ergebnis ist:

- Die Allokation Id und die Anteile der MuttergesellschaftMarketplaceRevenueShareArn.
- RevenueSharePercent, EffectiveDate, ExpirationDate.
- Status(ACTIVEoderINACTIVE).
- Die übergeordnete Aktie steht MarketplaceRevenueShareVersion für optimistisches Warten auf nachfolgende Updates.
- CreatedDate und LastModifiedDate.

Umsatzanteile von Marketplace auflisten

Partner können mithilfe der ListMarketplaceRevenueShares API-Aktion alle Marketplace Revenue Shares einsehen, die ihrem Konto gehören. Dadurch wird eine paginierte Liste mit Aktienübersichten zurückgegeben.

Partner können Ergebnisse nach folgenden Kriterien filtern:

- **Catalog**— AWS oder Sandbox (erforderlich).
- **ProductIds**— eine Liste bestimmter Marketplace-Produkt-IDs, die abgerufen werden sollen.

Die Antwort enthält eine Zusammenfassung für jede Aktie: `ArnCatalog`, `ProductId`, die neuesten `VersionCreatedDate`, und `LastModifiedDate`.

Verwenden Sie `MaxResults` und `NextToken`, um große Ergebnismengen zu paginieren.

Zuordnungen auflisten

Partner können mithilfe der `ListMarketplaceRevenueShareAllocations` API-Aktion die Zuteilungen für eine bestimmte Marketplace-Umsatzbeteiligung auflisten. Dadurch wird eine paginierte Liste von Allokationsübersichten mit Filterfunktionen zurückgegeben.

Partner können Ergebnisse nach folgenden Kriterien filtern:

- **Status**— `ACTIVE` oder `INACTIVE`.
- Gültigkeitszeitraum —`EffectiveDateOnOrAfter/EffectiveDateOnOrBefore`, um Zuordnungen abzurufen, deren Gültigkeitsdatum in ein bestimmtes Fenster fällt.

Die Antwort enthält zusammenfassende Informationen für jede Zuordnung: `Id`, `RevenueSharePercent`, `EffectiveDate`, `ExpirationDate`, `StatusMarketplaceRevenueShareVersion`, und Prüfzeitstempel.

Umsatzanteile von Marketplace taggen

Standardmäßige AWS Tagging-Operationen werden auf der Marketplace-Ressource Revenue Share unterstützt:

- **TagResource**— Hinzufügen oder Überschreiben von Tags auf einer Marketplace-Umsatzbeteiligung. Maximal 50 Tags pro Aktie.
- **UntagResource**— Entferne bestimmte Tags per Schlüssel.
- **ListTagsForResource**— ruft alle Tags ab, die derzeit auf eine Marketplace-Umsatzbeteiligung angewendet wurden.

Zuordnungen unterstützen keine Tags. Taggen Sie die übergeordnete Aktie, um zugehörige Zuordnungen zu organisieren.

Bei jedem Tagging-Vorgang wird das Format Arn (Formatarn: `{partition}:partnercentral:{region}:{account-id}:catalog/{catalog}/marketplace-revenue-share/{productId}`) des Share als Ressourcen-ID akzeptiert. Das standardmäßige AWS Tag-on-Create-Feld wird auch über das Feld `on` unterstützt. Tags `CreateMarketplaceRevenueShare`

Unterstützt AWS Regionen

Die folgenden Abschnitte enthalten Informationen zu den unterstützten AWS-Regionen.

Themen

- [Unterstützt AWS Regionen für AWS Konto-API für Partner Central](#)
- [Unterstützt AWS Regionen für AWS Vertriebs-API für Partner Central](#)
- [Unterstützt AWS Regionen für AWS API für Vorteile von Partner Central](#)
- [Unterstützt AWS Regionen für AWS Channel-API für Partner Central](#)
- [Unterstützt AWS Regionen für AWS API zur Umsatzmessung in Partner Central](#)

Unterstützt AWS Regionen für AWS Konto-API für Partner Central

Sie können von jeder AWS-Region USA Ost (Nord-Virginia) mit dem folgenden Endpunkt auf den AWS Partner Central-Kontoservice zugreifen.

```
partnercentral-account.us-east-1.api.aws
```

Unterstützt AWS Regionen für AWS Vertriebs-API für Partner Central

Sie können von jeder AWS-Region USA Ost (Nord-Virginia) mit dem folgenden Endpunkt auf den AWS Partner Central-Verkaufsservice zugreifen.

```
partnercentral-selling.us-east-1.api.aws
```

Unterstützt AWS Regionen für AWS API für Vorteile von Partner Central

Sie können von jeder AWS-Region USA Ost (Nord-Virginia) mit dem folgenden Endpunkt auf den AWS Partner Central-Vorteilsservice zugreifen.

```
partnercentral-benefits.us-east-1.api.aws
```

Unterstützt AWS Regionen für AWS Channel-API für Partner Central

Sie können von jeder AWS-Handelsregion aus auf den AWS Partner Central Channel Management Service mit dem folgenden Endpunkt zugreifen.

```
partnercentral-channel.global.api.aws
```

Signieren von Anfragen mit SigV4a

Da die AWS Partner Central Channel API einen globalen Endpunkt verwendet, werden Anfragen mit Signature Version 4a (SigV4a) signiert, einer Erweiterung von Sigv4, die das Signieren mit asymmetrischer Kryptografie unterstützt und Signaturen ermöglicht, die in mehreren Regionen verifiziert werden können. AWS

Verwenden des AWS SDK oder der CLI

Wenn Sie ein AWS SDK oder die AWS CLI verwenden, wird die SigV4A-Signatur automatisch verarbeitet, wenn Sie den globalen Endpunkt aufrufen. Es ist keine zusätzliche Konfiguration erforderlich.

Manuelles Signieren von Anfragen

Wenn Sie Anfragen manuell ohne SDK signieren, beachten Sie die folgenden Unterschiede zum Standard-SigV4-Standard:

- SigV4a leitet ein ECDSA-Schlüsselpaar (Elliptic Curve Digital Signature Algorithm) aus Ihrem vorhandenen AWS geheimen Zugriffsschlüssel ab, anstatt die Schlüsselableitung zu verwenden. HMAC-based
- Im Gültigkeitsbereich der Anmeldeinformationen wird * für die Regionskomponente anstelle eines bestimmten Regionsnamens verwendet, da die Signatur regionsübergreifend gültig ist.

Weitere Informationen zur Funktionsweise von SigV4a finden Sie unter [AWS Signature Version 4 für API-Anfragen](#). [Codebeispiele für die Sigv4A-Signatur finden Sie im Projekt sigv4a-signing-examples unter GitHub](#)

Unterstützt AWS Regionen für AWS API zur Umsatzmessung in Partner Central

Sie können von jeder AWS-Handelsregion aus auf den AWS Partner Central Revenue Measurement Service mit dem folgenden Endpunkt zugreifen.

```
partnercentral-prm.global.api.aws
```

Signieren von Anfragen mit SigV4a

Da der Service zur Umsatzmessung einen globalen Endpunkt verwendet, werden Anfragen mit Signature Version 4a (SigV4a) signiert, einer Erweiterung von Sigv4, die das Signieren mit asymmetrischer Kryptografie unterstützt und Signaturen ermöglicht, die in mehreren Regionen verifiziert werden können. AWS

Verwenden des AWS SDK oder der CLI

Wenn Sie ein AWS SDK oder die AWS CLI verwenden, wird die SigV4A-Signatur automatisch verarbeitet, wenn Sie den globalen Endpunkt aufrufen. Es ist keine zusätzliche Konfiguration erforderlich.

Manuelles Signieren von Anfragen

Wenn Sie Anfragen manuell ohne SDK signieren, beachten Sie die folgenden Unterschiede zum Standard-SigV4-Standard:

- SigV4a leitet ein ECDSA-Schlüsselpaar (Elliptic Curve Digital Signature Algorithm) aus Ihrem vorhandenen AWS geheimen Zugriffsschlüssel ab, anstatt die Schlüsselableitung zu verwenden. HMAC-based
- Im Gültigkeitsbereich der Anmeldeinformationen wird * für die Regionskomponente anstelle eines bestimmten Regionsnamens verwendet, da die Signatur regionsübergreifend gültig ist.

Weitere Informationen zur Funktionsweise von SigV4a finden Sie unter [AWS Signature Version 4 für API-Anfragen](#). [Codebeispiele für die Sigv4A-Signatur finden Sie im Projekt sigv4a-signing-examples unter](#). [GitHub](#)

Berechtigungen

Die folgenden Abschnitte enthalten Informationen zu Berechtigungen.

Themen

- [Zugriff für die Account-API](#)
- [Zugriff für die Verkaufs-API](#)
- [Zugang für die Benefits API](#)
- [Zugriff für die Channel-API](#)
- [Zugriff für die Revenue Measurement API](#)

Zugriff für die Account-API

Zugriffskontrolle und Berechtigungen werden von AWS Identity and Access Management(IAM) verwaltet. Dieser Abschnitt enthält Anleitungen zur Konfiguration der erforderlichen Berechtigungen für die Interaktion mit der Account-API.

Voraussetzungen

Stellen Sie vor der Konfiguration von Berechtigungen sicher, dass Ihre mit Ihren IAM-Rollen und -Benutzern verknüpft AWS-Konto ist und dass Sie die erforderlichen IAM-Rollen und -Benutzer erstellt haben. Weitere Informationen finden Sie unter [Einrichtung und Authentifizierung](#).

Verwenden AWS Verwaltete Richtlinien

AWS stellt verwaltete Richtlinien bereit, die die erforderlichen Berechtigungen für die Interaktion mit der Konto-API gewähren. Um den erforderlichen Zugriff für die Verwaltung der Kontoressourcen bereitzustellen, fügen Sie die `AWSPartnerCentralFullAccess` Richtlinie Ihren IAM-Identitäten hinzu. Weitere Informationen finden Sie unter [AWS Verwaltete Richtlinien für Benutzer](#).

Zuweisen von Richtlinien zu IAM-Rollen und -Benutzern

Gehen Sie wie folgt vor, um IAM-Rollen und -Benutzern Richtlinien zuzuweisen:

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Navigieren Sie zum IAM-Dienst.
3. Wählen Sie Rollen oder Benutzer aus und wählen Sie die IAM-Rolle oder den IAM-Benutzer aus, dem Sie eine Richtlinie zuordnen möchten.
4. Ordnen Sie die `AWSPartnerCentralFullAccess` Richtlinie der ausgewählten IAM-Rolle oder dem ausgewählten IAM-Benutzer zu.

Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Verwaltung von Berechtigungen mithilfe von Bedingungsschlüsseln

Bedingungsschlüssel in IAM-Richtlinien bieten Berechtigungen auf Ressourcenebene, mit denen festgelegt werden kann, wann Richtlinien für Aussagen durchgesetzt werden müssen. Sie können Bedingungsschlüssel verwenden, um Bedingungen festzulegen, die vorschreiben, wann bestimmte Berechtigungen erlaubt oder verweigert werden.

Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingungs-Operatoren](#).

Übersicht über die Bedingungsschlüssel

Bedingungsschlüssel	Description	Anwendbare Aktionen	Zulässige Werte
Partner Central: Katalog	filtert den Zugriff nach dem Typ der zugehörigen Katalogeigenschaft	alle Aktionen	AWS, Sandkasten

Zusammenfassung der erforderlichen Berechtigungen

Zusammenfassung der erforderlichen Berechtigungen

Action	Description
Partnercentral: AcceptConnectionInvitation	ermöglicht das Annehmen von Verbindungseinladungen
Partnercentral: AssociateAwsTrainingCertificationEmailDomain	ermöglicht die Verknüpfung von E-Mail-Domains AWS für Schulungszertifizierungen
Partnercentral: CancelConnection	ermöglicht das Abbrechen von Verbindungen
Partnercentral: CancelConnectionInvitation	ermöglicht das Stornieren von Verbindungseinladungen
Partnercentral: CancelProfileUpdateTask	ermöglicht das Abbrechen von Aufgaben zur Profilaktualisierung
Partnercentral: CreateConnectionInvitation	ermöglicht das Erstellen von Verbindungseinladungen
Partnercentral: CreatePartner	ermöglicht das Erstellen von Partnern
Partnercentral: DisassociateAwsTrainingCertificationEmailDomain	ermöglicht die Trennung von E-Mail-Domains für AWS Schulungszertifikate
Partnercentral: GetAllianceLeadContact	ermöglicht das Abrufen der Kontaktdaten von Allianz-Leads

Action	Description
Partnercentral: GetConnection	ermöglicht das Abrufen von Verbindungsdetails
Partnercentral: GetConnectionInvitation	ermöglicht das Abrufen von Verbindungseinladungsdetails
Partnercentral: GetConnectionPreferences	ermöglicht das Abrufen von Verbindungseinstellungen
Partnercentral: GetPartner	ermöglicht das Abrufen von Partnerdetails
Partnercentral: GetProfileUpdateTask	ermöglicht das Abrufen von Aufgabendetails zur Profilaktualisierung
Partnercentral: GetProfileVisibility	ermöglicht das Abrufen der Einstellungen für die Profilsichtbarkeit
Partnercentral: GetVerification	ermöglicht das Abrufen von Bestätigungsdetails
Partnercentral: ListConnectionInvitations	ermöglicht das Auflisten von Verbindungseinladungen
Partnercentral: ListConnections	ermöglicht das Auflisten von Verbindungen
Partnercentral: ListPartners	ermöglicht das Auflisten von Partnern
Partnercentral: PutAllianceLeadContact	ermöglicht die Aktualisierung der Kontaktdaten des Allianzleiters
Partnercentral: PutProfileVisibility	ermöglicht die Aktualisierung der Sichtbarkeitseinstellungen des Profils
Partnercentral: RejectConnectionInvitation	ermöglicht das Ablehnen von Verbindungseinladungen
Partnercentral: SendEmailVerificationCode	ermöglicht das Senden von Bestätigungs_codes per E-Mail

Action	Description
Partnercentral: StartProfileUpdateTask	ermöglicht das Starten von Aufgaben zur Profilaktualisierung
Partnercentral: StartVerification	ermöglicht das Starten von Verifizierungsprozessen
Partnercentral: UpdateConnectionPreferences	ermöglicht die Aktualisierung der Verbindungseinstellungen

Zugriff für die Verkaufs-API

Zugriffskontrolle und Berechtigungen werden von AWS Identity and Access Management(IAM) verwaltet. Dieser Abschnitt enthält Anleitungen zur Konfiguration der erforderlichen Berechtigungen für die Interaktion mit der API, einschließlich der Berechtigungen, die zum Auflisten von AWS Marketplace Entitäten erforderlich sind.

Voraussetzungen

Stellen Sie vor der Konfiguration von Berechtigungen sicher, dass Ihre mit Partner Central verknüpft AWS-Konto ist und dass Sie die erforderlichen IAM-Rollen und -Benutzer erstellt haben. Weitere Informationen finden Sie unter [Einrichtung und Authentifizierung](#).

Verwenden AWS Verwaltete Richtlinien

AWS stellt verwaltete Richtlinien bereit, die die erforderlichen Berechtigungen für die Interaktion mit der API gewähren. Um den erforderlichen Zugriff für die Verwaltung von Opportunities bereitzustellen, fügen Sie die `AWSPartnerCentralOpportunityManagement` Richtlinie Ihren IAM-Identitäten hinzu. Weitere Informationen finden Sie unter [AWS Verwaltete Richtlinien für AWS Partner Central-Benutzer](#).

AWSPartnerCentralOpportunityManagement Richtlinie

Diese Richtlinie gewährt vollen Zugriff auf die Opportunity-Management-Aktionen von 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/*"
        ]
    }
]
}

```

Benutzerdefinierte Richtlinien

Wenn die verwalteten Richtlinien nicht Ihren Anforderungen entsprechen, erstellen Sie benutzerdefinierte IAM-Richtlinien, die die für Ihren Anwendungsfall erforderlichen Berechtigungen gewähren. Das folgende Beispiel ist eine benutzerdefinierte Richtlinie, die Berechtigungen zum Auflisten von AWS Marketplace Entitäten gewährt:

Example Beispiel für eine benutzerdefinierte Richtlinie

JSON

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "partnercentral:ListOpportunities",
                "aws-marketplace:ListEntities"
            ],

```

```

        "Resource": "*"
    }
]
}

```

Benutzerdefinierte permissive Richtlinie

Diese Richtlinie bietet umfassenden Zugriff auf die Verkaufsaktionen von Partner Central, einschließlich Funktionen, die in future hinzugefügt werden können, ohne dass die Richtlinien aktualisiert werden müssen. Durch die Verwendung der Wildcard-Aktion `partnercentral:*` gewährt diese Richtlinie automatisch Zugriff auf neue Verkaufsfunktionen von Partner Central, sobald diese verfügbar sind, wodurch der Bedarf an manuellen Updates reduziert wird. Diese Richtlinie umfasst auch Berechtigungen für die Interaktion mit AWS Marketplace Entitäten, wodurch sichergestellt wird, dass der Zugriff sowohl für Verkaufs- als auch für Marketplace-Aktionen gewährt bleibt.

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"
        }
    }
}
]
```

Zuweisen von Richtlinien zu IAM-Rollen und -Benutzern

Gehen Sie wie folgt vor, um IAM-Rollen und -Benutzern Richtlinien zuzuweisen:

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Navigieren Sie zum IAM-Dienst.
3. Wählen Sie Rollen oder Benutzer aus und wählen Sie die IAM-Rolle oder den IAM-Benutzer aus, dem Sie eine Richtlinie zuordnen möchten.
4. Ordnen Sie die `AWSPartnerCentralOpportunityManagement` Richtlinie oder Ihre benutzerdefinierte Richtlinie der ausgewählten IAM-Rolle oder dem ausgewählten IAM-Benutzer zu.

Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Verwaltung von Berechtigungen mithilfe von Bedingungsschlüsseln

Bedingungsschlüssel in IAM-Richtlinien bieten Berechtigungen auf Ressourcenebene, mit denen festgelegt werden kann, wann Richtlinien für Aussagen durchgesetzt werden müssen. Sie können Bedingungsschlüssel verwenden, um Bedingungen festzulegen, die vorschreiben, wann bestimmte Berechtigungen erlaubt oder verweigert werden.

Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingungs-Operatoren](#).

Übersicht über die Bedingungsschlüssel

Bedingungsschlüssel	Description	Anwendbare Aktionen	Zulässige Werte
Partner Central: Katalog	filtert den Zugriff nach dem Typ der zugehörigen Katalogeinheit	alle Aktionen	AWS, Sandkasten
AWS-Marktplatz: PartyType	filtert den Zugriff nach der Art der Partei (z. B. Antragsteller)	SearchAgreements, DescribeAgreement	Vorschlagsgeber

Zusammenfassung der erforderlichen Genehmigungen

Zusammenfassung der erforderlichen Berechtigungen

Action	Description
Partnercentral: CreateOpportunity	ermöglicht die Schaffung von Möglichkeiten
Partnercentral: UpdateOpportunity	ermöglicht die Aktualisierung von Möglichkeiten
Partnercentral: ListOpportunities	ermöglicht das Auflisten von Möglichkeiten
Partnercentral: GetOpportunity	ermöglicht das Abrufen von Opportunity-Details
Partnercentral: ListSolutions	ermöglicht das Auflisten von Lösungen
Partnercentral: AssociateOpportunity	ermöglicht die Verknüpfung von Opportunities mit anderen Entitäten
Partnercentral: DisassociateOpportunity	ermöglicht es, Opportunities von anderen Entitäten zu trennen
Partnercentral: AcceptEngagementInvitation	ermöglicht das Annehmen von Einladungen zur Teilnahme

Action	Description
Partnercentral: RejectEngagementInvitation	ermöglicht das Ablehnen von Einladungen zur Verlobung
Partnercentral: GetEngagementInvitation	ermöglicht das Abrufen von Details zur Einladung zur Teilnahme
Partnercentral: ListEngagementInvitations	ermöglicht das Auflisten von Einladungen zur Teilnahme
Partnercentral: SubmitOpportunity	ermöglicht das Einreichen von Möglichkeiten
Partnercentral: GetAwsOpportunitySummary	ermöglicht das Abrufen AWS einer Zusammenfassung der Geschäftschancen
Partnercentral: StartProspectingFromEngagementTask	Erstellt anhand von Engagements eine Aufgabe zur Kundenakquise
Partnercentral: GetProspectingFromEngagementTask	Gibt Details zur Akquise-Aufgabe zurück
Partnercentral: ListProspectingFromEngagementTasks	Gibt eine Liste von Aufgaben zur Kundenakquise zurück
aws-marketplace: ListEntities	ermöglicht das Auflisten von Entitäten AWS Marketplace
aws-marketplace: DescribeEntity	ermöglicht die Beschreibung von Entitäten AWS Marketplace
aws-marketplace: SearchAgreements	ermöglicht die Suche nach Vereinbarungen in AWS Marketplace
aws-marketplace: DescribeAgreement	ermöglicht die Beschreibung von Vereinbarungen in AWS Marketplace

Zugang für die Benefits API

Zugriffskontrolle und Berechtigungen werden von AWS Identity and Access Management(IAM) verwaltet. Dieser Abschnitt enthält Anleitungen zur Konfiguration der erforderlichen Berechtigungen für die Interaktion mit der Benefits API.

Voraussetzungen

Stellen Sie vor der Konfiguration von Berechtigungen sicher, dass Ihre mit Ihren IAM-Rollen und -Benutzern verknüpft AWS-Konto ist und dass Sie die erforderlichen IAM-Rollen und -Benutzer erstellt haben. Weitere Informationen finden Sie unter [Einrichtung und Authentifizierung](#).

Verwenden AWS Verwaltete Richtlinien

AWS stellt verwaltete Richtlinien bereit, die die erforderlichen Berechtigungen für die Interaktion mit der Benefits API gewähren. Um den erforderlichen Zugriff für die Verwaltung der Leistungsressourcen bereitzustellen, fügen Sie die `AWSPartnerCentralFullAccess` Richtlinie Ihren IAM-Identitäten hinzu. Weitere Informationen finden Sie unter [AWS Verwaltete Richtlinien für Benutzer](#).

Zuweisen von Richtlinien zu IAM-Rollen und -Benutzern

Gehen Sie wie folgt vor, um IAM-Rollen und -Benutzern Richtlinien zuzuweisen:

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Navigieren Sie zum IAM-Dienst.
3. Wählen Sie Rollen oder Benutzer aus und wählen Sie die IAM-Rolle oder den IAM-Benutzer aus, dem Sie eine Richtlinie zuordnen möchten.
4. Ordnen Sie die `AWSPartnerCentralFullAccess` Richtlinie der ausgewählten IAM-Rolle oder dem ausgewählten IAM-Benutzer zu.

Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Verwaltung von Berechtigungen mithilfe von Bedingungsschlüsseln

Bedingungsschlüssel in IAM-Richtlinien bieten Berechtigungen auf Ressourcenebene, mit denen festgelegt werden kann, wann Richtlinien für Aussagen durchgesetzt werden müssen. Sie können Bedingungsschlüssel verwenden, um Bedingungen festzulegen, die vorschreiben, wann bestimmte Berechtigungen erlaubt oder verweigert werden.

Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingungs-Operatoren](#).

Übersicht über die Bedingungsschlüssel

Bedingungsschlüssel	Description	Anwendbare Aktionen	Zulässige Werte
Partner Central: Katalog	filtert den Zugriff nach dem Typ der zugehörigen Katalogeinheit	alle Aktionen	AWS, Sandkasten

Zusammenfassung der erforderlichen Berechtigungen

Zusammenfassung der erforderlichen Berechtigungen

Action	Description
Partnercentral: AmendBenefitApplication	ermöglicht die Änderung von Leistungsanträgen
Partnercentral: AssociateBenefitApplicationResource	ermöglicht die Zuordnung von Ressourcen zu Leistungsanträgen
Partnercentral: CancelBenefitApplication	ermöglicht die Stornierung von Leistungsanträgen
Partnercentral: CreateBenefitApplication	ermöglicht die Erstellung von Leistungsanträgen
Partnercentral: DisassociateBenefitApplicationResource	ermöglicht die Trennung von Ressourcen und Leistungsanträgen
Partnercentral: GetBenefit	ermöglicht das Abrufen von Leistungsdetails
Partnercentral: GetBenefitAllocation	ermöglicht das Abrufen von Details zur Leistungszuweisung
Partnercentral: GetBenefitApplication	ermöglicht das Abrufen von Details zur Beantragung von Leistungen
Partnercentral: ListBenefitAllocations	ermöglicht die Auflistung von Leistungszuweisungen

Action	Description
Partnercentral: ListBenefitApplications	ermöglicht die Auflistung von Leistungsanträgen
Partnercentral: ListBenefits	ermöglicht das Auflisten von Leistungen
Partnercentral: RecallBenefitApplication	ermöglicht den Rückruf von Leistungsanträgen
Partnercentral: SubmitBenefitApplication	ermöglicht das Einreichen von Leistungsanträgen
Partnercentral: UpdateBenefitApplication	ermöglicht die Aktualisierung von Leistungsanträgen

Zugriff für die Channel-API

Zugriffskontrolle und Berechtigungen werden von AWS Identity and Access Management(IAM) verwaltet. Dieser Abschnitt enthält Anleitungen zur Konfiguration der erforderlichen Berechtigungen für die Interaktion mit der AWS Partner Central Channel API.

Einrichtung des Channel-Management-Kontos

Für die Channel-Management-Funktionen in AWS Partner Central müssen IAM-Rollen in zwei Kontotypen AWS bereitgestellt werden:

1. [AWS Mit Partner Central verknüpft AWS-Konto:](#)

Dies ist mit Ihrem AWS Partner Network- und Partner Central-Konto AWS-Konto verknüpft. AWS-Konto Pro Partner ist nur ein AWS Partner Central verknüpft. Als einmalige Einrichtungsaktivität erstellen Sie dabei eine IAM-Rolle AWS-Konto mithilfe einer von Partner Central verwalteten Channel-Richtlinie.

2. [Konto AWS-Konto für die Programmverwaltung:](#)

Dies wird als Bill-Transfer Konto AWS-Konto zur Verwaltung der Abrechnung und Bezahlung für den Endkundenverbrauch verwendet. Dies AWS-Konto wird in der Kanalverwaltung von AWS Partner Central als Programmverwaltungskonto gemeldet. Möglicherweise haben Sie mehrere Programmverwaltungskonten, und Sie müssen für jedes Konto, das AWS-Konto Sie als Programmverwaltungskonto angeben, eine IAM-Rolle erstellen.

Für jeden AWS-Konto Typ müssen Sie verschiedene IAM-Rollen bestimmten Berechtigungs- und Vertrauensrichtlinien zuordnen.

Zugriff für AWS Partner Central verlinkt AWS-Konto

Erstellen Sie im AWS-Konto verknüpften AWS Partner Central eine IAM-Rolle zur Verwaltung der Kanalressourcen, die Partnern Central-Benutzern zugewiesen wird. Diese IAM-Rolle ermöglicht es Benutzern, Channel-Management-Ressourcen in AWS Partner Central einzusehen und zu verwalten. Der Rollename muss mit `PartnerCentralRoleFor` beginnen, und der empfohlene Name ist `PartnerCentralRoleForChannel`. Gehen Sie wie folgt vor, um die Rolle zu konfigurieren:

- Fügen Sie die folgende benutzerdefinierte Vertrauensrichtlinie hinzu, damit Cloud-Administratoren und Allianzleiter von AWS Partner Central [IAM-Rollen Partnern Central-Benutzern zuordnen können](#):

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

- Ordnen Sie die [AWSPartnerCentralChannelManagement](#) verwaltete Richtlinie der `PartnerCentralRoleForChannel` Rolle zu.

Zugriff für das Programmverwaltungskonto AWS Konto (e)

Erstellen Sie für jedes Konto, das in AWS Partner Central als Programmverwaltungskonto AWS-Konto gemeldet wird, die folgenden IAM-Rollen mit den folgenden erforderlichen Namen:

- `PartnerCentralChannelHandshakeApprovalManagement`: Verwenden Sie diese Rolle, um Handshakes zwischen Partner Central und Ihrem Programmmanagement zu verwalten. AWS-Konto Diese IAM-Rolle kann verwendet werden, um auf die Aktivierung Ihrer Programmverwaltungskonten in AWS Partner Central zu reagieren. Ordnen Sie bei der Erstellung

dieser Rolle die `AWSPartnerCentralChannelHandshakeApprovalManagement` verwaltete Richtlinie zu.

- `PartnerCentralChannelBillingTransferReadOnly`: Verwenden Sie diese kontoübergreifende Rolle, um den Status der Rechnungsüberweisung in AWS Partner Central sichtbar zu machen. In dieser Rolle wird der Status der Rechnungsübertragung von Ihnen AWS-Konto an AWS Partner Central weitergegeben, sodass Sie den Status der Rechnungsübertragung zentral einsehen können. Wenn diese Rolle erstellt wurde, können Benutzer in AWS Partner Central den Status der Rechnungsübertragung in AWS Partner Central einsehen. Diese Rolle gewährt AWS Partnern Central-Benutzern jedoch nicht die Berechtigung, auf Abrechnungstransfers in der AWS Billing and Cost Management-Konsole zuzugreifen.
- Vertrauensrichtlinie:
 - Ersetzen Sie in der folgenden JSON-Datei `{PartnerCentralLinkedAccountId}` bei der Erstellung der Rolle die durch Ihre tatsächliche mit AWS Partner Central verknüpfte AWS-Konto 12-stellige ID.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{PartnerCentralLinkedAccountId}:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Genehmigungsrichtlinie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BillingTransferReadOnly",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeResponsibilityTransfer",
        "organizations:ListHandshakesForOrganization",
        "organizations:ListInboundResponsibilityTransfers",

```

```

        "organizations:ListOutboundResponsibilityTransfers"
    ],
    "Resource": "*"
}
]
}

```

- **PartnerCentralChannelBillingTransferManagement(optional):** Verwenden Sie diese kontenübergreifende Rolle, um die Erstellung und Verwaltung von Rechnungsüberweisungen von AWS Partner Central aus zu ermöglichen. Mit dieser Rolle können AWS Partner Central-Benutzer auf die IAM-Rolle zugreifen, die `AWSPartnerCentralChannelManagement` verwaltete Richtlinien enthält, um Rechnungsübertragungen im Programmverwaltungs-konto zu verwalten. AWS-Konto Mit dieser kontenübergreifenden Rolle können AWS Partner Central-Benutzer in der Billing and Cost Management-Konsole über die Schaltfläche „In AWS Abrechnung verwalten“ in der AWS Partner Central-Channelverwaltung auf AWS Abrechnungstransfers zugreifen und diese verwalten.
- **Vertrauensrichtlinie:**
 - Ersetzen Sie in der folgenden JSON-Datei `{PartnerCentralLinkedAccountId}` bei der Erstellung der Rolle die durch Ihre tatsächliche mit AWS Partner Central verknüpfte AWS-Konto 12-stellige ID.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{PartnerCentralLinkedAccountId}:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

- **Genehmigungsrichtlinie:**

```

{
  "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": "*"
  }
]
}

```

Verwenden AWS Verwaltete Richtlinien

AWS bietet verwaltete Richtlinien, die die erforderlichen Berechtigungen für die Interaktion mit der Channel-API gewähren. Um den erforderlichen Zugriff für die Verwaltung aller Kanalressourcen bereitzustellen, fügen Sie die `AWSPartnerCentralChannelManagement` Richtlinie Ihren IAM-Identitäten hinzu. Fügen Sie die `AWSPartnerCentralChannelHandshakeApprovalManagement` Richtlinie Ihren IAM-Identitäten bei, um den erforderlichen Zugriff zum Anzeigen und Beantworten von Channel-Handshake-Anfragen zu gewähren. Weitere Informationen finden Sie unter [AWS Verwaltete Richtlinien für AWS Partner Central](#)-Benutzer.

AWSPartnerCentralChannelManagement Richtlinie

Diese Richtlinie gewährt vollen Zugriff auf die Channel-Management-Aktionen von 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 Richtlinie

Diese Richtlinie gewährt Zugriff auf das Ansehen und Beantworten von Channel-Handshake-Anfragen. Diese Richtlinie sollte auf Rollen in den AWS Konten angewendet werden, die Channel-Handshake-Anfragen erhalten.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ChannelHandshakeManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:ListChannelHandshakes",
        "partnercentral:AcceptChannelHandshake",
        "partnercentral:RejectChannelHandshake"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "AWS",
            "Sandbox"
          ]
        }
      }
    }
  ]
}

```

```
    }  
  }  
]  
}
```

Zuweisen von Richtlinien zu IAM-Rollen und -Benutzern

Gehen Sie wie folgt vor, um IAM-Rollen und -Benutzern Richtlinien zuzuweisen:

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Navigieren Sie zum IAM-Dienst.
3. Wählen Sie Rollen oder Benutzer aus und wählen Sie die IAM-Rolle oder den IAM-Benutzer aus, dem Sie eine Richtlinie zuordnen möchten.
4. Ordnen Sie die `AWSPartnerCentralChannelManagement` Richtlinie oder Ihre benutzerdefinierte Richtlinie der ausgewählten IAM-Rolle oder dem ausgewählten IAM-Benutzer zu.

Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Verwaltung von Berechtigungen mithilfe von Bedingungsschlüsseln

Bedingungsschlüssel in IAM-Richtlinien bieten Berechtigungen auf Ressourcenebene, mit denen festgelegt werden kann, wann Richtlinien für Kontoauszüge durchgesetzt werden müssen. Sie können Bedingungsschlüssel verwenden, um Bedingungen festzulegen, die vorschreiben, wann bestimmte Berechtigungen erlaubt oder verweigert werden.

Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingungs-Operatoren](#).

Übersicht über die Bedingungsschlüssel

Bedingungsschlüssel	Description	Anwendbare Aktionen	Zulässige Werte
Partner Central: Katalog	Filtert den Zugriff nach dem Typ der zugehörigen Katalogeinheit	Alle Channel-Management-Aktionen	AWS, Sandkasten
Partnerzentrale: ChannelHandshakeType	Filtert den Zugriff basierend auf der	CreateChannelHandshake, AcceptChannelHandshake,	PROGRAM_MANAGEMENT_ACCOUNT,

Bedingungsschlüssel	Description	Anwendbare Aktionen	Zulässige Werte
	Art des Kanal-Handshakes	RejectChannelHandshake, CancelChannelHandshake, ListChannelHandshakes	START_SERVICE_PERIOD, REVOKE_SERVICE_PERIOD

Zusammenfassung der erforderlichen Berechtigungen

Zusammenfassung der erforderlichen Berechtigungen

Action	Description
Partnercentral: CreateProgramManagementAccount	ermöglicht das Erstellen von Konten für die Programmverwaltung
Partnercentral: UpdateProgramManagementAccount	ermöglicht die Aktualisierung von Konten für die Programmverwaltung
Partnercentral: DeleteProgramManagementAccount	ermöglicht das Löschen von Konten für die Programmverwaltung
Partnercentral: ListProgramManagementAccounts	ermöglicht das Auflisten von Konten für die Programmverwaltung
Partnercentral: GetProgramManagementAccount	ermöglicht das Abrufen von Kontodetails für die Programmverwaltung
Partnercentral: CreateRelationship	ermöglicht das Erstellen von Beziehungen
Partnercentral: UpdateRelationship	ermöglicht die Aktualisierung von Beziehungen
Partnercentral: DeleteRelationship	ermöglicht das Löschen von Beziehungen
Partnercentral: GetRelationship	ermöglicht das Abrufen von Beziehungsdetails
Partnercentral: ListRelationships	ermöglicht das Auflisten von Beziehungen

Action	Description
Partnercentral: CreateChannelHandshake	ermöglicht das Erstellen von Channel-Handshakes
Partnercentral: AcceptChannelHandshake	ermöglicht das Akzeptieren von Kanal-Handshakes
Partnercentral: RejectChannelHandshake	ermöglicht das Ablehnen von Kanal-Handshakes
Partnercentral: CancelChannelHandshake	ermöglicht das Stornieren von Kanal-Handshakes
Partnercentral: ListChannelHandshakes	ermöglicht das Auflisten von Channel-Handshakes

Zugriff für die Revenue Measurement API

Zugriffskontrolle und Berechtigungen werden von AWS Identity and Access Management(IAM) verwaltet. Dieser Abschnitt enthält Anleitungen zur Konfiguration der erforderlichen Berechtigungen für die Interaktion mit der Revenue Measurement API, einschließlich der Berechtigungen, die zum Auflisten von AWS Marketplace Entitäten erforderlich sind.

Voraussetzungen

Stellen Sie vor der Konfiguration von Berechtigungen sicher, dass Ihre mit Partner Central verknüpft AWS-Konto ist und dass Sie die erforderlichen IAM-Rollen und -Benutzer erstellt haben. Weitere Informationen finden Sie unter [Einrichtung und Authentifizierung](#).

Verwenden AWS Verwaltete Richtlinien

AWS stellt verwaltete Richtlinien bereit, die die erforderlichen Berechtigungen für die Interaktion mit der Revenue Measurement API gewähren. Abhängig von Ihrer Persona sind zwei verwaltete Richtlinien verfügbar:

- Partner — fügen Sie die `AWSPartnerCentralRevenueAttributionManagement` Richtlinie Ihren IAM-Identitäten hinzu.

- Kunden — fügen Sie die AWSRevenueAttributionManagement Richtlinie Ihren IAM-Identitäten hinzu.

Weitere Informationen finden Sie unter [AWS Verwaltete Richtlinien für AWS Partner Central-Benutzer](#).

AWSPartnerCentralRevenueAttributionManagement Richtlinie

Diese Richtlinie bietet den erforderlichen Zugriff für Aktivitäten zur Verwaltung der Umsatzzuweisung für AWS Konten, die bei AWS Partner Central registriert sind.

```
{
  "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 Richtlinie

Diese Richtlinie bietet AWS Konten, die nicht bei AWS Partner Central registriert sind, den erforderlichen Zugriff auf Aktivitäten zur Verwaltung der Umsatzzuweisung.

```

{
  "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"
          ]
        }
      }
    }
  ]
}

```

Zuweisen von Richtlinien zu IAM-Rollen und -Benutzern

Gehen Sie wie folgt vor, um IAM-Rollen und -Benutzern Richtlinien zuzuweisen:

1. Melden Sie sich bei der AWS-Managementkonsole an.
2. Navigieren Sie zum IAM-Dienst.
3. Wählen Sie Rollen oder Benutzer aus und wählen Sie die IAM-Rolle oder den IAM-Benutzer aus, dem Sie eine Richtlinie zuordnen möchten.
4. Ordnen Sie die `AWSPartnerCentralRevenueAttributionManagement` Richtlinie (für Partner) oder die `AWSRevenueAttributionManagement` Richtlinie (für Kunden) der ausgewählten IAM-Rolle oder dem ausgewählten IAM-Benutzer zu.

Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

Verwaltung von Berechtigungen mithilfe von Bedingungsschlüsseln

Bedingungsschlüssel in IAM-Richtlinien bieten Berechtigungen auf Ressourcenebene, mit denen festgelegt werden kann, wann Richtlinien für Kontoauszüge durchgesetzt werden müssen. Sie können Bedingungsschlüssel verwenden, um Bedingungen festzulegen, die vorschreiben, wann bestimmte Berechtigungen erlaubt oder verweigert werden.

Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingungs-Operatoren](#).

Übersicht über die Bedingungsschlüssel

Bedingungsschlüssel	Description	Anwendbare Aktionen	Zulässige Werte
Partner Central: Katalog	Filtert den Zugriff nach dem Typ der zugehörigen Katalogeigenschaft	Alle Aktionen	AWS, Sandbox

Zusammenfassung der erforderlichen Berechtigungen

Zusammenfassung der erforderlichen Berechtigungen

Action	Description
Partnercentral: CreateRevenueAttribution	Ermöglicht die Erstellung eines neuen Datensatzes zur Umsatzzuweisung
Partnercentral: UpdateRevenueAttribution	Ermöglicht die Aktualisierung eines bestehenden Einnahmenezuweisungsdatensatzes
Partnercentral: GetRevenueAttribution	Ermöglicht das Abrufen der Details einer bestimmten Umsatzzuweisung
Partnercentral: ListRevenueAttributions	Ermöglicht die Auflistung von Umsatzzuordnungen mit optionalen Filtern
Partnercentral: GetRevenueAttributionAllocation	Ermöglicht das Abrufen einer einzelnen Zuordnung anhand ihrer ID
Partnercentral: ListRevenueAttributionAllocations	Ermöglicht das Auflisten von zugesagten Zuweisungen mit Filterunterstützung
Partnercentral: StartRevenueAttributionAllocationsTask	Ermöglicht das Einreichen eines Stapels von bis zu 250 Zuordnungsänderungen für die asynchrone Verarbeitung
Partnercentral: GetRevenueAttributionAllocationsTask	Ermöglicht das Abrufen des Status einer zuvor eingereichten Zuordnungsaufgabe
Partnercentral: CreateMarketplaceRevenueShare	Ermöglicht die Erstellung einer neuen Marketplace-Ressource zur Umsatzbeteiligung
Partnercentral: GetMarketplaceRevenueShare	Ermöglicht das Abrufen von Details zu einer bestimmten Marktplatz-Umsatzbeteiligung
Partnercentral: ListMarketplaceRevenueShares	Ermöglicht die Auflistung von Umsatzanteilen auf Marktplätzen mit optionalen Filtern

Action	Description
Partnercentral: CreateMarketplaceRevenueShareAllocation	Ermöglicht die Erstellung einer neuen Zuteilung von Umsatzanteilen auf dem Marktplatz
Partnercentral: GetMarketplaceRevenueShareAllocation	Ermöglicht das Abrufen von Details zu einer bestimmten Zuteilung von Umsatzanteilen auf dem Marktplatz
Partnercentral: UpdateMarketplaceRevenueShareAllocation	Ermöglicht die Aktualisierung einer bestehenden Zuteilung von Umsatzanteilen auf dem Marktplatz
Partnercentral: ListMarketplaceRevenueShareAllocations	Ermöglicht die Auflistung von Allokationen im Rahmen einer Marktplatz-Umsatzbeteiligung
Partnercentral: TagResource	Ermöglicht das Hinzufügen oder Überschreiben von Tags für eine Ressource
Partnercentral: UntagResource	Ermöglicht das Entfernen von Tags aus einer Ressource
Partnercentral: ListTagsForResource	Ermöglicht das Auflisten von Tags, die mit einer Ressource verknüpft sind

Kontingente für AWS API für Partner Central

Die folgenden Abschnitte enthalten Informationen zu Servicekontingenten.

Themen

- [Kontingente für AWS Partner Central-Konto-API](#)
- [Kontingente für AWS Vertriebs-API von Partner Central](#)
- [Kontingente für AWS API für Vorteile von Partner Central](#)
- [Kontingente für AWS Channel-API für Partner Central](#)
- [Kontingente für die AWS API zur Umsatzmessung in Partner Central](#)

Kontingente für AWS Partner Central-Konto-API

Die AWS Partner Central Account API hat die folgenden Kontingente.

Zusätzliche Kontingente

Zusätzliche Kontingente

Anzeigenname	Katalog	Description	Standardwert
Öffnen Sie Verbindun gseinladungen pro Konto	AWS	Die maximale Anzahl von Einladungen für offene Verbindungen, die Sie mit Partnerko nten im AWS Katalog verwalten können	1.000
Aktive Verbindungen pro Konto	AWS	Die maximale Anzahl aktiver Verbindungen, die Sie mit Partnerko nten im AWS Katalog aufrechterhalten können	1.000
E-Mail-Domänen pro Partner	AWS	Die maximale Anzahl von E-Mail-Domänen, die einem Partnerko nto für die AWS Schulungszertifizi erung im AWS Katalog zugeordnet werden können	50
Rate der Verbindun gseinladungen pro Konto	AWS	Die maximale Anzahl von Verbindun gseinladungen pro Tag, die Sie im AWS Katalog versenden können	50

Anzeigenname	Katalog	Description	Standardwert
Rate der Aufgaben zur Profilaktualisierung pro Konto	AWS	Die maximale Anzahl von Aufgaben zur Profilaktualisierung pro Tag, die Sie im AWS Katalog initiieren können	10
Rate der Aktualisierungen der Profilsichtbarkeit pro Konto	AWS	Die maximale Anzahl von Aktualisierungen der Profilsichtbarkeit pro Tag, die Sie im AWS Katalog initiieren können	5
Rate der E-Mail-Bestätigungs_codes pro E-Mail-Adresse	AWS	Die maximale Anzahl von E-Mail-Bestätigungs_codes pro Tag, die Sie im AWS Katalog anfordern können	5
Öffnen Sie Verbindungseinladungen pro Konto	Sandbox	Die maximale Anzahl von Einladungen für offene Verbindungen, die Sie mit Partnerkonten im Sandbox-Katalog verwalten können	1.000
Aktive Verbindungen pro Konto	Sandbox	Die maximale Anzahl aktiver Verbindungen, die Sie mit Partnerkonten im Sandbox-Katalog aufrechterhalten können	1.000

Anzeigenname	Katalog	Description	Standardwert
E-Mail-Domänen pro Partner	Sandbox	Die maximale Anzahl von E-Mail-Domänen, die einem Partnerkonto für die AWS Schulungszertifizierung im Sandbox-Katalog zugeordnet werden können	50
Rate der Verbindungseinladungen pro Konto	Sandbox	Die maximale Anzahl von Verbindungseinladungen pro Tag, die Sie im Sandbox-Katalog versenden können	50
Rate der Aufgaben zur Profilaktualisierung pro Konto	Sandbox	Die maximale Anzahl von Aufgaben zur Profilaktualisierung pro Tag, die Sie im Sandbox-Katalog initiieren können	10
Rate der Aktualisierungen der Profilsichtbarkeit pro Konto	Sandbox	Die maximale Anzahl von Aktualisierungen der Profilsichtbarkeit pro Tag, die Sie im Sandbox-Katalog initiieren können	5

Anzeigenname	Katalog	Description	Standardwert
Rate der E-Mail-Bestätigungs-codes pro E-Mail-Adresse	Sandbox	Die maximale Anzahl von E-Mail-Bestätigungs-codes pro Tag, die Sie im Sandbox-Katalog anfordern können. Aus dem Sandbox-Katalog werden keine E-Mails gesendet.	N/A

Kontingente verstehen und verwalten

Ratenbegrenzung

Wenn ein API-Ratenlimit erreicht ist, antwortet der Dienst mit einem `ThrottlingException`. Um die Ratenbegrenzung besser handhaben zu können, empfiehlt AWS, exponentielle Backoff- und Wiederholungsstrategien in Ihrer Anwendung zu implementieren.

Beantragen einer Kontingenterhöhung

Wenn die Standardkontingente nicht Ihren Anforderungen entsprechen, können Sie auf der [Seite Servicekontingente eine Erhöhung des Kontingents](#) beantragen. Die Service Quotas Console ist eine browserbasierte Oberfläche, mit der Sie Ihre Servicekontingente anzeigen und verwalten können. Sie können von jeder AWS-Managementkonsole Seite aus auf Service Quotas zugreifen, indem Sie sie in der oberen Navigationsleiste auswählen oder indem Sie in der nach Service Quotas suchen AWS-Managementkonsole.

Kontingente für AWS Vertriebs-API von Partner Central

AWS Die Vertriebs-API von Partner Central setzt Kontingente durch, um eine faire Nutzung zu gewährleisten und den Service vor Missbrauch zu schützen. Im Folgenden finden Sie die detaillierten Kontingente für verschiedene API-Operationen und -Verknüpfungen pro Opportunity.

Kontingente für API-Operationen

Typ	API-Operation	Kontingent (pro Partnerkonto)
Aktionen lesen	GetOpportunity	10 pro Sekunde; 100.000 pro 24 Stunden
	GetAwsOpportunitySummary	
	ListOpportunities	
	ListSolutions	
	GetEngagementInvitation	
	ListEngagementInvitations	
Akquise, Lesevorgänge	GetProspectingFromEngagementTask	100 pro Sekunde
Aktionen zur Kundenakquise	ListProspectingFromEngagementTasks	50 pro Sekunde
Schreiben Sie Aktionen	CreateOpportunity	1 pro Sekunde; 10.000 pro 24 Stunden
	UpdateOpportunity	
	AssociateOpportunity	
	DisassociateOpportunity	
	RejectEngagementInvitation	
	AssignOpportunity	
	StartEngagementFromOpportunityTask	
	StartEngagementByAcceptingInvitationTask	

Typ	API-Operation	Kontingent (pro Partnerkonto)
Akquise, Schreibaktionen	StartProspectingFromEngagementTask	2 pro Sekunde
Engagement: Schreiben Sie Aktionen	CreateEngagement	15 pro Sekunde

Vereinsquoten pro Opportunity

Verbundene Entität	Kontingent
AWS produkte	20 pro Gelegenheit
Partnerlösungen	10 pro Gelegenheit
AWS Marketplace Lösungen	10 pro Gelegenheit
AWS Marketplace produkte	10 pro Gelegenheit
AWS Marketplace private Angebote	1 pro Gelegenheit

Kontingente verstehen und verwalten

Ratenbegrenzung

Wenn ein API-Ratenlimit erreicht ist, antwortet der Dienst mit einem `ThrottlingException`. Um die Ratenbegrenzung besser handhaben zu können, empfiehlt AWS, exponentielle Backoff- und Wiederholungsstrategien in Ihrer Anwendung zu implementieren.

Zeitfenster für Kontingente

Die täglichen Kontingente werden auf einen fortlaufenden Zeitraum von 24 Stunden zurückgesetzt. Ihre Anfragen würden gedrosselt, z. B. wenn Sie in den letzten 24 Stunden 10.000 Schreibaktionen ausgeführt haben und versuchen, die 10.001. Anfrage auszuführen. Stellen Sie sicher, dass die Nutzungsmuster Ihrer Anwendung dies berücksichtigen, um eine unbeabsichtigte Drosselung zu verhindern.

Beantragen einer Kontingenterhöhung

Wenn die Standardkontingente nicht Ihren Anforderungen entsprechen, können Sie auf der [Seite Servicekontingente eine Erhöhung des Kontingents](#) beantragen. Die Service Quotas Console ist eine browserbasierte Oberfläche, mit der Sie Ihre Servicekontingenten anzeigen und verwalten können. Sie können von jeder AWS-Managementkonsole Seite aus auf Service Quotas zugreifen, indem Sie sie in der oberen Navigationsleiste auswählen oder indem Sie in der nach Service Quotas suchen AWS-Managementkonsole.

Kontingente für AWS API für Vorteile von Partner Central

Die AWS Partner Central Benefits API hat die folgenden Kontingente.

Anforderungskontingente

Anforderungskontingente

API-Operationen	Kontingent (pro AWS Konto)
ListBenefits	20 pro Sekunde
GetBenefit	20 pro Sekunde
AmendBenefitApplication	10 pro Sekunde
CancelBenefitApplication	10 pro Sekunde
CreateBenefitApplication	10 pro Sekunde
GetBenefitApplication	20 pro Sekunde
ListBenefitApplications	30 pro Sekunde
RecallBenefitApplication	10 pro Sekunde
SubmitBenefitApplication	10 pro Sekunde
UpdateBenefitApplication	10 pro Sekunde
GetBenefitAllocation	20 pro Sekunde
ListBenefitAllocations	20 pro Sekunde

API-Operationen	Kontingent (pro AWS Konto)
AssociateBenefitApplicationResource	10 pro Sekunde
DisassociateBenefitApplicationResource	10 pro Sekunde

Zusätzliche Kontingente

Zusätzliche Kontingente

Anzeigenname	Katalog	Description	Standardwert
Nutzungsanträge	AWS	Die maximale Anzahl von Leistungsanträgen, die Sie im AWS Katalog erstellen können	10.000
Anträge auf Sozialleistungen	Sandbox	Die maximale Anzahl von Leistungsanwendungen, die Sie im Sandbox-Katalog erstellen können	10.000

Kontingente verstehen und verwalten

Ratenbegrenzung

Wenn ein API-Ratenlimit erreicht ist, antwortet der Dienst mit einem `ThrottlingException`. Um die Ratenbegrenzung besser handhaben zu können, empfiehlt AWS, exponentielle Backoff- und Wiederholungsstrategien in Ihrer Anwendung zu implementieren.

Beantragen einer Kontingenterhöhung

Wenn die Standardkontingente nicht Ihren Anforderungen entsprechen, können Sie auf der [Seite Servicekontingente eine Erhöhung des Kontingents](#) beantragen. Die Service Quotas Console ist eine browserbasierte Oberfläche, mit der Sie Ihre Servicekontingente anzeigen und verwalten können. Sie können von jeder AWS-Managementkonsole Seite aus auf Service Quotas zugreifen, indem Sie

sie in der oberen Navigationsleiste auswählen oder indem Sie in der nach Service Quotas suchen AWS-Managementkonsole.

Kontingente für AWS Channel-API für Partner Central

Die AWS Partner Central Channel API hat die folgenden Kontingente.

Anforderungskontingente

Anforderungskontingente

API-Operationen	Kontingent (pro AWS Konto)
CreateProgramManagementAccount	3 pro Sekunde
UpdateProgramManagementAccount	3 pro Sekunde
ListProgramManagementAccounts	5 pro Sekunde
DeleteProgramManagementAccount	3 pro Sekunde
CreateChannelHandshake	3 pro Sekunde
ListChannelHandshakes	5 pro Sekunde
AcceptChannelHandshake	3 pro Sekunde
RejectChannelHandshake	3 pro Sekunde
CancelChannelHandshake	3 pro Sekunde
CreateRelationship	3 pro Sekunde
GetRelationship	5 pro Sekunde
ListRelationships	5 pro Sekunde
UpdateRelationship	3 pro Sekunde
DeleteRelationship	3 pro Sekunde

Zusätzliche Kontingente

Zusätzliche Kontingente

Anzeigenname	Katalog	Description	Standardwert
Konten für die Programmverwaltung	AWS	Die maximale Anzahl von Programmverwaltungsconten, die Sie im AWS Katalog erstellen können	50
Beziehungen pro Programmverwaltungs-konto	AWS	Die maximale Anzahl von Beziehungen, die Sie pro Programmverwaltungs-konto im AWS Katalog einrichten können	1.000
Handshakes für das Programmverwaltungs-konto pro Konto	AWS	Die maximale Anzahl aktiver Handshakes für Programmverwaltungsconten im Katalog AWS	100
Handshakes für den Startzeitraum pro Konto	AWS	Die maximale Anzahl aktiver Handshakes für die Einrichtung von Servicezeiträumen im Katalog AWS	200
Widerrufen Sie die Handshakes für den Servicezeitraum pro Konto	AWS	Die maximale Anzahl aktiver Handshakes für den Widerruf von Servicezeiträumen im Katalog AWS	200
Anzahl der Handshakes pro	AWS	Die maximale Anzahl von Handshakes für	5

Anzeigenname	Katalog	Description	Standardwert
Tag bei Programmverwaltungs-konten		Programmverwaltungskonten, die Sie pro Tag im Katalog an dasselbe AWS Konto senden können AWS	
Anzahl der Handshakes pro Tag, die während des Servicezeitraums gestartet wurden	AWS	Die maximale Anzahl von Handshakes, die Sie pro Tag im Katalog an dasselbe AWS Konto senden können AWS	5
Anzahl der Handshakes, die während der Servicezeit widerrufen werden, pro Tag	AWS	Die maximale Anzahl von Handshakes, die Sie pro Tag im Katalog an dasselbe AWS Konto senden können AWS	5
Konten für die Programmverwaltung	Sandbox	Die maximale Anzahl von Programmverwaltungs-konten, die Sie im Sandbox-Katalog erstellen können	50
Beziehungen pro Programmverwaltungs-konto	Sandbox	Die maximale Anzahl von Beziehungen, die Sie pro Programmverwaltungs-konto im Sandbox-Katalog einrichten können	1.000

Anzeigenname	Katalog	Description	Standardwert
Handshakes für das Programmverwaltungskonto pro Konto	Sandbox	Die maximale Anzahl aktiver Handshakes für Programmverwaltungskonten im Sandbox-Katalog	100
Handshakes für den Startzeitraum pro Konto	Sandbox	Die maximale Anzahl aktiver Handshakes für die Einrichtung von Servicezeiträumen im Sandbox-Katalog	200
Widerrufen Sie die Handshakes für den Servicezeitraum pro Konto	Sandbox	Die maximale Anzahl aktiver Handshakes für den Widerruf von Servicezeiträumen im Sandbox-Katalog	200
Anzahl der Handshakes pro Tag bei Programmverwaltungskonten	Sandbox	Die maximale Anzahl von Handshakes für Programmverwaltungskonten, die Sie pro Tag im Sandbox-Katalog an dasselbe AWS Konto senden können	5
Anzahl der Handshakes pro Tag, die während des Servicezeitraums gestartet wurden	Sandbox	Die maximale Anzahl von Handshakes, die Sie pro Tag im Sandbox-Katalog an dasselbe AWS Konto senden können	5

Anzeigenname	Katalog	Description	Standardwert
Häufigkeit von Handshakes für den Widerruf der Serviceperiode pro Tag	Sandbox	Die maximale Anzahl von Handshakes zum Widerruf des Servicezeitraums, die Sie pro Tag im Sandbox-Katalog an dasselbe AWS Konto senden können	5

Kontingente verstehen und verwalten

Ratenbegrenzung

Wenn ein API-Ratenlimit erreicht ist, antwortet der Dienst mit einem `ThrottlingException`. Um die Ratenbegrenzung besser handhaben zu können, empfiehlt AWS, exponentielle Backoff- und Wiederholungsstrategien in Ihrer Anwendung zu implementieren.

Beantragen einer Kontingenterhöhung

Wenn die Standardkontingente nicht Ihren Anforderungen entsprechen, können Sie auf der [Seite Servicekontingente eine Erhöhung des Kontingents](#) beantragen. Die Service Quotas Console ist eine browserbasierte Oberfläche, mit der Sie Ihre Servicekontingenten anzeigen und verwalten können. Sie können von jeder AWS-Managementkonsole Seite aus auf Service Quotas zugreifen, indem Sie sie in der oberen Navigationsleiste auswählen oder indem Sie in der nach Service Quotas suchen AWS-Managementkonsole.

Kontingente für die AWS API zur Umsatzmessung in Partner Central

AWS Die Partner Central Revenue Measurement API setzt Kontingente durch, um eine faire Nutzung zu gewährleisten und den Service vor Missbrauch zu schützen. Im Folgenden finden Sie die detaillierten Kontingente für verschiedene API-Operationen.

Kontingente für API-Operationen

Typ	API-Operation	Kontingent (pro Partnerkonto)
Aktionen lesen	GetRevenueAttribution	50 pro Sekunde
	GetRevenueAttributionAllocation	
	GetRevenueAttributionAllocationsTask	
	GetMarketplaceRevenueShare	
	GetMarketplaceRevenueShareAllocation	
Aktionen auflisten	ListRevenueAttributions	10 pro Sekunde
	ListRevenueAttributionAllocations	
	ListMarketplaceRevenueShares	
	ListMarketplaceRevenueShareAllocations	
	ListTagsForResource	
Schreiben Sie Aktionen	CreateRevenueAttribution	2 pro Sekunde
	UpdateRevenueAttribution	
	CreateMarketplaceRevenueShare	
	CreateMarketplaceRevenueShareAllocation	

Typ	API-Operation	Kontingent (pro Partnerkonto)
	UpdateMarketplaceRevenueShareAllocation	
	TagResource	
	UntagResource	
Asynchrone Schreibaktionen	StartRevenueAttributionAllocationsTask	1 pro Sekunde

Kontingente verstehen und verwalten

Ratenbegrenzung

Wenn ein API-Ratenlimit erreicht ist, antwortet der Dienst mit einem `ThrottlingException`. Um die Ratenbegrenzung besser handhaben zu können, AWS empfiehlt es sich, exponentielle Backoff- und Wiederholungsstrategien in Ihrer Anwendung zu implementieren.

Beantragen einer Kontingenterhöhung

Wenn die Standardkontingente nicht Ihren Anforderungen entsprechen, können Sie auf der [Seite Servicekontingente eine Erhöhung des Kontingents](#) beantragen. Die Service Quotas Console ist eine browserbasierte Oberfläche, mit der Sie Ihre Servicekontingenten anzeigen und verwalten können. Sie können von jeder AWS-Managementkonsole Seite aus auf Service Quotas zugreifen, indem Sie sie in der oberen Navigationsleiste auswählen oder indem Sie im nach Service Quotas suchen AWS-Managementkonsole.

Protokollierung für die AWS Partner Central-API

Die folgenden Abschnitte enthalten Informationen zur Protokollierung.

Themen

- [Protokollierung der AWS Konto-API für Partner Central](#)
- [Protokollierung der AWS Vertriebs-API für Partner Central](#)
- [Protokollierung der AWS API für Vorteile von Partner Central](#)
- [Protokollierung der AWS Channel-API für Partner Central](#)

- [Protokollierung der AWS API zur Umsatzzuweisung in Partner Central](#)

Protokollierung der AWS Konto-API für Partner Central

[AWS CloudTrail](#) ist ein Service, der die Unternehmensführung, die Einhaltung von Vorschriften, die betriebliche Prüfung und die Risikoprüfung Ihres Unternehmens ermöglicht AWS-Konto. Damit können Sie Kontoaktivitäten im Zusammenhang mit AWS CloudTrail Aktionen in Ihrer gesamten AWS Infrastruktur protokollieren, kontinuierlich überwachen und speichern. AWS Die API-Aktivitäten für das Partner Central-Konto werden als Ereignisse in aufgezeichnet CloudTrail. Sie können einen Trail erstellen, eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht.

-Übersicht

Die AWS Partner Central-Konto-API ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in AWS Partner Central ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für die AWS Partner Central-Konto-API als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von AWS Partner Central und von Code-Aufrufen an die API-Operationen des AWS Partner Central-Kontos.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für die AWS Partner Central Account API. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf einsehen.

Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an die AWS Partner Central Account API gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Verstehen AWS Einträge in der Protokolldatei der Partner Central Account API

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht. Wenn Ihr Trail API-Ereignisse für das AWS Partner Central-Konto verfolgt, CloudTrail werden die Ereignisse als Protokolldateien in allen Regionen verarbeitet. Jede Protokolldatei kann ein oder mehrere Ereignisse enthalten.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreatePartner` Aktion auf der AWS Partner Central Account API demonstriert:

```

{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/CloudTrailTestUser",
    "accountId": "123456789012",
    "accessKeyId": "ABCDEFGHJKLMNOP1234",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROEXAMPLE52AGFT725JGDZ",
        "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"
  }
}

```

In diesem Beispiel wurde die `CreatePartner` Aktion von dem IAM-Benutzer mit dem Namen `CloudTrailTestUser` aufgerufen. Die Anfrage wurde am 7. November 2025 um 19:45:58 UTC gestellt. Durch die Anfrage wurde ein neuer Partner mit der ID `EXAMPLE_PARTNER_ID` für den AWS-Katalog mit dem legalen Namen `"ExampleLegalName"` erstellt.

Felder in AWS Einträge in der Protokolldatei der Partner Central-Konto-API

Jeder Eintrag in einer CloudTrail Protokolldatei enthält Informationen darüber, wer eine Anfrage gestellt hat, auf welche Ressourcen bei der Anfrage reagiert wurde und welche Antwortelemente von der AWS Partner Central Account API zurückgegeben wurden. Die Liste der Felder in einem Protokolleintrag, z. B. `eventVersionuserIdentity`, `undeventTime`, enthält detaillierte Informationen über die Aktion. In dem `sourceIPAddress` Feld wird beispielsweise die IP-Adresse angezeigt, von der aus die Anfrage gestellt wurde.

Protokollierung der AWS Vertriebs-API für Partner Central

[AWS CloudTrail](#) ist ein Service, der die Unternehmensführung, die Einhaltung von Vorschriften, die betriebliche Prüfung und die Risikoprüfung Ihres Unternehmens ermöglicht AWS-Konto. Damit können Sie Kontoaktivitäten im Zusammenhang mit AWS CloudTrail Aktionen in Ihrer gesamten AWS Infrastruktur protokollieren, kontinuierlich überwachen und speichern. AWS Die API-Aktivitäten von Partner Central werden als Ereignisse in aufgezeichnet CloudTrail. Sie können einen Trail erstellen, eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht.

-Übersicht

Die AWS Partner Central-API ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in AWS Partner Central ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für AWS Partner Central als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von AWS Partner Central und von Codeaufrufen an die AWS Partner Central-API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS Partner Central. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf einsehen.

Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an AWS Partner Central gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Verständnis AWS Einträge in der Protokolldatei der Partner Central Selling API

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht. Wenn Ihr Trail Ereignisse in AWS Partner Central verfolgt, CloudTrail verarbeitet er die Ereignisse als Protokolldateien in allen Regionen. Jede Protokolldatei kann ein oder mehrere Ereignisse enthalten.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `ListOpportunities` Aktion in AWS Partner Central demonstriert:

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP1234",
    "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"
}
```

In diesem Beispiel wurde die `ListOpportunities` Aktion von dem IAM-Benutzer mit dem Namen `CloudTrailTestUser` aufgerufen. Die Aktion wurde in der US-East-1AWS-Region aufgerufen und die Anfrage wurde am 17. Oktober 2023 um 21:49:23 UTC gestellt.

Felder in AWS Einträge in der Protokolldatei der Partner Central Selling API

Jeder Eintrag in einer CloudTrail Protokolldatei enthält Informationen darüber, wer eine Anfrage gestellt hat, auf welche Ressourcen bei der Anfrage reagiert wurde und welche Antwortelemente von AWS Partner Central zurückgegeben wurden. Die Liste der Felder in einem Protokolleintrag, wie `eventVersion`, `userIdentity`, `eventTime`, enthält detaillierte Informationen über die Aktion. In dem `sourceIPAddress` Feld wird beispielsweise die IP-Adresse angezeigt, von der aus die Anfrage gestellt wurde.

Protokollierung der AWS API für Vorteile von Partner Central

[AWS CloudTrail](#) ist ein Service, der die Unternehmensführung, die Einhaltung von Vorschriften, die betriebliche Prüfung und die Risikoprüfung Ihres Unternehmens ermöglicht AWS-Konto. Damit können Sie Kontoaktivitäten im Zusammenhang mit AWS CloudTrail Aktionen in Ihrer gesamten AWS Infrastruktur protokollieren, kontinuierlich überwachen und speichern. AWS Die API-Aktivitäten von Partner Central Benefits werden als Ereignisse in aufgezeichnet CloudTrail. Sie können einen Trail erstellen, eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht.

-Übersicht

Die AWS Partner Central Benefits API ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in AWS Partner Central ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für die AWS Partner Central Benefits API als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von AWS Partner Central und von Codeaufrufen an die AWS Partner Central Benefits API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für die AWS Partner Central Benefits API. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf einsehen.

Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an die AWS Partner Central Benefits API gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Verstehen AWS Einträge in der Protokolldatei der Partner Central Benefits API

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht. Wenn Ihr Trail API-Ereignisse von AWS Partner Central Benefits

verfolgt, CloudTrail verarbeitet er die Ereignisse als Protokolldateien in allen Regionen. Jede Protokolldatei kann ein oder mehrere Ereignisse enthalten.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `ListBenefitApplications` Aktion auf der AWS Partner Central Benefits API demonstriert:

```
{
  "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"  
}
```

In diesem Beispiel wurde die `ListBenefitApplications` Aktion vom IAM-Benutzer namens Alice aufgerufen. Die Anfrage wurde am 21. Oktober 2025 um 03:10:59 UTC gestellt. In der Anfrage wurden Leistungsanträge für den AWS-Katalog aufgeführt.

Felder in AWS Einträge in der Protokolldatei der Partner Central Benefits API

Jeder Eintrag in einer CloudTrail Protokolldatei enthält Informationen darüber, wer eine Anfrage gestellt hat, auf welche Ressourcen bei der Anfrage reagiert wurde und welche Antwortelemente von der AWS Partner Central Benefits API zurückgegeben wurden. Die Liste der Felder in einem Protokolleintrag, z. B. `eventVersionuserIdentity`, `undeventTime`, enthält detaillierte Informationen über die Aktion. In dem `sourceIPAddress` Feld wird beispielsweise die IP-Adresse angezeigt, von der aus die Anfrage gestellt wurde.

Protokollierung der AWS Channel-API für Partner Central

[AWS CloudTrail](#) ist ein Service, der die Unternehmensführung, die Einhaltung von Vorschriften, die betriebliche Prüfung und die Risikoprüfung Ihres Unternehmens ermöglicht AWS-Konto. Damit können Sie Kontoaktivitäten im Zusammenhang mit AWS CloudTrail Aktionen in Ihrer gesamten AWS Infrastruktur protokollieren, kontinuierlich überwachen und speichern. AWS Die Aktivitäten der Partner Central Channel API werden als Ereignisse in aufgezeichnet CloudTrail. Sie können einen Trail erstellen, eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht.

-Übersicht

Die AWS Partner Central Channel API ist in einen Dienst integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Dienst in AWS Partner Central ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für die AWS Partner Central Channel API als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von AWS Partner Central und von Codeaufrufen an die AWS Partner Central Channel API-Operationen.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für die AWS Partner Central Channel API. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf einsehen.

Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an die AWS Partner Central Channel API gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Verstehen AWS Einträge in der Protokolldatei der Partner Central Channel API

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht. Wenn Ihr Trail Ereignisse der AWS Partner Central Channel API verfolgt, CloudTrail verarbeitet er die Ereignisse als Protokolldateien in allen Regionen. Jede Protokolldatei kann ein oder mehrere Ereignisse enthalten.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateProgramManagementAccount` Aktion auf der AWS Partner Central Channel API demonstriert:

```
{
  "eventVersion": "1.11",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROEXAMPLE52AGFT725JGDZ:example-user-Isengard",
    "arn": "arn:aws:sts::123456789012:assumed-role/Admin/example-user-Isengard",
    "accountId": "123456789012",
    "accessKeyId": "ASIAEXAMPLE52AGL6IXQLZ5",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROEXAMPLE52AGFT725JGDZ",
        "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"
    }
  }
}

```

In diesem Beispiel wurde die `CreateProgramManagementAccount` Aktion von der IAM-Rolle aufgerufen, die `ExampleRole` über eine angenommene Rollensitzung benannt wurde. Die Anfrage wurde am 21. Oktober 2025 um 17:07:26 UTC gestellt. Durch die Anfrage wurde ein neues Programmverwaltungskonto mit ID `pma-example123456789` für das Solution Provider-Programm erstellt.

Felder in AWS Einträge in der Protokolldatei der Partner Central Channel API

Jeder Eintrag in einer CloudTrail Protokolldatei enthält Informationen darüber, wer eine Anfrage gestellt hat, auf welche Ressourcen bei der Anfrage reagiert wurde und welche Antwortelemente von der AWS Partner Central Channel API zurückgegeben wurden. Die Liste der Felder in einem Protokolleintrag, z. B. `eventVersionuserIdentity`, `undeventTime`, enthält detaillierte Informationen über die Aktion. In dem `sourceIPAddress` Feld wird beispielsweise die IP-Adresse angezeigt, von der aus die Anfrage gestellt wurde.

Protokollierung der AWS API zur Umsatzzuweisung in Partner Central

[AWS CloudTrail](#) ist ein Service, der die Unternehmensführung, die Einhaltung von Vorschriften, die betriebliche Prüfung und die Risikoprüfung Ihres Unternehmens ermöglicht AWS-Konto. Damit können Sie Kontoaktivitäten im Zusammenhang mit AWS CloudTrail Aktionen in Ihrer gesamten AWS Infrastruktur protokollieren, kontinuierlich überwachen und speichern. AWS Die API-Aktivitäten von Partner Central Revenue Measurement werden als Ereignisse in aufgezzeichnet CloudTrail. Sie können einen Trail erstellen, eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht.

-Übersicht

Die AWS Partner Central Revenue Measurement API ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen eines Benutzers, einer Rolle oder eines AWS Dienstes in AWS Partner Central bereitstellt. CloudTrail erfasst alle API-Aufrufe für AWS Partner Central Revenue Attribution als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS Partner Central-Konsole und von Code-Aufrufen an die API-Operationen für die AWS Partner Central Revenue Measurement.

Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für AWS Partner Central Revenue Measurement. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf einsehen.

Anhand der von gesammelten Informationen können Sie die Anfrage CloudTrail, die an AWS Partner Central Revenue Measurement gestellt wurde, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Die folgenden API-Aktionen für die AWS Partner Central Revenue Measurement sind angemeldet CloudTrail:

- `CreateRevenueAttribution`
- `UpdateRevenueAttribution`
- `GetRevenueAttribution`
- `ListRevenueAttributions`
- `TagResource`
- `UntagResource`
- `ListTagsForResource`
- `CreateMarketplaceRevenueShare`
- `GetMarketplaceRevenueShare`
- `ListMarketplaceRevenueShares`
- `CreateMarketplaceRevenueShareAllocation`
- `GetMarketplaceRevenueShareAllocation`
- `UpdateMarketplaceRevenueShareAllocation`
- `ListMarketplaceRevenueShareAllocations`
- `StartRevenueAttributionAllocationsTask`
- `GetRevenueAttributionAllocationsTask`
- `ListRevenueAttributionAllocations`
- `GetRevenueAttributionAllocation`

Verstehen AWS Einträge in der Protokolldatei der Partner Central Revenue Measurement API

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen Amazon S3 S3-Bucket ermöglicht. Wenn Ihr Trail Ereignisse von AWS Partner Central Revenue Measurement verfolgt, werden die Ereignisse als Protokolldateien in allen Regionen CloudTrail verarbeitet. Jede Protokolldatei kann ein oder mehrere Ereignisse enthalten.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateRevenueAttribution` Aktion bei AWS Partner Central Revenue Attribution demonstriert:

```
{
  "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-eeef-4578-8db0-01ee2e34ee56",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

In diesem Beispiel wurde die `CreateRevenueAttribution` Aktion von dem IAM-Benutzer mit dem Namen `CloudTrailTestUser` aufgerufen. Die Aktion wurde in der `US-East-1` AWS-Region aufgerufen und die Anfrage wurde am 2. Juni 2026 um 11:53:54 UTC gestellt. Eine neue Ressource zur Umsatzzuweisung wurde mit der ID erstellt. `ra-0123456789abcdef0`

Felder in AWS Einträge in der Protokolldatei der Partner Central Revenue Measurement API

Jeder Eintrag in einer CloudTrail Protokolldatei enthält Informationen darüber, wer eine Anfrage gestellt hat, auf welche Ressourcen bei der Anfrage reagiert wurde und welche Antwortelemente von AWS Partner Central Revenue Measurement zurückgegeben wurden. Die Liste der Felder in einem Protokolleintrag, z. B. `eventVersionuserIdentity`, `undeventTime`, enthält detaillierte Informationen über die Aktion. In dem `sourceIPAddress` Feld wird beispielsweise die IP-Adresse angezeigt, von der aus die Anfrage gestellt wurde.

Benachrichtigungen für AWS Partner Central-API

Die folgenden Abschnitte enthalten Informationen zu Benachrichtigungen.

Themen

- [Benachrichtigungen für AWS Konto-API für Partner Central](#)
- [Benachrichtigungen für die AWS Vertriebs-API von Partner Central](#)
- [Benachrichtigungen für AWS API für Vorteile von Partner Central](#)
- [Benachrichtigungen für AWS Channel-API für Partner Central](#)

Benachrichtigungen für AWS Konto-API für Partner Central

Mithilfe von Verbindungsbenachrichtigungen für Partnerkonten können Partner über Ereignisse im Verbindungslebenszyklus auf dem Laufenden bleiben, die sich direkt auf ihre Geschäftsbeziehungen und betrieblichen Arbeitsabläufe auswirken. Diese Ereignisse sind für Partner von entscheidender Bedeutung, um:

- Reagieren Sie umgehend auf neue Kollaborationen
- Behalten Sie den Status ihres Verbindungsportfolios im Blick
- Ergreifen Sie geeignete Maßnahmen, wenn Verbindungen hergestellt oder beendet werden
- Stellen Sie die Geschäftskontinuität sicher, indem Sie wissen, wann sich Beziehungen ändern

Themen

- [Erfüllen Sie die Voraussetzungen für die Überwachung von Ereignissen](#)
- [Amazon für EventBridge die Überwachung von Ereignissen konfigurieren](#)

- [Erfahren Sie mehr über API-Ereignisse für Kontoverbindungen](#)

Erfüllen Sie die Voraussetzungen für die Überwachung von Ereignissen

Benutzer benötigen die entsprechenden IAM-Berechtigungen, um auf Ereignisse zuzugreifen und diese zu verwalten, die von der Kontoverbindungs-API veröffentlicht wurden. Weitere Informationen zu den verfügbaren Aktionen, Ressourcen und Bedingungsschlüsseln für EventBridge finden Sie unter [Verwenden von IAM-Richtlinienbedingungen in Amazon EventBridge](#) im EventBridge Amazon-Benutzerhandbuch. Einer der Bedingungsschlüssel ist `events:detail-type`, der verwendet werden kann, um Berechtigungen auf bestimmte Ereignistypen zu beschränken.

Die folgende Beispielrichtlinie zeigt, wie Sie die Berechtigungen für die vorgeschlagenen Ereignisse anpassen und einschränken können. Die `AllowPutRuleForPartnercentralAccountEvents` Anweisung ermöglicht die Erstellung von Regeln, jedoch nur für Ereignisse aus der `aws.partnercentral-account` Quelle.

Ausführliche Beispiele für IAM-Richtlinien finden Sie in der AWS-Dokumentation zu EventBridge Berechtigungen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralAccountEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-account.connection"
        }
      }
    }
  ]
}
```

Amazon für EventBridge die Überwachung von Ereignissen konfigurieren

Um API-Ereignisse für Kontoverbindungen zu überwachen, erstellen Sie eine EventBridge Regel, die den Ereignissen entspricht, die Sie erfassen möchten. Sie können die SDKs AWS-

Managementkonsole oder die AWS SDKs verwenden, um Regeln zu erstellen und zu verwalten. In den folgenden Abschnitten wird erklärt, wie Regeln mit beiden Methoden erstellt werden. Unabhängig von der verwendeten Methode müssen Sie die Regel in der us-east-1 Region USA Ost (Nord-Virginia) erstellen.

AWS-Managementkonsole Einrichten

Um eine EventBridge Regel mit der einzurichten AWS-Managementkonsole, folgen Sie den Schritten unter [Erstellen von Regeln, die auf Ereignisse EventBridge in Amazon reagieren](#) im EventBridge Amazon-Benutzerhandbuch. Bei der Erstellung der Regel müssen Sie den Event-Bus auf die Standardeinstellung setzen und die Regel in der us-east-1 Region USA Ost (Nord-Virginia) erstellen.

Im Folgenden finden Sie ein Beispiel für eine Ereignisregel:

```
{
  "source": ["aws.partnercentral-account"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS SDK-Setup

Sie können die AWS SDKs verwenden, um EventBridge Regeln programmgesteuert zu erstellen und zu verwalten. Weitere Informationen finden Sie [PutRule](#) in der Amazon EventBridge API-Referenz.

Das folgende Beispiel verwendet das AWS SDK for 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'])
```

Erfahren Sie mehr über API-Ereignisse für Kontoverbindungen

In den folgenden Abschnitten werden die API-Ereignistypen für Kontoverbindungen, Szenarien, die sie auslösen, und Beispiele für Ereignisse beschrieben.

Event types (Ereignistypen)

Im Folgenden sind die Ereignistypen und ihre Auslöser aufgeführt.

- [Einladung zur Partnerverbindung erhalten](#): Informiert den Empfänger, dass ein anderer Partner eine Verbindung herstellen möchte
- [Einladung zur Partnerverbindung akzeptiert](#): Informiert den Absender, dass seine Einladung angenommen wurde und nun eine Verbindung aktiv ist
- [Einladung zur Partnerverbindung abgelehnt](#): Informiert den Absender, dass seine Einladung abgelehnt wurde
- [Partnerverbindung abgebrochen](#): Benachrichtigt den anderen verbundenen Teilnehmer, wenn eine aktive Verbindung beendet wird
- [Einladung zur Partnerverbindung storniert](#): Informiert den Empfänger, dass der Absender seine Einladung zurückgezogen hat, bevor Maßnahmen ergriffen wurden
- [Einladung zur Partnerverbindung abgelaufen](#): Sowohl der Absender als auch der Empfänger werden darüber informiert, dass die Einladung abgelaufen ist

Beispielereignisse

Die folgenden Abschnitte enthalten Beispiele für die Ereignisse, die zuvor im vorherigen Abschnitt aufgeführt wurden.

Themen

- [Einladung zur Partnerverbindung erhalten](#)
- [Einladung zur Partnerverbindung akzeptiert](#)
- [Einladung zur Partnerverbindung abgelehnt](#)
- [Partnerverbindung wurde abgebrochen](#)
- [Einladung zur Partnerverbindung storniert](#)

- [Die Einladung zur Partnerverbindung ist abgelaufen](#)

Einladung zur Partnerverbindung erhalten

Auslösender Kontext

Dieses Ereignis wird generiert, wenn eine Verbindungseinladung erfolgreich erstellt und über die API im Datenspeicher von PAC gespeichert wurde. CreateConnectionInvitation Das Ereignis wird unmittelbar nach dem erfolgreichen Abschluss der Einladungserstellung gesendet und im Datenspeicher von PAC gespeichert, nicht erst, wenn der API-Aufruf erfolgt.

Das Ereignis wird automatisch gesendet, wenn eine neue Verbindungseinladung erstellt wird. Pro erstellter Einladung wird eine Veranstaltung ohne doppelte Ereignisse für dieselbe Einladung erstellt.

Empfänger

Das AWS Konto des Empfängers in der Verbindungseinladung.

Voraussichtlicher Umgang

Typische Kundenaktionen:

- Sofortige Benachrichtigung: Lassen Sie Geschäftsteams über neue Partnerschaftsmöglichkeiten benachrichtigen
- Workflow-Automatisierung: Automatische Aktualisierung der Partnerverwaltungssysteme mit ausstehenden Einladungen
- Entscheidungsunterstützung: Leitet Einladungen je nach Verbindungstyp an geeignete Entscheidungsträger weiter
- Auditprotokollierung: Notieren Sie den Eingang der Einladung, um die Einhaltung der Vorschriften einzuhalten und die Partnerschaft nachzuverfolgen
- Antwortautomatisierung: Für vertrauenswürdige Partner können bestimmte Einladungstypen möglicherweise automatisch akzeptiert werden

Allgemeine Integrationsmuster:

- Lambda-Funktion → Partnerportal-Dashboard aktualisieren
- SQS-Warteschlange → Einladungen zur Überprüfung Batch verarbeiten
- SNS-Thema → Email/Slack Benachrichtigungen an Geschäftsteams

- API-Ziel → Webhook zu externen Managementsystemen CRM/partner

Beispiel

```
{
  "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>"
    }
  }
}
```

Einladung zur Partnerverbindung akzeptiert

Auslösender Kontext

Dieses Ereignis wird generiert, wenn eine Verbindungseinladung erfolgreich angenommen wurde und eine Verbindung erstellt und über die API im Datenspeicher von PAC gespeichert wird.

AcceptConnectionInvitation Das Ereignis wird unmittelbar nach dem erfolgreichen Abschluss der Annahme der Einladung und dem Verbindungsaufbau gesendet.

Die Veranstaltung wird automatisch gesendet, wenn eine Verbindungseinladung angenommen wird. Pro Einladung wird eine Veranstaltung akzeptiert, ohne dass es zu doppelten Veranstaltungen für dieselbe Annahme kommt.

Empfänger

Beide AWS Konten waren an der Verbindungseinladung beteiligt: das Absenderkonto, das die Einladung ursprünglich gesendet hat, und das Empfängerkonto, das die Einladung angenommen hat.

Erwartete Bearbeitung

Typische Kundenaktionen:

- Aktivierung der Partnerschaft: Auslösen von Workflows zur Aktivierung von Layer-2-Geschäftsaktivitäten
- Statusaktualisierungen: Aktualisieren Sie die Partnerverwaltungssysteme mit dem aktiven Verbindungsstatus
- Benachrichtigung: Informieren Sie die Geschäftsteams über den erfolgreichen Aufbau einer Partnerschaft
- Zugriffsbereitstellung: Erteilen Sie automatisch die entsprechenden Berechtigungen für die Zusammenarbeit
- Analytik: Verfolgen Sie die Konversionsraten und Erfolgsmetriken von Partnerschaften

Allgemeine Integrationsmuster:

- Lambda-Funktion → Berechtigungen für die gemeinsame Nutzung von Daten aktivieren und das Partnerportal aktualisieren
- SQS-Warteschlange → Verbindungsaktivierungen für die Geschäftseinrichtung Batch verarbeiten
- SNS-Thema → Email/Slack Benachrichtigungen an Geschäfts- und Technikteams
- API-Ziel → Webhook zu CRM-Systemen zur Aktualisierung des Partnerschaftsstatus

Beispiel

```
{
  "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"
  }
}
}

```

Einladung zur Partnerverbindung abgelehnt

Auslösender Kontext

Dieses Ereignis wird generiert, wenn eine Verbindungseinladung über die `RejectConnectionInvitation` API erfolgreich abgelehnt wurde. Das Ereignis wird unmittelbar nach der Verarbeitung der Ablehnung der Einladung gesendet und im Datenspeicher von PAC gespeichert.

Das Ereignis wird automatisch gesendet, wenn eine Verbindungseinladung abgelehnt wird. Ein Ereignis pro Einladung wurde abgelehnt, ohne dass es zu doppelten Ereignissen für dieselbe Ablehnung kam.

Empfänger

Das AWS Konto, das ursprünglich die Verbindungseinladung gesendet hat (das Absenderkonto).

Erwartete Bearbeitung

Typische Kundenaktionen:

- **Statusaktualisierungen:** Aktualisieren Sie die Partnerverwaltungssysteme mit dem Ablehnungsstatus
- **Follow-up Aktionen:** Initiieren Sie Workflows für alternative Partnerschaftsansätze
- **Benachrichtigung:** Informieren Sie die Geschäftsteams über die Entscheidung über eine Partnerschaft

- Säuberung: Entfernen Sie ausstehende Einladungsverweise aus internen Systemen

Allgemeine Integrationsmuster:

- Lambda-Funktion → Partnerportal aktualisieren und Folgeworkflows auslösen
- SQS-Warteschlange → Ablehnungen von Batch-Prozessen für Geschäftsanalysen
- SNS-Thema → Email/Slack Benachrichtigungen an Geschäftsentwicklungsteams
- API-Ziel → Webhook zu CRM-Systemen, um den Opportunity-Status zu aktualisieren

Beispiel

```
{
  "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-****"
    }
  }
}
```

Partnerverbindung wurde abgebrochen

Auslösender Kontext

Dieses Ereignis wird generiert, wenn eine aktive Verbindung erfolgreich abgebrochen wurde und der aktive Status im Datenspeicher von PAC über die CancelConnection API auf „Beendet“ aktualisiert

wird. Das Ereignis wird unmittelbar nach der Verarbeitung des Verbindungsabbruchs und der Aktualisierung des Verbindungsstatus gesendet.

Das Ereignis wird automatisch gesendet, wenn eine Verbindung beendet wird. Ein Ereignis pro Verbindung wurde abgesagt, ohne dass es zu doppelten Ereignissen für dieselbe Stornierung kam.

Empfänger

Das AWS Konto des anderen Teilnehmers an der Verbindung (nicht des Teilnehmers, der die Stornierung veranlasst hat).

Voraussichtlicher Umgang

Typische Kundenaktionen:

- **Widerruf des Zugriffs:** Widerrufen Sie sofort Berechtigungen und deaktivieren Sie die gemeinsame Nutzung von Daten
- **Statusaktualisierungen:** Aktualisieren Sie die Partnerverwaltungssysteme mit dem Status „Verbindung beendet“
- **Benachrichtigung:** Informieren Sie Geschäfts- und Technikteams über die Kündigung der Partnerschaft
- **Säuberung:** Entfernen Sie Verbindungsreferenzen und bereinigen Sie gemeinsam genutzte Ressourcen
- **Analytik:** Verfolgen Sie die Verbindungsdauer und die Kündigungsmuster

Allgemeine Integrationsmuster:

- **Lambda-Funktion** → Berechtigungen widerrufen und Partnerportalstatus aktualisieren
- **SQS-Warteschlange** → Verbindungsabbrüche im Batch-Prozess für Bereinigungsworkflows
- **SNS-Thema** → Email/Slack Benachrichtigungen an Geschäfts- und Technikteams
- **API-Ziel** → Webhook zu CRM-Systemen zur Aktualisierung des Partnerschaftsstatus

Beispiel

```
{
  "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>>"
  }
}
}

```

Einladung zur Partnerverbindung storniert

Auslösender Kontext

Dieses Ereignis wird generiert, wenn eine ausstehende Verbindungseinladung erfolgreich über die `CancelConnectionInvitation` API storniert wurde. Die Veranstaltung wird unmittelbar nach der Bearbeitung der Stornierung der Einladung gesendet und der Einladungsstatus auf STORNIERT aktualisiert.

Die Veranstaltung wird automatisch gesendet, wenn eine Verbindungseinladung storniert wird. Eine Veranstaltung pro Verbindungseinladung wurde storniert, ohne dass es zu doppelten Terminen für dieselbe Stornierung kam.

Empfänger

Das AWS Konto, das die Verbindungseinladung erhalten sollte (das Empfängerkonto).

Voraussichtlicher Umgang

Typische Kundenaktionen:

- Statusaktualisierungen: Aktualisieren Sie die Partnerverwaltungssysteme, um ausstehende Einladungsverweise zu entfernen
- Benachrichtigung: Machen Sie die Geschäftsteams darauf aufmerksam, dass ein potenzielles Partnerschaftsangebot zurückgezogen wurde

- Säuberung: Entfernen Sie Einladungsverweise aus den internen Nachverfolgungssystemen
- Analytik: Verfolge die Stornierungsmuster von Einladungen, um Einblicke in deine Partnerschaft zu gewinnen

Allgemeine Integrationsmuster:

- Lambda-Funktion → Partnerportal aktualisieren und ausstehende Einladungsbenachrichtigungen entfernen
- SQS-Warteschlange → Stornierungen von Batch-Prozessen für Geschäftsanalysen
- SNS-Thema → Email/Slack Benachrichtigungen an Geschäftsentwicklungsteams
- API-Ziel → Webhook zu CRM-Systemen, um den Opportunity-Status zu aktualisieren

Beispiel

```
{
  "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-****"
    }
  }
}
```

Die Einladung zur Partnerverbindung ist abgelaufen

Auslösender Kontext

Dieses Ereignis wird generiert, wenn eine ausstehende Verbindungseinladung nach Erreichen ihres ExpiresAt Zeitstempels automatisch abläuft, ohne vom Empfänger akzeptiert oder abgelehnt zu werden. Das Ereignis wird automatisch vom System ausgelöst, wenn die Ablaufzeit der Einladung erreicht ist.

Das Ereignis wird automatisch gesendet, wenn eine Verbindungseinladung abläuft. Ein Ereignis pro Einladung ist abgelaufen, und es gab keine doppelten Ereignisse für denselben Ablauf.

Empfänger

Beide AWS Konten waren an der Verbindungseinladung beteiligt: das Absenderkonto, das die Einladung ursprünglich gesendet hat, und das Empfängerkonto, das die Einladung erhalten hat.

Voraussichtlicher Umgang

Typische Kundenaktionen:

- Statusaktualisierungen: Aktualisieren Sie die Partnerverwaltungssysteme, sodass die Einladung als abgelaufen markiert wird
- Follow-up Aktionen: Initiieren Sie Workflows für eine erneute Zusammenarbeit oder alternative Partnerschaftsansätze
- Benachrichtigung: Informieren Sie Geschäftsteams über verpasste Partnerschaftsmöglichkeiten
- Säuberung: Entfernen Sie abgelaufene Einladungsreferenzen aus den internen Nachverfolgungssystemen

Allgemeine Integrationsmuster:

- Lambda-Funktion → Partnerportal aktualisieren und Folgeworkflows auslösen
- SQS-Warteschlange → Ablaufdaten von Batch-Prozessen für Geschäftsanalysen
- SNS-Thema → Email/Slack Benachrichtigungen an Geschäftsentwicklungsteams
- API-Ziel → Webhook zu CRM-Systemen, um den Opportunity-Status zu aktualisieren

Beispiel

```
{
```

```
"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-****"
  }
}
```

Benachrichtigungen für die AWS Vertriebs-API von Partner Central

Ereignisse für die Verkaufs-API bieten Benachrichtigungen in Echtzeit über Statusänderungen oder Einzelheiten von Verkaufschancen. Diese Ereignisse tragen dazu bei, dass Ihre Systeme mit AWS Partner Central synchron bleiben und dass zeitnahe Antworten und Aktualisierungen gewährleistet sind.

Themen

- [Erfüllen Sie die Voraussetzungen für die Überwachung von Ereignissen](#)
- [Amazon für EventBridge die Überwachung von Ereignissen konfigurieren](#)
- [Erfahren Sie mehr über den Verkauf von API-Events](#)

Erfüllen Sie die Voraussetzungen für die Überwachung von Ereignissen

Benutzer benötigen die entsprechenden IAM-Berechtigungen, um auf Ereignisse zuzugreifen und diese zu verwalten, die von der Partner Central-Vertriebs-API veröffentlicht wurden. Weitere Informationen zu den verfügbaren Aktionen, Ressourcen und Bedingungsschlüsseln für EventBridge finden Sie unter [Verwenden von IAM-Richtlinienbedingungen in Amazon EventBridge](#) im

EventBridge Amazon-Benutzerhandbuch. Einer der Bedingungsschlüssel ist `events:detail-type`, der verwendet werden kann, um Berechtigungen auf bestimmte Ereignistypen zu beschränken.

Die folgende Beispielrichtlinie zeigt, wie Sie die Berechtigungen für die vorgeschlagenen Ereignisse anpassen und einschränken können. Die `AllowPutRuleForPartnercentralSellingEvents` Anweisung ermöglicht die Erstellung von Regeln, jedoch nur für Ereignisse aus der `aws.partnercentral-selling` Quelle.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "AllowPutRuleForPartnercentralSellingEvents",
    "Effect": "Allow",
    "Action": "events:PutRule",
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "events:source": "aws.partnercentral-selling"
      }
    }
  }
}
```

Amazon für EventBridge die Überwachung von Ereignissen konfigurieren

Um API-Verkaufsereignisse zu überwachen, erstellen Sie eine EventBridge Regel, die den Ereignissen entspricht, die Sie erfassen möchten. Sie können die AWS-Managementkonsole oder die AWS SDKs verwenden, um Regeln zu erstellen und zu verwalten. In den folgenden Abschnitten wird erklärt, wie Regeln mit beiden Methoden erstellt werden. Unabhängig von der verwendeten Methode müssen Sie die Regel in der `us-east-1` Region USA Ost (Nord-Virginia) erstellen.

AWS-Managementkonsole Einrichten

Um eine EventBridge Regel mit der einzurichten AWS-Managementkonsole, folgen Sie den Schritten unter [Erstellen von Regeln, die auf Ereignisse EventBridge in Amazon reagieren](#) im EventBridge Amazon-Benutzerhandbuch. Bei der Erstellung der Regel müssen Sie den Event-Bus

auf die Standardeinstellung setzen und die Regel in der us-east-1 Region USA Ost (Nord-Virginia) erstellen.

Im Folgenden finden Sie ein Beispiel für eine Ereignisregel:

```
{
  "source": ["aws.partnercentral-selling"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS SDK-Setup

Sie können die AWS SDKs verwenden, um EventBridge Regeln programmgesteuert zu erstellen und zu verwalten. Weitere Informationen finden Sie [PutRule](#) in der Amazon EventBridge API-Referenz.

Das folgende Beispiel verwendet das AWS SDK for 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'])
```

Erfahren Sie mehr über den Verkauf von API-Events

In den folgenden Abschnitten werden die Verkaufs-API-Ereignistypen, Szenarien, die sie auslösen, und Beispiele für Ereignisse beschrieben.

Event types (Ereignistypen)

Im Folgenden sind die Ereignistypen und ihre Auslöser aufgeführt.

- [Opportunity erstellt](#): Wird ausgelöst, wenn eine neue Opportunity erstellt wird.
- [Opportunity aktualisiert](#): Wird ausgelöst, wenn eine Opportunity (Opportunity oder AWS die dazugehörige Opportunity-Zusammenfassung) aktualisiert wird.
- [Interaktionseinladung erstellt](#): Wird ausgelöst, wenn eine AWS Empfehlungseinladung erstellt wird.
- [Einladung zur Verlobung angenommen](#): Wird ausgelöst, wenn ein Partner eine Einladung zur AWS Verlobung annimmt und damit sein Interesse an einer Zusammenarbeit AWS im Rahmen der Verkaufschance bestätigt.
- [Einladung zur Interaktion abgelehnt](#): Wird ausgelöst, wenn ein Partner eine Einladung zur AWS Interaktion ablehnt.
- [Verlobungseinladung abgelaufen](#): Dieses Ereignis wird ausgelöst, wenn ein Partner eine ablehnt. EngagementInvitation Es benachrichtigt den sendenden und empfangenden Partner darüber, dass die Einladung abgelaufen ist.
- [Engagement-Mitglied hinzugefügt](#): Dieses Ereignis wird ausgelöst, wenn ein neues Mitglied dem Engagement beitrifft, nachdem es eine Einladung angenommen hat. Es benachrichtigt alle aktuellen Mitglieder des Engagements darüber, dass das neue Mitglied zum Engagement hinzugefügt wurde.
- [Snapshot der Engagement-Ressource erstellt](#): Dieses Ereignis wird ausgelöst, wenn die neue Version des Snapshots der Ressourcen (z. B. Opportunities) erstellt wird, die einem Engagement zugeordnet sind. Es benachrichtigt alle aktuellen Mitglieder des Engagements über die Änderungen am zugehörigen Ressourcen-Snapshot.
- [Engagement erstellt](#): Wird ausgelöst, wenn ein neues Engagement erstellt wird.
- [Engagement aktualisiert](#): Wird ausgelöst, wenn ein Engagement aktualisiert wird.

Event-Szenarien

In der folgenden Tabelle wird beschrieben, wie Ereignisse in verschiedenen Szenarien funktionieren. Für diese Szenarien werden die folgenden Annahmen getroffen:

- AWS ist auch ein Partner in diesen Szenarien.
- ResourceSnapshotJob wird vom Partner korrekt konfiguriert.
- Die bei der Opportunity aktualisierten Felder befinden sich auch in der Opportunity-Vorlage für Resource Snapshot.

Aktion von Ihnen als Partner	Empfangene Ereignisse	Art des Teilnehmers
Sie schaffen eine Gelegenheit	Gelegenheit geschaffen	
Sie verwenden <code>StartEngagementFromOpportunityTask</code> , um eine Opportunity einzureichen	Opportunity aktualisiert, Engagement erstellt, Snapshot der Engagement-Ressource erstellt, Engagement-Mitglied hinzugefügt, Engagement-Einladung erstellt	
AWS genehmigt Ihre eingereichte Verkaufschance	Die Einladung zum Engagement wurde akzeptiert, ein Mitglied wurde hinzugefügt, die Verkaufschance wurde aktualisiert, ein Snapshot der Engagement-Ressource wurde erstellt	
AWS lehnt Ihre eingereichte Verkaufschance ab	Einladung zum Engagement abgelehnt, Geschäftschance aktualisiert, Snapshot der Engagement-Ressource erstellt	
Sie verknüpfen ein AWS Marketplace Angebot mit einer Verkaufschance	Verkaufschance aktualisiert	
Sie verbinden eine Lösung mit einer Geschäftschance	Geschäftschance aktualisiert, Momentaufnahme der Engagement-Ressourcen erstellt	
Sie aktualisieren eine Opportunity	Opportunity aktualisiert, Snapshot der Engagement-Ressource erstellt (optional , wenn sie eingerichtet	

Aktion von Ihnen als Partner	Empfangene Ereignisse	Art des Teilnehmers
	ResourceSnapshotJob ist und die Feldaktualisierungen in der Vorlage enthalten sind)	
AWS aktualisiert Ihre Verkaufschance	Opportunity aktualisiert, Snapshot der Engagement-Ressourcen erstellt	
Sie laden einen anderen Partner zu einem Engagement ein	Einladung zur Verlobung erstellt	Absender
Ihre Einladung zur Teilnahme an einem Engagement wurde von einem Partner akzeptiert	Einladung zum Engagement akzeptiert, Engagement-Mitglied hinzugefügt	Absender
Ihre Einladung zur Teilnahme an einem Engagement wurde von einem Partner abgelehnt	Einladung zum Engagement abgelehnt	Absender
Ihr Partner reagiert 15 Tage lang nicht auf Ihre Einladung zur Teilnahme an einem Engagement	Die Einladung zum Engagement ist abgelaufen	Absender
Sie erhalten eine Einladung von einem anderen Partner, an einem Engagement teilzunehmen	Einladung zum Engagement erstellt	Empfänger

Aktion von Ihnen als Partner	Empfangene Ereignisse	Art des Teilnehmers
Sie nehmen die Einladung eines anderen Partners zur Teilnahme an einem Engagement an über <code>StartEngagementByAcceptingInvitationTask</code>	Einladung zum Engagement angenommen, Engagement-Mitglied hinzugefügt, Opportunity erstellt, Opportunity aktualisiert, Snapshot der Engagement-Ressource erstellt	Empfänger
Sie lehnen die Einladung eines anderen Partners zur Teilnahme an einem Engagement ab	Einladung zum Engagement abgelehnt	Empfänger
Sie folgen 15 Tage lang keiner Einladung eines anderen Partners, an einem Engagement teilzunehmen	Die Einladung zum Engagement ist abgelaufen	Empfänger
Sie erstellen ein Engagement	Engagement geschaffen	Absender
Sie aktualisieren ein Engagement	Engagement aktualisiert	Absender, Empfänger

Beispielereignisse

Die folgenden Abschnitte enthalten Beispiele für die Ereignisse, die zuvor im vorherigen Abschnitt aufgeführt wurden.

Themen

- [Gelegenheit geschaffen](#)
- [Opportunity aktualisiert](#)
- [Einladung zur Verlobung erstellt](#)
- [Einladung zur Verlobung akzeptiert](#)
- [Einladung zur Verlobung abgelehnt](#)
- [Die Einladung zum Engagement ist abgelaufen](#)

- [Engagement-Mitglied hinzugefügt](#)
- [Snapshot der Engagement-Ressource wurde erstellt](#)
- [Engagement wurde erstellt](#)
- [Engagement aktualisiert](#)

Gelegenheit geschaffen

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer darüber, dass eine neue Opportunity erstellt wurde.
- Lösen Sie automatisierte Workflows aus, z. B. die Aktualisierung von CRM-Systemen oder die Benachrichtigung von Vertriebsteams über neue Geschäftschancen.

Das folgende Beispiel zeigt ein typisches Ereignis vom Typ Opportunity Created.

```
{
  "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"
    }
  }
}
```

Opportunity aktualisiert

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer darüber, dass eine bestehende Opportunity aktualisiert wurde.
- Lösen Sie automatisierte Workflows aus, z. B. die Aktualisierung von CRM-Systemen oder die Benachrichtigung von Vertriebsteams.

Das folgende Beispiel zeigt ein typisches Opportunity Updated-Ereignis.

```
{
  "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"
    }
  }
}
```

Einladung zur Verlobung erstellt

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer über die Neuigkeiten, die EngagementInvitation sie erhalten haben.
- Lösen Sie automatisierte Workflows aus, um die Einladung anzunehmen oder abzulehnen, z. B. um CRM-Systeme zu aktualisieren oder Vertriebsteams zu benachrichtigen.

Das folgende Beispiel zeigt ein typisches Ereignis, bei dem eine Einladung erstellt wurde.

```
{
  "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
    }
  }
}

```

Einladung zur Verlobung akzeptiert

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer darüber, dass das akzeptiert EngagementInvitation wurde.
- Aktualisieren Sie ihre internen Aufzeichnungen, um den neuen Partner im Engagement widerzuspiegeln.
- Lösen Sie automatisierte Workflows aus, um Daten zu synchronisieren oder andere Teammitglieder über das neue Engagement-Mitglied zu informieren.

Das folgende Beispiel zeigt ein typisches Ereignis, bei dem eine Einladung zur Teilnahme angenommen wurde.

```

{
  "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
    }
  }
}

```

Einladung zur Verlobung abgelehnt

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer darüber, dass das abgelehnte EngagementInvitation wurde.
- Aktualisieren Sie ihre internen Aufzeichnungen, um die abgelehnte Einladung widerzuspiegeln.
- Initiieren Sie alle erforderlichen Workflows, z. B. die Benachrichtigung des Vertriebsteams über die Ablehnung.

Das folgende Beispiel zeigt ein typisches Ereignis, bei dem eine Einladung zur Teilnahme abgelehnt wurde.

```

{
  "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
    }
}
}

```

Die Einladung zum Engagement ist abgelaufen

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer darüber, dass das EngagementInvitation abgelaufen ist und keine weiteren Aktionen mehr möglich sind.
- Aktualisieren Sie ihre internen Aufzeichnungen, um die abgelaufene Einladung widerzuspiegeln.
- Initiieren Sie alle erforderlichen Workflows, z. B. die Benachrichtigung des Vertriebsteams über die abgelaufene Einladung.

Das folgende Beispiel zeigt ein typisches Ereignis, bei dem die Einladung zur Teilnahme abgelaufen ist.

```

{
  "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
  }
}
}

```

Engagement-Mitglied hinzugefügt

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer über die Änderungen an der Engagement-Mitgliedschaft.
- Aktualisieren Sie ihre internen Aufzeichnungen, um das neue Engagement-Mitglied widerzuspiegeln.
- Initiieren Sie alle erforderlichen Workflows, z. B. die Benachrichtigung der Teammitglieder über die Änderungen.

Das folgende Beispiel zeigt ein typisches Ereignis, bei dem das Engagement Member Added stattfindet.

```

{
  "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"
    }
  }
}

```

```
}
}
```

Snapshot der Engagement-Ressource wurde erstellt

Partner nutzen dieses Ereignis, um Änderungen an Ressourcen im Zusammenhang mit einem Engagement nachzuverfolgen und automatisierte Workflows wie die Aktualisierung von CRM-Systemen oder die Benachrichtigung von Vertriebsteams auszulösen. Die Snapshot-Vorlage bestimmt die Art der Aktualisierung:

OpportunitySummaryView

- Informieren Sie alle Mitglieder eines Engagements darüber, dass das Engagement aktualisiert wurde.

AwsOpportunitySummaryFullView

- Verfolgen Sie Updates, die speziell von AWS zu Opportunities im Rahmen eines Engagements vorgenommen wurden.
- Diese Benachrichtigung und der Snapshot sind nur für den Eigentümer der Opportunity sichtbar.
- Beinhaltet Aktualisierungen zur Angebotsgröße, die nicht über die OpportunitySummaryView Vorlage verfügbar sind.

Das folgende Beispiel zeigt ein typisches Ereignis vom Typ Engagement Resource Snapshot Created.

```
{
  "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" : "0123211231",
    "createdBy": "123123123123",
    "targetMemberAccounts": ["123123123123", "aws"],
    "resourceSnapshotTemplateName": "temp-name"
  }
}
}

```

Engagement wurde erstellt

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer darüber, dass ein neues Engagement erstellt wurde.
- Lösen Sie automatisierte Workflows aus, z. B. die Aktualisierung von CRM-Systemen oder die Benachrichtigung von Vertriebsteams über neue Interaktionen.

Das folgende Beispiel zeigt ein typisches „Engagement Created“-Ereignis.

```

{
  "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 aktualisiert

Partner nutzen diese Veranstaltung, um:

- Informieren Sie ihre Benutzer darüber, dass ein vorhandenes Engagement aktualisiert wurde.
- Lösen Sie automatisierte Workflows aus, z. B. die Aktualisierung von CRM-Systemen oder die Benachrichtigung von Vertriebsteams.

Das folgende Beispiel zeigt ein typisches Engagement Updated-Ereignis.

```
{  
  "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"]  
    }  
  }  
}
```

Benachrichtigungen für AWS API für Vorteile von Partner Central

Der Benefits Service bietet EventBridge Veranstaltungen BenefitApplications sowohl für als auch BenefitAllocations. Diese Veranstaltungen ermöglichen es Kunden, umfassende, ereignisgesteuerte Architekturen aufzubauen, die auf Änderungen während des gesamten Ressourcenlebenszyklus reagieren — von der ersten Anwendung über die Zuweisung bis hin zur endgültigen Ausführung.

Themen

- [Erfüllen Sie die Voraussetzungen für die Überwachung von Ereignissen](#)
- [Amazon für EventBridge die Überwachung von Ereignissen konfigurieren](#)
- [Erfahren Sie mehr über API-Veranstaltungen zu den Vorteilen](#)

Erfüllen Sie die Voraussetzungen für die Überwachung von Ereignissen

Benutzer benötigen die entsprechenden IAM-Berechtigungen, um auf Ereignisse zuzugreifen und diese zu verwalten, die von der Benefits API veröffentlicht wurden. Weitere Informationen zu den verfügbaren Aktionen, Ressourcen und Bedingungsschlüsseln für EventBridge finden Sie unter [Verwenden von IAM-Richtlinienbedingungen in Amazon EventBridge](#) im EventBridge Amazon-Benutzerhandbuch. Einer der Bedingungsschlüssel ist `events:detail-type`, der verwendet werden kann, um Berechtigungen auf bestimmte Ereignistypen zu beschränken.

Die folgende Beispielrichtlinie zeigt, wie Sie die Berechtigungen für die vorgeschlagenen Ereignisse anpassen und einschränken können. Die `AllowPutRuleForPartnerCentralBenefitsEvents` Anweisung ermöglicht die Erstellung von Regeln, jedoch nur für Ereignisse aus der `aws.partnercentral-benefits` Quelle.

Ausführliche Beispiele für IAM-Richtlinien finden Sie in der AWS-Dokumentation zu EventBridge Berechtigungen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralBenefitsEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```
        "events:source": "aws.partnercentral-benefits"
    }
}
]
```

Amazon für EventBridge die Überwachung von Ereignissen konfigurieren

Um API-Ereignisse für Leistungen zu überwachen, erstellen Sie eine EventBridge Regel, die den Ereignissen entspricht, die Sie erfassen möchten. Sie können die AWS-Managementkonsole oder die AWS SDKs verwenden, um Regeln zu erstellen und zu verwalten. In den folgenden Abschnitten wird erklärt, wie Regeln mit beiden Methoden erstellt werden. Unabhängig von der verwendeten Methode müssen Sie die Regel in der us-east-1 Region USA Ost (Nord-Virginia) erstellen.

AWS-Managementkonsole Einrichten

Um eine EventBridge Regel mit der einzurichten AWS-Managementkonsole, folgen Sie den Schritten unter [Erstellen von Regeln, die auf Ereignisse EventBridge in Amazon reagieren](#) im EventBridge Amazon-Benutzerhandbuch. Bei der Erstellung der Regel müssen Sie den Event-Bus auf die Standardeinstellung setzen und die Regel in der us-east-1 Region USA Ost (Nord-Virginia) erstellen.

Im Folgenden finden Sie ein Beispiel für eine Ereignisregel:

```
{
  "source": ["aws.partnercentral-benefits"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS SDK-Setup

Sie können die AWS SDKs verwenden, um EventBridge Regeln programmgesteuert zu erstellen und zu verwalten. Weitere Informationen finden Sie [PutRule](#) in der Amazon EventBridge API-Referenz.

Das folgende Beispiel verwendet das AWS SDK for 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'])
```

Erfahren Sie mehr über API-Veranstaltungen zu den Vorteilen

In den folgenden Abschnitten werden die Vorteile von API-Ereignistypen, Szenarien, die sie auslösen, und Beispiele für Ereignisse beschrieben.

Event types (Ereignistypen)

Im Folgenden sind die Ereignistypen und ihre Auslöser aufgeführt.

Nutzenanwendungen

- [Leistungsantrag erstellt](#): Wenn ein Leistungsantrag erstellt wird
- [Leistungsantrag wird geprüft](#): Dieses Ereignis wird ausgelöst, wenn ein Leistungsantrag eingereicht wird oder wenn der Antrag von „Maßnahme erforderlich“ zurück zu „Wird geprüft“ wechselt.
- [Maßnahme im Leistungsantrag erforderlich](#): Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status des Antrags aktualisiert, um darauf hinzuweisen, dass der Partner zusätzliche Informationen, Klarstellungen oder Änderungen vorlegen muss, bevor die Prüfung fortgesetzt werden kann.
- [Antrag auf Leistung abgelehnt](#): Dieses Ereignis tritt auf, wenn ein interner Prüfer den Status des Antrags aktualisiert, um anzuzeigen, dass der Antrag die Leistungskriterien oder -anforderungen nicht erfüllt und abgelehnt wurde
- [Leistungsantrag genehmigt](#): Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status des Antrags aktualisiert, um anzuzeigen, dass der Antrag für die Leistungszuweisung genehmigt wurde.
- [Phase der Beantragung von Leistungen geändert](#): Dieses Ereignis tritt auf, wenn ein interner Prüfer den Status des Antrags im Rahmen von Prüfverfahren wie Geschäftsprüfung, AWS Prüfung, Finanzprüfung usw. aktualisiert.

Zuweisung von Leistungen

- [Leistungszuweisung aktiviert](#): Wenn eine Leistungszuweisung erstellt oder durch einen internen Dienst wieder auf Aktiviert aktualisiert wird.
- [Leistungszuweisung inaktiviert](#): Dieses Ereignis tritt auf, wenn ein interner Prüfer den Status der Zuweisung aktualisiert oder wenn eine Leistungszuweisung automatisch abläuft.
- [Leistungszuweisung erfüllt](#): Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status der Allokation aktualisiert, um anzuzeigen, dass die Allokation vollständig ausgeschöpft wurde.

Beispielereignisse

Die folgenden Abschnitte enthalten Beispiele für die zuvor im vorherigen Abschnitt aufgeführten Ereignisse.

Themen

- [Der Antrag auf Leistung wurde erstellt](#)
- [Der Antrag auf Leistungen wird geprüft](#)
- [Maßnahme zur Beantragung von Leistungen erforderlich](#)
- [Antrag auf Leistung abgelehnt](#)
- [Antrag auf Leistungen genehmigt](#)
- [Die Phase der Beantragung von Leistungen wurde geändert](#)
- [Die Zuweisung von Leistungen ist aktiviert](#)
- [Zuweisung von Leistungen inaktiviert](#)
- [Die Zuweisung der Leistungen ist abgeschlossen](#)

Der Antrag auf Leistung wurde erstellt

Auslösender Kontext

Das Ereignis „Benefit Application Created“ wird auf Amazon veröffentlicht EventBridge, wenn ein neuer Leistungsantrag erfolgreich über den Benefits Service erstellt wurde. Dieses Ereignis tritt ein, wenn der CreateBenefitApplication API-Aufruf erfolgreich abgeschlossen wurde, was darauf hinweist, dass ein Partner eine Anfrage für eine bestimmte Leistung gestellt hat.

Dieses Ereignis stellt den Beginn des Lebenszyklus der Leistungsanträge dar und informiert Kunden sofort, wenn neue Anträge auf ihrem Konto eingehen.

Empfänger

Das Konto, für das der Eigentümer zuständig ist, erhält die Veranstaltung für die Gewährung von Leistungen.

Beispiel

```
{
  "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"]
    }
  }
}
```

Der Antrag auf Leistungen wird geprüft

Auslösender Kontext

Das Ereignis „Antrag auf Leistungen wird geprüft“ wird auf Amazon veröffentlicht, EventBridge wenn ein Leistungsantrag über den Benefits Service in den Status IN_REVIEW übergeht. Dieses Ereignis tritt ein, wenn:

- Ein Partner reicht seine Bewerbung über den SubmitBenefitApplication API-Aufruf ein

- Eine Anwendung wechselt von ACTION_REQUIRED zurück zu IN_REVIEW, nachdem der Partner auf Feedback reagiert hat
- Eine Bewerbung wird zurückgerufen und dann erneut eingereicht

Dieses Ereignis weist darauf hin, dass der Antrag in das formelle Prüfverfahren eingetreten ist und von AWS internen Teams geprüft wird.

Empfänger

Der Eigentümer erhält die Veranstaltung für die Benefit Allocation.

Beispiel

```
{
  "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"]
    }
  }
}
```

Maßnahme zur Beantragung von Leistungen erforderlich

Auslösender Kontext

Das Ereignis „Maßnahme für Leistungsantrag erforderlich“ wird auf Amazon veröffentlicht, EventBridge wenn ein Leistungsantrag über den Benefits Service in den Status ACTION_REQUIRED übergeht. Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status des Antrags über die UpdateBenefitApplicationInternal API aktualisiert, um darauf hinzuweisen, dass der Partner zusätzliche Informationen, Klarstellungen oder Änderungen bereitstellen muss, bevor die Prüfung fortgesetzt werden kann.

Dieses Ereignis stellt einen kritischen Punkt im Anwendungslebenszyklus dar, an dem ein Eingreifen des Partners erforderlich ist, um den Antrag im Überprüfungsprozess voranzubringen.

Empfänger

Das Konto, für das der Eigentümer verantwortlich ist, erhält die Veranstaltung im Rahmen der Benefit-Allokation

Beispiel

```
{
  "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"]  
  }  
}  
}
```

Antrag auf Leistung abgelehnt

Auslösender Kontext

Das Ereignis „Antrag auf Leistung abgelehnt“ wird auf Amazon veröffentlicht, EventBridge wenn ein Leistungsantrag über den Benefits Service in den Status ABGELEHNT übergeht. Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status des Antrags über die UpdateBenefitApplicationInternal API aktualisiert, um anzuzeigen, dass der Antrag die Leistungskriterien oder Anforderungen nicht erfüllt und abgelehnt wurde.

Dieses Ereignis stellt einen Endzustand im Lebenszyklus eines Antrags dar, in dem der Antrag formell abgelehnt wurde und nicht für die Zuweisung von Leistungen verwendet wird.

Empfänger

Das Ereignis für die Zuweisung von Leistungen erhält das Konto, für das der Eigentümer zuständig ist.

Beispiel

```
{  
  "id": "0kl111098-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"]
  }
}
}

```

Antrag auf Leistungen genehmigt

Auslösender Kontext

Die Veranstaltung Leistungsantrag genehmigt wird auf Amazon veröffentlicht, EventBridge wenn ein Leistungsantrag über den Benefits Service in den Status GENEHMIGT übergeht. Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status des Antrags über die UpdateBenefitApplicationInternal API aktualisiert, um anzugeben, dass der Antrag alle Leistungskriterien und -anforderungen erfüllt und für die Zuweisung von Leistungen genehmigt wurde.

Dieses Ereignis stellt einen erfolgreichen Abschluss des Antragsprüfungsprozesses dar und löst in der Regel die Erstellung einer entsprechenden Leistungszuweisung aus.

Empfänger

Das Konto, für das der Eigentümer zuständig ist, erhält die Veranstaltung für die Leistungszuweisung.

Beispiel

```

{
  "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": "l1l1i2844-i63f-85h8-e150-880099884456"
  },
  "benefit": {
    "id": "ben-xyz789uvw012",
    "name": "AWS Partner Credits",
    "program": ["AWS Partner Network"]
  }
}
}

```

Die Phase der Beantragung von Leistungen wurde geändert

Auslösender Kontext

Das Ereignis „Phase der Änderung des Leistungsantrags“ wird auf Amazon veröffentlicht EventBridge, wenn die Überprüfungsphase eines Leistungsantrags über den Benefits Service aktualisiert wird. Dieses Ereignis tritt ein, wenn ein interner Prüfer die Phase des Antrags auf einen der möglichen Enum-Werte aktualisiert (FINANCE_REVIEW, BUSINESS_REVIEW, TECH_REVIEW und _REVIEW).AWS

Im Gegensatz zu Statusänderungen handelt es sich bei Phasenänderungen um schreibgeschützte Informationsaktualisierungen, die Einblick in den Verlauf des internen Prüfungsablaufs bieten, ohne dass ein Eingreifen des Partners erforderlich ist.

Empfänger

Der Eigentümer erhält die Veranstaltung im Rahmen der Benefit Allocation.

Beispiel

```

{
  "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"]
  }
}
}

```

Die Zuweisung von Leistungen ist aktiviert

Auslösender Kontext

Das Ereignis „Benefit Allocation Activated“ wird auf Amazon veröffentlicht EventBridge , wenn eine Benefit Allocation über den Benefits Service in den Status AKTIV übergeht. Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status der Zuteilung über die UpdateBenefitAllocationInternal API aktualisiert, um eine zuvor inaktive Zuteilung wieder zu aktivieren.

Dieses Ereignis tritt in der Regel auf, wenn eine Zuteilung, die vorübergehend inaktiviert wurde (aufgrund von Ablauf, Richtlinienänderungen oder administrativen Gründen), wieder in den aktiven Status versetzt wird, sodass die Vorteile wieder für Partner verfügbar sind.

Empfänger

Das AWS Konto des Partners erhält die Nachricht. Mit anderen Worten, das AWS Konto, dem die Ressource gehört.

Beispielereignis

```

{
  "id": "3no44321-q6pn-j64m-o858-8117352s696t",
  "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"]
    }
  }
}
```

Zuweisung von Leistungen inaktiviert

Auslösender Kontext

Das Ereignis Vorteilsallokation inaktiviert wird auf Amazon veröffentlicht, EventBridge wenn eine Leistungszuweisung über den Benefits Service in den Status INAKTIV übergeht. Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status der Zuteilung über die UpdateBenefitAllocationInternal API aktualisiert oder wenn eine Leistungszuweisung automatisch abläuft.

Dieses Ereignis tritt in der Regel auf, wenn eine Zuteilung vorübergehend ausgesetzt wird (aufgrund von Ablauf, Richtlinienänderungen, Compliance-Problemen oder administrativen Gründen), sodass die Leistungen erst nach einer erneuten Aktivierung für Partner in Anspruch genommen werden können.

Empfänger

Das AWS Konto des Partners erhält die Nachricht. Mit anderen Worten, das AWS Konto, dem die Ressource gehört.

Beispielereignis

```
{
  "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"]
    }
  }
}
```

Die Zuweisung der Leistungen ist abgeschlossen

Auslösender Kontext

Das Ereignis „Benefit Allocation Fulfilled“ wird auf Amazon veröffentlicht EventBridge , wenn eine Benefit Allocation über den Benefits Service in den Status FULFILLED übergeht. Dieses Ereignis tritt ein, wenn ein interner Prüfer den Status der Zuteilung über die UpdateBenefitAllocationInternal API aktualisiert, um anzuzeigen, dass das Kontingent vollständig ausgeschöpft wurde oder dass die Leistungsbedingungen erfüllt wurden.

Dieses Ereignis stellt den Abschluss des Leistungslebenszyklus dar und weist darauf hin, dass der Partner die zugewiesene Leistung erfolgreich in Anspruch genommen hat oder dass die Zuweisung ihren natürlichen Abschluss erreicht hat.

Empfänger

Das AWS Konto des Partners erhält die Nachricht. Mit anderen Worten, das AWS Konto, dem die Ressource gehört.

Beispielereignis

```
{
  "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"]
    }
  }
}
```

Benachrichtigungen für AWS Channel-API für Partner Central

Ereignisse für die Kanal-API bieten Benachrichtigungen in Echtzeit über Änderungen des Status oder der Details kanalbezogener Aktivitäten. Diese Ereignisse tragen dazu bei, dass Ihre Systeme mit AWS Partner Central synchron bleiben und dass zeitnahe Antworten und Aktualisierungen gewährleistet sind.

Voraussetzungen

Sie benötigen die entsprechenden IAM-Berechtigungen, um über die AWS Partner Central Channel API auf Ereignisse zuzugreifen und diese zu verwalten. Informationen zu den verfügbaren Aktionen, Ressourcen und Bedingungsschlüsseln für EventBridge finden Sie unter [Verwenden von IAM-Richtlinienbedingungen in Amazon EventBridge](#) im EventBridge Amazon-Benutzerhandbuch. Einer der Bedingungsschlüssel ist `events:detail-type`, mit dem Sie Berechtigungen auf bestimmte Ereignistypen beschränken können.

Die folgende Beispielrichtlinie zeigt, wie Sie die Berechtigungen für die Ereignisse anpassen und einschränken können. Die `AllowPutRuleForPartnerCentralChannelEvents` Anweisung ermöglicht die Erstellung von Regeln, jedoch nur für Ereignisse aus der `aws.partnercentral-channel` Quelle.

Ausführliche Beispiele für IAM-Richtlinien finden Sie in der AWS-Dokumentation zu EventBridge Berechtigungen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralChannelEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "events:source": "aws.partnercentral-channel"
        }
      }
    }
  ]
}
```

Amazon für EventBridge die Überwachung von Ereignissen konfigurieren

Um Channel-API-Ereignisse zu überwachen, erstellen Sie eine EventBridge Regel, die den Ereignissen entspricht, die Sie erfassen möchten. Sie können die AWS-Managementkonsole oder die AWS SDKs verwenden, um Regeln zu erstellen und zu verwalten. In den folgenden Abschnitten wird erklärt, wie Regeln mit beiden Methoden erstellt werden. Unabhängig von der verwendeten Methode müssen Sie die Regel in der us-east-1 Region USA Ost (Nord-Virginia) erstellen.

AWS-Managementkonsole Einrichten

Um eine EventBridge Regel mit der einzurichten AWS-Managementkonsole, folgen Sie den Schritten unter [Erstellen von Regeln, die auf Ereignisse EventBridge in Amazon reagieren](#) im EventBridge Amazon-Benutzerhandbuch. Bei der Erstellung der Regel müssen Sie den Event-Bus auf die Standardeinstellung setzen und die Regel in der us-east-1 Region USA Ost (Nord-Virginia) erstellen.

Im Folgenden finden Sie ein Beispiel für eine Ereignisregel:

```
{
  "source": ["aws.partnercentral-channel"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS SDK-Setup

Sie können die AWS SDKs verwenden, um EventBridge Regeln programmgesteuert zu erstellen und zu verwalten. Weitere Informationen finden Sie [PutRule](#) in der Amazon EventBridge API-Referenz.

Das folgende Beispiel verwendet das AWS SDK for 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'])

```

Erfahre mehr über Channel-API-Ereignisse

In den folgenden Abschnitten werden die Ereignistypen der Kanal-API, Szenarien, die sie auslösen, und Beispiele für Ereignisse beschrieben.

Event types (Ereignistypen)

Im Folgenden sind die Ereignistypen und ihre Auslöser aufgeführt.

- [Kanal-Handshake erstellt](#): Wird ausgelöst, wenn ein neuer Kanal-Handshake erstellt wird.
- [Kanal-Handshake akzeptiert](#): Wird ausgelöst, wenn ein neuer Kanal-Handshake akzeptiert wird.
- [Kanal-Handshake abgelehnt](#): Wird ausgelöst, wenn ein neuer Kanal-Handshake abgelehnt wird.

Kanal-Handshake wurde erstellt

Die folgenden Beispiele zeigen typische Channel Handshake Created-Ereignisse für verschiedene Handshake-Typen.

Starten Sie den Service Period 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-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"
      }
    }
  }
}

```

Handshake für den Servicezeitraum widerrufen:

```

{
  "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"
      }
    }
  }
}

```

Kontohandshake für die Programmverwaltung:

```

{
  "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"
      }
    }
  }
}

```

```
}
```

Kanal-Handshake akzeptiert

Die folgenden Beispiele zeigen typische Channel Handshake Accepted-Ereignisse für verschiedene Handshake-Typen.

Starten Sie den Service Period 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-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"
      }
    }
  }
}
```

Handshake für den Servicezeitraum widerrufen:

```
{
  "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"
      }
    }
  }
}
```

Kontohandshake für die Programmverwaltung:

```
{
  "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"
        }
      }
    }
  }
}

```

Kanal-Handshake wurde abgelehnt

Die folgenden Beispiele zeigen typische Channel-Handshake-Ablehnungs-Ereignisse für verschiedene Handshake-Typen.

Starten Sie den Service Period 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-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"
    }
  }
}
}

```

Handshake für den Servicezeitraum widerrufen:

```

{
  "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"
      }
    }
  }
}

```

Kontohandshake für die Programmverwaltung:

```

{
  "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"
      }
    }
  }
}

```

```
}  
  }  
} }
```

Testen in einer Sandbox

Partner können eine Sandbox verwenden, um ihre AWS Partner Central-API-Interaktionen in einer sicheren und isolierten Umgebung zu testen und zu validieren und so einen reibungslosen Betrieb sicherzustellen, bevor ihre Lösung in der Produktionsumgebung eingeführt wird. AWS bietet Benutzern der AWS Partner Central API eine dynamische Sandbox, die Antworten zurückgibt, die denen der Produktionsumgebung ähneln. AWS stellt keine Benutzeroberfläche für die Sandbox-Umgebung bereit. Daher müssen sich Partner beim Testen ihrer Lösungen auf die programmatischen Antworten verlassen.

Zugriff auf die Sandbox-Umgebung

Partner erhalten Zugriff auf die Testumgebung, sobald sie ihr Konto mit AWS-Konto dem Partner Central-Konto verknüpfen. Weitere Informationen finden Sie unter [Ihr AWS Konto mit AWS einem Partner Central-Konto verknüpfen](#). Jede Anfrage enthält einen `catalog` Parameter, der die Datenumgebung bestimmt. Wenn auf `catalog` `istAWS` gesetzt ist, verweist sie auf Produktionsdaten, und wenn sie auf `istSandbox` gesetzt ist, verweist sie auf Sandbox-Daten.

Wichtige Details zur Sandbox-Umgebung

1. **Datenaktualisierung:** Einmal pro Jahr werden AWS die Daten in der Sandbox-Umgebung aktualisiert (normalerweise zu Beginn des Jahres). Nach dieser Aktualisierung können Sie einige Daten in Ihrer Testumgebung verlieren.
2. **Testumfang:** Die Sandbox-Umgebung wird in der Regel für Funktionstests und nicht für Skalierbarkeits- oder Leistungstests verwendet.

Themen

- [Testen in einer Sandbox für die AWS Konto-API für Partner Central](#)
- [Testen in einer Sandbox für AWS Vertriebs-API von Partner Central](#)
- [Testen in einer Sandbox für die AWS API für Vorteile von Partner Central](#)
- [Testen in einer Sandbox für die AWS Channel-API von Partner Central](#)

- [Testen in einer Sandbox für AWS API zur Umsatzmessung in Partner Central](#)

Testen in einer Sandbox für die AWS Konto-API für Partner Central

Wie benutzt man die Sandbox

Gehen Sie wie folgt vor, um die Sandbox zu verwenden:

1. Erstellen Sie eine IAM-Rolle:

Erstellen Sie eine IAM-Rolle in dem mit Ihrem AWS Partner Central AWS-Konto verknüpften Konto.

2. Richtlinie zuweisen:

Weisen Sie der IAM-Rolle die folgende Richtlinie zu. Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

```
{
  "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. Verwenden Sie die Anmeldeinformationen für die IAM-Rolle:

Verwenden Sie die Anmeldeinformationen (geheimer Schlüssel und Zugriffsschlüssel) dieser IAM-Rolle in Ihrer Lösung, um die API-Aktionen auszuführen.

Testen AWS actions

Während der Testphase ist es oft notwendig, AWS Aktionen zu simulieren. Diese Simulation ermöglicht es Partnern, den gesamten Ablauf ihrer Integration mit AWS Diensten gründlich zu testen. Jede Anfrage enthält einen Katalogparameter, der die Datenumgebung bestimmt.

Simulation der Gründung eines Partners

Um die Gründung eines Partners zu simulieren, nehmen Sie ihn in die Payload der CreatePartner Aktion "catalog": "Sandbox" auf.

Beispiel:

```
{
  "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"
}
```

Hinweis: In der Sandbox ist die E-Mail-Bestätigung deaktiviert. Sie können eine beliebige sechsstellige Zahl EmailVerificationCode zum Testen verwenden. Verwenden Sie in der Anfrage auch keine „Tags“.

Zusätzliche Hinweise zum Testen

1. Um andere Aktionen zu testen, legen Sie sie "catalog": "Sandbox" in der Payload fest.
2. Um Verbindungen mit Partnerkonten in der Sandbox zu testen, müssen Sie die StartProfileUpdateTask Aktion ausführen, nachdem Sie den Partner im Sandbox-Katalog erstellt haben.
3. Um zur Produktion überzugehen, ändern Sie den Katalogwert von bisSandbox. AWS

Testen von Ereignissen in der Sandbox-Umgebung

Partner können Ereignisse aus der Sandbox-Umgebung nutzen, um die ereignisbasierten Implementierungen zu testen. Richten Sie EventBridge in derselben AWS-Konto Weise Regeln ein, um auf Sandbox-Ereignisse zu warten, indem Sie dies `catalog: Sandbox` in den Ereignisdetails angeben. Weitere Informationen finden Sie unter [Benachrichtigungen für AWS Konto-API für Partner Central](#).

Beispiel für eine Ereignisregel:

```
{
  "source": ["aws.partnercentral-account"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

Ereignisregeln, die angeben `catalog`, dass nur Ereignisse berücksichtigt Sandbox werden, die aus der Sandbox stammen und durch die Aktionen generiert wurden, die Sie in der Sandbox-Umgebung ausführen.

Testen in einer Sandbox für AWS Vertriebs-API von Partner Central

Wie benutzt man die Sandbox

1. Erstellen Sie eine IAM-Rolle:

Erstellen Sie eine IAM-Rolle in dem mit Ihrem AWS Partner Central AWS-Konto verknüpften Konto.

2. Richtlinie zuweisen:

Weisen Sie der IAM-Rolle die folgende Richtlinie zu. Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

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. Verwenden Sie die Anmeldeinformationen für die IAM-Rolle:

Verwenden Sie die Anmeldeinformationen (geheimer Schlüssel und Zugriffsschlüssel) dieser IAM-Rolle in Ihrer Lösung, um die API-Aktionen auszuführen.

Testen AWS actions

Während der Testphase ist es oft notwendig, AWS Aktionen zu simulieren. Diese Simulation ermöglicht es Partnern, den gesamten Ablauf ihrer Integration mit AWS Diensten gründlich zu testen.

Simuliert die Entstehung eines AWS entstandene Gelegenheit

Um zu simulieren, dass eine AWS entstandene Gelegenheit entsteht, fügen Sie in die Nutzlast der CreateOpportunity Aktion "Catalog": "Sandbox" und "Origin": "AWS Referral" ein.

Beispiel:

```

{
  "Catalog": "Sandbox",
  "Origin": "AWS Referral",
  "OpportunityIdentifier": "0123456",
  "Title": "Test Opportunity",
  ...
}

```

Simulation von Aktualisierungen einer Opportunity, die von einem Partner initiiert wurde

Um Aktualisierungen oder andere Aktionen für eine von einem Partner initiierte Opportunity zu simulieren, verwenden Sie die UpdateOpportunity Aktion mit "Catalog": "Sandbox" und Lifecycle.ReviewStatus: "Approved" oder in der Payload. "Action Required"

Wenn Sie die Payload aus der GetOpportunity Aktion übernehmen, um das Update durchzuführen, stellen Sie sicher, dass Sie die ID in ändern und die Identifier folgenden Felder entfernen:

- CreatedDate
- OpportunityTeam
- RelatedEntityIdentifiers

Beispiel:

```
{
  "Catalog": "Sandbox",
  "Identifier": "0123456",
  "Title": "Updated Test Opportunity",
  "Lifecycle": {
    "ReviewStatus": "Approved"
  }
  ...
}
```

Ereignisse in der Sandbox-Umgebung testen

Partner können Opportunity-Ereignisse aus der Sandbox-Umgebung nutzen, um die ereignisbasierten Implementierungen zu testen. Richten Sie EventBridge in derselben AWS-Konto Weise Regeln ein, um auf Sandbox-Ereignisse zu warten, indem Sie dies `catalog: sandbox` in den Ereignisdetails angeben. Weitere Informationen finden Sie unter [Benachrichtigungen für die AWS Vertriebs-API von Partner Central](#).

Beispiel für eine Ereignisregel:

```
{
  "source": ["aws.partnercentral-selling"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

Ereignisregeln, die angeben `catalog`, dass nur Ereignisse berücksichtigt werden, die aus der Sandbox stammen und durch die Aktionen generiert wurden, die Sie in der Sandbox-Umgebung ausführen.

Zusätzliche Hinweise zum Testen

1. AssociateOpportunityAktion testen:

- a. Verwenden Sie die Standardlösung "S-1234567" zum Testen der AssociateOpportunity Aktion.
- b. Verwenden Sie zum Testen von Marketplace-Angeboten eine echte Angebots-ID aus Ihrem Konto.

2. Um zur Produktion überzugehen, ändern Sie den `catalog` Wert von Sandbox bis AWS.

Hilfe erhalten

Wenn Sie bei der Integration Ihres CRM auf Probleme stoßen oder wenn Sie ein bestimmtes Szenario testen müssen, wenden Sie sich bitte an den Support, indem Sie über die folgenden Schritte einen Fall einreichen:

1. Melden Sie sich mit Ihren [AWS Partner Network-Anmeldeinformationen bei AWS Partner Central](#) an.
2. Wählen Sie im [Support Center für Partner Central](#) aus, ob Open New Case Sie einen neuen Fall protokollieren möchten. Füllen Sie die Felder wie folgt aus:
 - a. Art des Support-Falls: AWS Partner Central.
 - b. Frage zu: Partner Central Tools or ACE leads and opportunities.
 - c. Spezifisch werden: Wählen Sie den am besten geeigneten Falltyp für die CRM-Integration aus.
 - d. Betreff: Fügen Sie eine kurze Beschreibung der Anfrage bei.
 - e. Beschreibung: Geben Sie eine detaillierte Beschreibung der Probleme, Fragen, Fehler und Schritte zur Fehlerbehebung an.
 - f. Anlagen: Fügen Sie gegebenenfalls Protokolle und Screenshots bei.

Testen in einer Sandbox für die AWS API für Vorteile von Partner Central

Wie benutzt man die Sandbox

Gehen Sie wie folgt vor, um die Sandbox zu verwenden:

1. Erstellen Sie eine IAM-Rolle:

Erstellen Sie eine IAM-Rolle in dem mit Ihrem AWS Partner Central AWS-Konto verknüpften Konto.

2. Richtlinie zuweisen:

Weisen Sie der IAM-Rolle die folgende Richtlinie zu. Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

```
{
  "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:DisassociateBenefitApplicationResource",
      "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. Verwenden Sie die Anmeldeinformationen für die IAM-Rolle:

Verwenden Sie die Anmeldeinformationen (geheimer Schlüssel und Zugriffsschlüssel) dieser IAM-Rolle in Ihrer Lösung, um die API-Aktionen auszuführen.

Testen AWS actions

Während der Testphase ist es oft notwendig, AWS Aktionen zu simulieren. Diese Simulation ermöglicht es Partnern, den gesamten Ablauf ihrer Integration mit AWS Diensten gründlich zu testen. Jede Anfrage enthält einen Katalogparameter, der die Datenumgebung bestimmt.

Simulation der Erstellung eines Benefit-Antrags

Um die Erstellung eines Leistungsantrags zu simulieren, nehmen Sie ihn in die Nutzlast der `CreateBenefitApplication` Maßnahme "catalog": "Sandbox" auf.

Beispiel:

```
{
  "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"
    }
  ]
}
```

Zusätzliche Hinweise zum Testen

1. Um andere Aktionen zu testen, legen Sie sie "catalog": "Sandbox" in der Payload fest.
2. Die Benefit-IDs im Sandbox-Katalog und im AWS-Katalog sind unterschiedlich. Führen Sie einen GetBenefit API-Aufruf durch "catalog": "Sandbox", um die Benefit Identifier in der Sandbox abzurufen.
3. Die Zuweisung der Vorteile im Sandbox-Katalog und im AWS-Katalog ist unterschiedlich.
4. Um zur Produktion überzugehen, ändern Sie den Katalogwert von Sandbox bis AWS.

Testen von Ereignissen in der Sandbox-Umgebung

Partner können Ereignisse aus der Sandbox-Umgebung nutzen, um die ereignisbasierten Implementierungen zu testen. Richten Sie EventBridge in derselben AWS-Konto Weise Regeln ein, um auf Sandbox-Ereignisse zu warten, indem Sie dies catalog: Sandbox in den Ereignisdetails angeben. Weitere Informationen finden Sie unter [Benachrichtigungen für AWS API für Vorteile von Partner Central](#).

Beispiel für eine Ereignisregel:

```
{
  "source": ["aws.partnercentral-benefits"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

Ereignisregeln, die angeben catalog, dass nur Ereignisse berücksichtigt Sandbox werden, die aus der Sandbox stammen und durch die Aktionen generiert wurden, die Sie in der Sandbox-Umgebung ausführen.

Testen in einer Sandbox für die AWS Channel-API von Partner Central

Konten in der Sandbox kommen nicht für Channel-Vorteile in Frage und wurden nicht für alle Programmanforderungen validiert.

Wie benutzt man die Sandbox

Gehen Sie wie folgt vor, um die Sandbox zu verwenden:

1. Erstellen Sie eine IAM-Rolle:

Erstellen Sie eine IAM-Rolle in dem mit Ihrem AWS Partner Central AWS-Konto verknüpften Konto.

2. Richtlinie zuweisen:

Weisen Sie der IAM-Rolle die folgende Richtlinie zu. Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

```
{
  "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. Verwenden Sie die Anmeldeinformationen für die IAM-Rolle:

Verwenden Sie die Anmeldeinformationen (geheimer Schlüssel und Zugriffsschlüssel) dieser IAM-Rolle in Ihrer Lösung, um die API-Aktionen auszuführen.

Testen AWS actions

Während der Testphase ist es oft notwendig, AWS Aktionen zu simulieren. Diese Simulation ermöglicht es Partnern, den gesamten Ablauf ihrer Integration mit AWS Diensten gründlich zu testen. Jede Anfrage enthält einen Katalogparameter, der die Datenumgebung bestimmt.

Simulation der Erstellung eines Programmverwaltungskontos

Um die Erstellung eines Programmverwaltungskontos zu simulieren, nehmen Sie es in die Payload der `CreateProgramManagementAccount` Aktion `"catalog": "Sandbox"` auf.

Beispiel:

```

{
  "catalog": "Sandbox",
  "accountId": "123456789012",

```

```
"displayName": "Test PMA Name",
"clientToken": "123abc456def789ghi",
"program": "SOLUTION_PROVIDER",
"tags": [
  {
    "key": "testkey",
    "value": "testvalue"
  }
]
```

Simulieren von Aktualisierungen einer Beziehung

Um Aktualisierungen einer Beziehung zu simulieren, verwenden Sie die `UpdateRelationship` Aktion mit `"catalog": "Sandbox"` in der Payload.

Beispiel:

```
{
  "catalog": "Sandbox",
  "displayName": "Updated Test PMA",
  "identifier": "rs-abc123def456g",
  "programManagementAccountIdentifier": "pma-123abc456def7"
}
```

Zusätzliche Hinweise zum Testen

1. Um andere Aktionen zu testen, legen Sie `"catalog": "Sandbox"` in der Payload fest.
2. Um zur Produktion überzugehen, ändern Sie den Katalogwert von `Sandbox` auf `AWS`.

Testen von Ereignissen in der Sandbox-Umgebung

Partner können Ereignisse aus der Sandbox-Umgebung nutzen, um die ereignisbasierten Implementierungen zu testen. Richten Sie EventBridge in derselben AWS-Konto Weise Regeln ein, um auf Sandbox-Ereignisse zu warten, indem Sie `catalog: Sandbox` in den Ereignisdetails angeben. Weitere Informationen finden Sie unter [Benachrichtigungen für AWS Channel-API für Partner Central](#).

Beispiel für eine Ereignisregel:

```
{
```

```
"source": ["aws.partnercentral-channel"],
"detail": {
  "catalog": ["Sandbox"]
}
}
```

Ereignisregeln, die angeben `catalog`, dass nur Ereignisse berücksichtigt werden, die aus der Sandbox stammen und durch die Aktionen generiert wurden, die Sie in der Sandbox-Umgebung ausführen.

Hilfe erhalten

Wenn Sie beim Testen auf Probleme stoßen oder wenn Sie ein bestimmtes Szenario testen müssen, das hier nicht behandelt wird, wenden Sie sich bitte an den Support, indem Sie über die folgenden Schritte einen Fall einreichen:

1. Melden Sie sich mit Ihren [AWS Partner Network-Anmeldeinformationen bei AWS Partner Central](#) an.
2. Wählen Sie im [Support Center für Partner Central](#) aus, ob `Open New Case` Sie einen neuen Fall protokollieren möchten. Füllen Sie die Felder wie folgt aus:
 - a. Art des Support-Falls: `Channel Program`
 - b. Frage zu: `General Program Inquiry`
 - c. Spezifisch werden: Wählen Sie den am besten geeigneten Channel-Gehäusetyp aus.
 - d. Betreff: Fügen Sie eine kurze Beschreibung der Anfrage bei.
 - e. Beschreibung: Geben Sie eine detaillierte Beschreibung der Probleme, Fragen, Fehler und Schritte zur Fehlerbehebung an.
 - f. Anlagen: Fügen Sie gegebenenfalls Protokolle und Screenshots bei.

Testen in einer Sandbox für AWS API zur Umsatzmessung in Partner Central

Wie benutzt man die Sandbox

Gehen Sie wie folgt vor, um die Sandbox zu verwenden:

1. Erstellen Sie eine IAM-Rolle:

Erstellen Sie eine IAM-Rolle in dem mit Ihrem AWS Partner Central AWS-Konto verknüpften Konto.

2. Richtlinie zuweisen:

Weisen Sie der IAM-Rolle die folgende Richtlinie zu. Weitere Informationen finden Sie unter [Hinzufügen und Entfernen von IAM-Identitätsberechtigungen](#).

```
{
  "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. Verwenden Sie die Anmeldeinformationen für die IAM-Rolle:

Verwenden Sie die Anmeldeinformationen (geheimer Schlüssel und Zugriffsschlüssel) dieser IAM-Rolle in Ihrer Lösung, um die API-Aktionen auszuführen.

Testen AWS actions

Während der Testphase ist es oft notwendig, AWS Aktionen zu simulieren. Diese Simulation ermöglicht es Partnern, den gesamten Ablauf ihrer Integration mit AWS Diensten gründlich zu testen. Jede Anfrage enthält einen Katalogparameter, der die Datenumgebung bestimmt.

Simuliert die Erstellung einer Revenue Attribution-ID

Um die Erstellung einer Umsatzzuordnungs-ID zu simulieren, nehmen Sie diese in die Nutzlast der CreateRevenueAttribution Aktion auf. "Catalog": "Sandbox"

Beispiel:

```
{
  "Catalog": "Sandbox",
  "ClientToken": "String",
  "Name": "String",
  "Description": "String",
  "TenancyModel": "MULTI_TENANT | SINGLE_TENANT",
  "ProductIdentifier": "String",
  "Tags": [
    {
      "Key": "String",
      "Value": "String"
    }
  ]
}
```

```
}
```

Zusätzliche Hinweise zum Testen

1. Um andere Aktionen zu testen, legen Sie sie "Catalog": "Sandbox" in der Payload fest.
2. Um zur Produktion überzugehen, ändern Sie den Katalogwert von Sandbox auf AWS.

Codebeispiele für die Verwendung von Partner Central AWS SDKs

Die folgenden Codebeispiele zeigen, wie Partner Central mit einem AWS Software Development Kit (SDK) verwendet wird.

Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Während Aktionen Ihnen zeigen, wie Sie einzelne Service-Funktionen aufrufen, können Sie Aktionen im Kontext der zugehörigen Szenarien anzeigen.

Szenarien sind Codebeispiele, die Ihnen zeigen, wie Sie bestimmte Aufgaben ausführen, indem Sie mehrere Funktionen innerhalb eines Service aufrufen oder mit anderen AWS-Services kombinieren.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Codebeispiele

- [Grundlegende Beispiele für die Verwendung von Partner Central AWS SDKs](#)
 - [Aktionen für Partner Central unter Verwendung AWS SDKs](#)
 - [Verwenden Sie AssignOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie AssociateOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie CreateOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie DisassociateOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie GetAwsOpportunitySummary mit einem AWS SDK](#)
 - [Verwenden Sie GetEngagementInvitation mit einem AWS SDK](#)
 - [Verwenden Sie GetOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie ListEngagementInvitations mit einem AWS SDK](#)
 - [Verwenden Sie ListOpportunities mit einem AWS SDK](#)
 - [Verwenden Sie ListSolutions mit einem AWS SDK](#)
 - [Verwenden Sie RejectEngagementInvitation mit einem AWS SDK](#)
 - [Verwenden Sie StartEngagementByAcceptingInvitationTask mit einem AWS SDK](#)
 - [Verwenden Sie StartEngagementFromOpportunityTask mit einem AWS SDK](#)

- [Verwenden Sie UpdateOpportunity mit einem AWS SDK](#)
- [Szenarien für die Verwendung von Partner Central AWS SDKs](#)
 - [Aktualisieren der zugehörigen Entität einer Opportunity](#)

Grundlegende Beispiele für die Verwendung von Partner Central AWS SDKs

Die folgenden Codebeispiele zeigen, wie Sie die Grundlagen von AWS Partner Central mit AWS SDKs verwenden können.

Beispiele

- [Aktionen für Partner Central unter Verwendung AWS SDKs](#)
 - [Verwenden Sie AssignOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie AssociateOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie CreateOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie DisassociateOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie GetAwsOpportunitySummary mit einem AWS SDK](#)
 - [Verwenden Sie GetEngagementInvitation mit einem AWS SDK](#)
 - [Verwenden Sie GetOpportunity mit einem AWS SDK](#)
 - [Verwenden Sie ListEngagementInvitations mit einem AWS SDK](#)
 - [Verwenden Sie ListOpportunities mit einem AWS SDK](#)
 - [Verwenden Sie ListSolutions mit einem AWS SDK](#)
 - [Verwenden Sie RejectEngagementInvitation mit einem AWS SDK](#)
 - [Verwenden Sie StartEngagementByAcceptingInvitationTask mit einem AWS SDK](#)
 - [Verwenden Sie StartEngagementFromOpportunityTask mit einem AWS SDK](#)
 - [Verwenden Sie UpdateOpportunity mit einem AWS SDK](#)

Aktionen für Partner Central unter Verwendung AWS SDKs

Die folgenden Codebeispiele zeigen, wie einzelne Partner Central-Aktionen mit AWS SDKs durchgeführt werden. Jedes Beispiel enthält einen Link zu GitHub, wo Sie Anweisungen zum Einrichten und Ausführen des Codes finden.

Diese Auszüge rufen die Partner-Central-API auf und sind Codeauszüge aus größeren Programmen, die im Kontext ausgeführt werden müssen. Sie können Aktionen im Kontext unter [Szenarien für die Verwendung von Partner Central AWS SDKs](#) anzeigen.

Die folgenden Beispiele enthalten nur die am häufigsten verwendeten Aktionen. Eine vollständige Liste finden Sie in der [API-Referenz für AWS-Partner-Central](#).

Beispiele

- [Verwenden Sie AssignOpportunitymit einem AWS SDK](#)
- [Verwenden Sie AssociateOpportunitymit einem AWS SDK](#)
- [Verwenden Sie CreateOpportunitymit einem AWS SDK](#)
- [Verwenden Sie DisassociateOpportunitymit einem AWS SDK](#)
- [Verwenden Sie GetAwsOpportunitySummarymit einem AWS SDK](#)
- [Verwenden Sie GetEngagementInvitationmit einem AWS SDK](#)
- [Verwenden Sie GetOpportunitymit einem AWS SDK](#)
- [Verwenden Sie ListEngagementInvitationsmit einem AWS SDK](#)
- [Verwenden Sie ListOpportunitiesmit einem AWS SDK](#)
- [Verwenden Sie ListSolutionsmit einem AWS SDK](#)
- [Verwenden Sie RejectEngagementInvitationmit einem AWS SDK](#)
- [Verwenden Sie StartEngagementByAcceptingInvitationTaskmit einem AWS SDK](#)
- [Verwenden Sie StartEngagementFromOpportunityTaskmit einem AWS SDK](#)
- [Verwenden Sie UpdateOpportunitymit einem AWS SDK](#)

Verwenden Sie **AssignOpportunity**mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie AssignOpportunity verwendet wird.

Java

SDK für Java 2.x

Weisen Sie eine bestehende Opportunity einem anderen Benutzer neu zu.

```
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;
    }
}

```

- Einzelheiten zur API finden Sie [AssignOpportunity](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Weisen Sie eine bestehende Opportunity einem anderen Benutzer neu zu.

```

#!/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()
```

- Einzelheiten zur API finden Sie [AssignOpportunity](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **AssociateOpportunity** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie AssociateOpportunity verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Aktualisieren der zugehörigen Entität einer Opportunity](#)

Java

SDK für Java 2.x

Erstellen Sie eine formelle Zuordnung zwischen einer Opportunity und verschiedenen verwandten Entitäten.

```
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 entityIdIdentifier = "S-00000000";

        AssociateOpportunityResponse response = getResponse(opportunityId,
            entityType, entityIdIdentifier );

        ReferenceCodesUtils.formatOutput(response);
    }

    static AssociateOpportunityResponse getResponse(String opportunityId, String
        entityType, String entityIdIdentifier) {

        AssociateOpportunityRequest associateOpportunityRequest =
            AssociateOpportunityRequest.builder()
                .catalog(Constants.CATALOG_T0_USE)

```

```

        .opportunityIdentifier(opportunityId)
        .relatedEntityType(entityType)
        .relatedEntityIdentifier(entityIdentifier)
        .build();

    AssociateOpportunityResponse response =
client.associateOpportunity(associateOpportunityRequest);

    return response;
}
}

```

- Einzelheiten zur API finden Sie [AssociateOpportunity](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Erstellen Sie eine formelle Zuordnung zwischen einer Opportunity und verschiedenen verwandten Entitäten.

```

#!/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_idenfifier, opportunityIdentifier):
    associate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : entity_type,
        "RelatedEntityIdentifier" : entity_idenfifier
    }
    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_idenfifier = "S-0059717"
    opportunityIdentifier = "05465588"

    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_idenfifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()

```

- Einzelheiten zur API finden Sie [AssociateOpportunity](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. [Verwenden AWS Partner Central-API mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **CreateOpportunity** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie CreateOpportunity verwendet wird.

.NET

SDK für .NET

Erstellen Sie eine Opportunity.

```
// 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.");
}

```

```
    }  
  }  
}
```

- Einzelheiten zur API finden Sie [CreateOpportunity](#) in der AWS SDK für .NET API-Referenz.

Java

SDK für Java 2.x

Erstellen Sie eine Opportunity.

```
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> expectedCustomerSpend = 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();
            expectedCustomerSpend.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(expectedCustomerSpend)
            .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(opportunityTeamContact)
        .softwareRevenue(softwareRevenue)
        .build();

    CreateOpportunityResponse response =
    client.createOpportunity(createOpportunityRequest);
    System.out.println("Successfully created: " + response);

    return response;
}
}

```

- Einzelheiten zur API finden Sie [CreateOpportunity](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Erstellen Sie eine Opportunity.

```
#!/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()
```

- Einzelheiten zur API finden Sie [CreateOpportunity](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. [Verwenden AWS Partner Central-API mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **DisassociateOpportunity** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie DisassociateOpportunity verwendet wird.

Beispiele für Aktionen sind Codeauszüge aus größeren Programmen und müssen im Kontext ausgeführt werden. Im folgenden Codebeispiel können Sie diese Aktion im Kontext sehen:

- [Aktualisieren der zugehörigen Entität einer Opportunity](#)

Java

SDK für Java 2.x

Entfernen Sie eine bestehende Verknüpfung zwischen einer Opportunity und verwandten Entitäten.

```
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.DisassociateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityResponse;

/*
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 entityIdIdentifier = "S-00000000";

        DisassociateOpportunityResponse response = getResponse(opportunityId,
            entityType, entityIdIdentifier );

        ReferenceCodesUtils.formatOutput(response);
    }

    static DisassociateOpportunityResponse getResponse(String opportunityId, String
        entityType, String entityIdIdentifier) {

        DisassociateOpportunityRequest disassociateOpportunityRequest =
            DisassociateOpportunityRequest.builder()

```

```
        .catalog(Constants.CATALOG_TO_USE)
        .opportunityIdentifier(opportunityId)
        .relatedEntityType(entityType)
        .relatedEntityIdentifier(entityIdentifier)
        .build();

        DisassociateOpportunityResponse response =
client.disassociateOpportunity(disassociateOpportunityRequest);

        return response;
    }
}
```

- Einzelheiten zur API finden Sie [DisassociateOpportunity](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Entfernen Sie eine bestehende Verknüpfung zwischen einer Opportunity und verwandten Entitäten.

```
#!/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 = "04397574"

    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()
```

- Einzelheiten zur API finden Sie [DisassociateOpportunity](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetAwsOpportunitySummary** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie GetAwsOpportunitySummary verwendet wird.

Java

SDK für Java 2.x

Ruft eine Zusammenfassung einer AWS Opportunity ab.

```
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.GetAwsOpportunitySummaryRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetAwsOpportunitySummaryResponse;

/*
 * 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;
}
}
```

- Einzelheiten zur API finden Sie [GetAwsOpportunitySummary](#) unter AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Ruft eine Zusammenfassung einer AWS Opportunity ab.

```
#!/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 = "05465588"

    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()
```

- Einzelheiten zur API finden Sie [GetAwsOpportunitySummary](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetEngagementInvitation** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie `GetEngagementInvitation` verwendet wird.

Java

SDK für Java 2.x

Ruft die Details einer Einladung zur Interaktion ab, die von AWS einem Partner geteilt wurde.

```
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.GetEngagementInvitationRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.GetEngagementInvitationResponse;

/*
```

```
* 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;
    }
}
```

- Einzelheiten zur API finden Sie [GetEngagementInvitation](#) unter AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Ruft die Details einer Einladung zur Interaktion ab, die von AWS einem Partner geteilt wurde.

```
#!/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_request)
        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-00000000IS0Qga"

    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()

```

- Einzelheiten zur API finden Sie [GetEngagementInvitation](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **GetOpportunity** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie GetOpportunity verwendet wird.

.NET

SDK für .NET

Rufen Sie eine Opportunity ab.

```

// 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 = "01111111";
        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.");
        }
    }
}
```

- Einzelheiten zur API finden Sie [GetOpportunity](#) in der AWS SDK für .NET API-Referenz.

Go

SDK für Go V2

Rufen Sie eine Opportunity ab.

```
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/partnercentral-selling"
)

func main() {
    config, err := config.LoadDefaultConfig(context.TODO())

    if err != nil {
        log.Fatal(err)
    }

    config.Region = "us-east-1"

    client := partnercentral-selling.NewFromConfig(config)
```

```

output, err := client.GetOpportunity(context.TODO(),
&partnercentralselling.GetOpportunityInput{
    Identifier: aws.String("01111111"),
    Catalog:    aws.String("AWS"),
})

if err != nil {
    log.Fatal(err)
}
log.Println("printing opportuniy...\n")

jsonOutput, err := json.MarshalIndent(output, "", "    ")

fmt.Println(string(jsonOutput))
}

```

- Einzelheiten zur API finden Sie [GetOpportunity](#) in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

Rufen Sie eine Opportunity ab.

```

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;
    }
}
```

- Einzelheiten zur API finden Sie [GetOpportunity](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Rufen Sie eine Opportunity ab.

```
#!/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 = "05465588"
```

```
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()
```

- Einzelheiten zur API finden Sie [GetOpportunity](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListEngagementInvitations** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie ListEngagementInvitations verwendet wird.

Java

SDK für Java 2.x

Ruft eine Liste mit Einladungen zur Interaktion ab, die an den Partner gesendet wurden.

```
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;
    }
}

```

- Einzelheiten zur API finden Sie [ListEngagementInvitations](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Ruft eine Liste mit Einladungen zur Interaktion ab, die an den Partner gesendet wurden.

```

#!/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()

```

- Einzelheiten zur API finden Sie [ListEngagementInvitations](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListOpportunities** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie ListOpportunities verwendet wird.

.NET

SDK für .NET

Listen Sie Opportunitys auf.

```

// 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.");
}
}
}
}
}

```

- Einzelheiten zur API finden Sie [ListOpportunities](#) in der AWS SDK für .NET API-Referenz.

Go

SDK für Go V2

Listen Sie Opportunitys auf.

```

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/partnercentral-selling"
)

```

```
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))
}
```

- Einzelheiten zur API finden Sie [ListOpportunities](#) in der AWS SDK für Go API-Referenz.

Java

SDK für Java 2.x

Listen Sie Opportunitys auf.

```
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 ListOpportunitiesPaging {

    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;
    }
}

```

- Einzelheiten zur API finden Sie [ListOpportunities](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Listen Sie Opportunitys auf.

```

#!/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()
```

- Einzelheiten zur API finden Sie [ListOpportunities](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **ListSolutions** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie ListSolutions verwendet wird.

Java

SDK für Java 2.x

Ruft eine Liste von Partner Solutions ab, die der Partner bei Partner Central registriert hat.

```
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_T0_USE)
    .maxResults(5)
    .build();

ListSolutionsResponse response = client.listSolutions(listSolutionsRequest);

solutionSummaries.addAll(response.solutionSummaries());

    return solutionSummaries;
}
}

```

- Einzelheiten zur API finden Sie [ListSolutions](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Ruft eine Liste von Partner Solutions ab, die der Partner bei Partner Central registriert hat.

```

#!/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()
```

- Einzelheiten zur API finden Sie [ListSolutions](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **RejectEngagementInvitation** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie `RejectEngagementInvitation` verwendet wird.

Java

SDK für Java 2.x

Lehnt jeden ab `EngagementInvitation`, der AWS geteilt wurde.

```
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(A0) 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;
    }
}

```

- Einzelheiten zur API finden Sie [RejectEngagementInvitation](#) unter AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Lehnt jeden ab EngagementInvitation , der AWS geteilt wurde.

```
#!/usr/bin/env python
```

```
"""
```

Purpose

PC-API-05 AWS Originated A0 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()
```

- Einzelheiten zur API finden Sie [RejectEngagementInvitation](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. [Verwenden AWS Partner Central-API mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **StartEngagementByAcceptingInvitationTask** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie **StartEngagementByAcceptingInvitationTask** verwendet wird.

Java

SDK für Java 2.x

Beginnt das Engagement mit der Annahme eines 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(A0)
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;  
    }  
}
```

- Einzelheiten zur API finden Sie [StartEngagementByAcceptingInvitationTask](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Beginnt das Engagement mit der Annahme eines 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_opportunity_engagement_invitation_request)
            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()

```

- Einzelheiten zur API finden Sie [StartEngagementByAcceptingInvitationTask](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. [Verwenden AWS Partner Central-API mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **StartEngagementFromOpportunityTask** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie **StartEngagementFromOpportunityTask** verwendet wird.

Java

SDK für Java 2.x

Initiiert den Interaktionsprozess von einer bestehenden Opportunity aus, indem die Einladung zur Interaktion angenommen und eine entsprechende Opportunity im System des Partners erstellt wird.

```
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.StartEngagementFromOpportunityTask;
import
    software.amazon.awssdk.services.partnercentralselling.model.StartEngagementFromOpportunityTaskRequest;
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;
    }
}

```

- Einzelheiten zur API finden Sie [StartEngagementFromOpportunityTask](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Initiiert den Interaktionsprozess von einer bestehenden Opportunity aus, indem die Einladung zur Interaktion angenommen und eine entsprechende Opportunity im System des Partners erstellt wird.

```
#!/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(identifizier):

    start_engagement_from_opportunity_task_request = {
        "AwsSubmission": {
            "InvolvementType": "Co-Sell",
            "Visibility": "Full"
        },
        "Catalog": CATALOG_TO_USE,
        "Identifizier" : identifizier,
        "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 = "05465588"

    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()

```

- Einzelheiten zur API finden Sie [StartEngagementFromOpportunityTask](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. [Verwenden AWS Partner Central-API mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Verwenden Sie **UpdateOpportunity** mit einem AWS SDK

Die folgenden Code-Beispiele zeigen, wie UpdateOpportunity verwendet wird.

Java

SDK für Java 2.x

Aktualisieren Sie eine Opportunity.

```
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.competitorName
            .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;
    }
}
}
}

```

- Einzelheiten zur API finden Sie [UpdateOpportunity](#) in der AWS SDK for Java 2.x API-Referenz.

Python

SDK für Python (Boto3)

Aktualisieren Sie eine Opportunity.

```

#!/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 = "05465588"
```

```
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()
```

- Einzelheiten zur API finden Sie [UpdateOpportunity](#) in AWS SDK for Python (Boto3) API Reference.

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter. [Verwenden AWS Partner Central-API mit einem AWS SDK](#) Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Szenarien für die Verwendung von Partner Central AWS SDKs

Die folgenden Codebeispiele zeigen Ihnen, wie Sie gängige Szenarien in Partner Central mit AWS SDKs implementieren. Diese Szenarien zeigen Ihnen, wie Sie bestimmte Aufgaben durch den Aufruf mehrerer Funktionen innerhalb von Partner Central oder in Kombination mit anderen AWS-Services ausführen können. Jedes Szenario enthält einen Link zum vollständigen Quell-Code, wo Sie Anleitungen zum Einrichten und Ausführen des Codes finden.

Szenarien zielen auf eine mittlere Erfahrungsebene ab, um Ihnen zu helfen, Service-Aktionen im Kontext zu verstehen.

Beispiele

- [Aktualisieren der zugehörigen Entität einer Opportunity](#)

Aktualisieren der zugehörigen Entität einer Opportunity

Die folgenden Code-Beispiele veranschaulichen Folgendes:

- Trennen Sie die Zuordnung einer alten Entität.
- Ordnen Sie eine neue Entität zu.

Java

SDK für Java 2.x

Note

Es gibt noch mehr dazu GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie es im [Scenarios](#)-Repository eingerichtet und ausgeführt wird.

Aktualisieren der zugehörigen Entität einer Opportunity

```
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.DisassociateOpportunityRequest;
import
    software.amazon.awssdk.services.partnercentralselling.model.DisassociateOpportunityResponse;

/*
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-00000000";
    String newEntityIdentifier = "S-00111111";

    disassociateOppornitityResponse(opportunityId, entityType,
originalEntityIdentifier );
    AssociateOppportunityResponse associateOppportunityResponse =
associateOppportunityResponse(opportunityId, entityType, newEntityIdentifier );

    ReferenceCodesUtils.formatOutput(associateOppportunityResponse);
}

private static AssociateOppportunityResponse associateOppportunityResponse(String
opportunityId, String entityType, String entityIdentifier) {

    AssociateOppportunityRequest associateOppportunityRequest =
AssociateOppportunityRequest.builder()
        .catalog(Constants.CATALOG_TO_USE)
        .opportunityIdentifier(opportunityId)
        .relatedEntityType(entityType)
        .relatedEntityIdentifier(entityIdentifier)
        .build();

    AssociateOppportunityResponse response =
client.associateOppportunity(associateOppportunityRequest);

    return response;
}

private static DisassociateOppportunityResponse
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;
    }
}

```

- Weitere API-Informationen finden Sie in den folgenden Themen der AWS SDK for Java 2.x-API-Referenz.
 - [AssociateOpportunity](#)
 - [DisassociateOpportunity](#)

Python

SDK für Python (Boto3)

Note

Es gibt noch mehr GitHub. Hier finden Sie das vollständige Beispiel und erfahren, wie Sie das [AWS-Code-Beispiel](#) einrichten und ausführen.

Aktualisieren der zugehörigen Entität einer Opportunity

```

#!/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_identifizier = "S-0050014"
opportunityIdentifizier = "04397574"

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_identifizier,
new_entity_identifizier, opportunityIdentifizier))

if __name__ == "__main__":
    usage_demo()
```

- Weitere API-Informationen finden Sie in den folgenden Themen der API-Referenz zum AWS SDK für Python (Boto3).
 - [AssociateOpportunity](#)
 - [DisassociateOpportunity](#)

Eine vollständige Liste der AWS SDK-Entwicklerhandbücher und Codebeispiele finden Sie unter [Verwenden AWS Partner Central-API mit einem AWS SDK](#). Dieses Thema enthält auch Informationen zu den ersten Schritten und Details zu früheren SDK-Versionen.

Versionshinweise

Diese Seite enthält eine Zusammenfassung der wesentlichen Änderungen an der Dokumentation für die AWS Partner Central-API, beginnend mit dem neuesten Update.

Versionsverlauf

Änderungen	Beschreibung	Date
RC 6.0	AWS Partner Central hat neue APIs zur Messung von Partnereinnahmen hinzugefügt.	2026-06-30
RC 5,3	ACE Resonate-Updates: GetAwsOpportunitySummary Antwort aktualisiert.	2026-06-16
RC 5.2	- Unterstützung für EUR-Währungen: CreateOpportunity, UpdateOpportunity, und akzeptiert CreateEngagementInvitation jetzt EUR zusätzlich USD zum MRR-Währungscode. Wenn die AWS-Partition aws-eusc (AWS European Sovereign Cloud) lautet, muss der Währungscode lauten EUR. - EUR bei Lesevorgängen: GetOpportunity, ListOpportunities, und GetAwsOpportunitySummary jetzt zurück EUR	2026-03-30

Änderungen	Beschreibung	Date
	in das CurrencyCode Feld für Opportunities in der aws-eusc Partition.	
RC 5.1	• Unterstützung für EUR-Währungen: CreateOpportunity UpdateOpportunity , und akzeptiert CreateEngagementInvitation jetzt EUR zusätzlich USD zum MRR-Währungscode. Wenn die AWS-Partition aws-eusc (AWS European Sovereign Cloud) lautet, muss der Währungscode lauten EUR. • EUR bei Lesevorgängen: GetOpportunity , ListOpportunities , und GetAwsOpportunitySummary jetzt zurück EUR in das CurrencyCode Feld für Opportunities in der aws-eusc Partition.	2026-03-30

Änderungen	Beschreibung	Date
RC 5	• Konto-API eingeführt: AWS Partner Central wurde um eine neue Konto-API erweitert, mit der Partner ihre Kontoinformationen, Registrierungen, Profile, Domainverwaltung und Partnerverbindungen verwalten können. • Referenzdokumentation zur Konto-API: Es wurde eine umfassende API-Referenzdokumentation für die Konto-API hinzugefügt, einschließlich Partnerregistrierung, Profilverwaltung, Domainverifizierung und Workflows zur Kontoverbindung. • Konto-API-Protokollierung: Protokollierungsunterstützung und Dokumentation für die Konto-API wurden hinzugefügt, um die API-Nutzung zu überprüfen und nachzuverfolgen. • Konto-API-Benachrichtigungen: Unterstützung für Ereignisbenachrichtigungen für die Konto-API hinzugefügt, sodass Partner in Echtzeit Updates zu kontobezogenen Aktivitäten, einschließlich Verbindungen	2025-11-30

Änderungen	Beschreibung	Date
	zu Partnerkonten, erhalten können. • Konto-API-Kontingente: Die Dokumentation zu Servicekontingenten für die Account-API wurde hinzugefügt. • Sandbox-Tests für die Account-API: Es wurden Sandbox-Testfunktionen für die Account-API hinzugefügt, sodass Partner Account-Management-Workflows in einer Produktionsumgebung testen können. • Einführung der Benefits API: AWS Partner Central wurde um eine neue Benefits API erweitert, mit der Partner über eine zentrale Schnittstelle Leistungen aus allen AWS Partnerprogrammen ermitteln, beantragen und verwalten können. • Referenzdokumentation zur Benefits API: Es wurde eine umfassende API-Referenzdokumentation für die Benefits API hinzugefügt, die sich mit der Beantragung von Leistungen, der Zuweisung und der	

Änderungen	Beschreibung	Date
	Erfüllung von Leistungen befasst. • Protokollierung der Benefits API: Protokollierungsunterstützung und Dokumentation für die Benefits API wurden hinzugefügt, um die API-Nutzung zu überprüfen und nachzuverfolgen. • API-Benachrichtigungen für Zusatzleistungen: Für die Benefits API wurde die Unterstützung für Benachrichtigungen über Ereignisse hinzugefügt, sodass Partner in Echtzeit über Anträge und Zuweisungen von Leistungen informiert werden können. • API-Kontingente für Zusatzleistungen: Die Dokumentation zu Servicekontingenten für die Benefits API wurde hinzugefügt. • Sandbox-Tests für die Benefits API: Es wurden Sandbox-Testfunktionen für die Benefits API hinzugefügt, sodass Partner Workflows für das Leistungsmanagement in einer Umgebung außerhalb der Produktion testen können.	

Änderungen	Beschreibung	Date
	• Aktualisierte Verkaufs-API: Verbesserte Aktionen zur Interaktion mit der Verkaufs-API zur Unterstützung des Lead-Management-Workflows. • Aktualisierungen der Verkaufs-API-Dokumentation: Verbesserte Dokumentation zur Verkaufs-API mit verbesserter Struktur und aktualisierten Workflows für das Lead-Management. • Updates für E-Mail-Benachrichtigungen für den Verkauf: Verbesserte Dokumentation zu Event-Benachrichtigungen für Engagement- und Engagement-Einladungen. • Erweiterung der API-Struktur: Erweiterte Dokumentationsstruktur zur Unterstützung von vier verschiedenen APIs (Account, Benefits, Selling und Channel) mit speziellen Abschnitten für die Berechtigungen, Protokollierung, Benachrichtigungen, Kontingente und Sandbox-Tests jeder API.	

Änderungen	Beschreibung	Date
	• Verbesserte Dokumentation zur API-Nutzung: Es wurden umfassende Nutzungshandbücher für Account- und Benefits-APIs mit detaillierten Workflows und Integrationsmustern hinzugefügt. • Dokumentation zu unterstützten Regionen: Dokumentation für unterstützte AWS Regionen für Account- und Benefits-APIs hinzugefügt. • Neuer Filter für <code>ListOpportunities</code> : Es wurde ein <code>TargetCloseDate</code> Filter hinzugefügt, mit dem Opportunities anhand des voraussichtlichen Abschlussdatums <code>AfterTargetCloseDate</code> und der <code>BeforeTargetCloseDate</code> Felder im YYYY-MM-DD Format gefiltert werden können.	

Änderungen	Beschreibung	Date
RC 4	• Channel-API eingeführt: AWS Partner Central wurde um eine neue Channel-API erweitert, mit der Partner ihre Kanalbeziehungen und die Zusammenarbeit zwischen Partnern verwalten können. • Referenzdokumentation zur Channel-API: Es wurde eine umfassende API-Referenzdokumentation für die Channel-API hinzugefügt, einschließlich aller verfügbaren Aktionen und Datenmodelle. • Kanal-API-Berechtigungen: Für die Channel-API wurden IAM-Berechtigungsrichtlinien und eine Dokumentation zur Zugriffskontrolle hinzugefügt. • Channel-API-Logging: Logging-Unterstützung und Dokumentation für die Channel-API wurden hinzugefügt, um die API-Nutzung zu prüfen und nachzuverfolgen. • Channel-API-Benachrichtigungen: Unterstützung für Event-Benachrichtigungen für die Channel-API hinzugefügt, sodass Partner	2025-11-19

Änderungen	Beschreibung	Date
	in Echtzeit über kanalbezogene Aktivitäten informiert werden können. • Kanal-API-Kontingente: Die Dokumentation zu Servicekontingenten für die Channel-API wurde hinzugefügt. • Sandbox-Tests für die Channel-API: Es wurden Sandbox-Testfunktionen für die Channel-API hinzugefügt, sodass Partner Channel-Workflows in einer Umgebung außerhalb der Produktion testen können. • Neuorganisation der API-Struktur: Die Dokumentation wurde neu organisiert, um zwischen der Verkaufs-API und der Channel-API zu unterscheiden, mit eigenen Abschnitten für die Berechtigungen, Protokollierung, Benachrichtigungen und Kontingente jeder API. • Dokumentation zu unterstützten Regionen: Dokumentation für unterstützte AWS Regionen sowohl für Verkaufs- als auch für Channel-APIs hinzugefügt.	

Änderungen	Beschreibung	Date
RC 3	• Aktualisierte API-Berechtigungsrichtlinien: Die Richtlinien wurden aktualisiert, um den in dieser Version eingeführten neuen Aktionen Rechnung zu tragen. Stellen Sie sicher, dass Ihre IAM-Rollen und -Berechtigungen entsprechend angepasst sind. • Details zur Verwendung des benutzerdefinierten Headers: Es wurden Details zur Verwendung des benutzerdefinierten Headers hinzugefügt <code>X-Amzn-User-Agent</code>, um die API-Nutzung zu prüfen und zu messen. • Aktionen durch asynchrone Aktionen ersetzt: Das Format wurde durch asynchrone Aktionen <code>AcceptOpportunityEngagementInvitation</code> ersetzt <code>StartEngagementByAcceptingInvitationTask</code> und das Payload-Format hat sich in Bezug auf RC2 geändert. • Umbenennung von Entitäten und Aktionen: Geändert zu. <code>OpportunityEngagem</code>	2024-10-17

Änderungen	Beschreibung	Date
	entInvitation EngagementInvitation Die Attribute für diese Aktionen wurden ebenfalls geändert, um sie an die neue Entität anzupassen. Die folgenden Aktionen wurden ersetzt: • GetOpportunityEngagementInvitation ist jetztGetEngagementInvitation . Mehrere Attribute in der Einladung wurden geändert, hinzugefügt oder entfernt. • ListOpportunityEngagementInvitations ist jetztListEngagementInvitations . • RejectOpportunityEngagementInvitation ist jetztRejectEngagementInvitation . • AssignOpportunity Parameter-Update: dauert AssignOpportunity jetzt Assignee stattAssigneeEmail . Aktualisieren Sie Ihre	

Änderungen	Beschreibung	Date
	Implementierungen entsprechend. • <code>SubmitOpportunity</code> Änderung der Funktionalität: <code>SubmitOpportunity</code> wird jetzt durch die asynchrone Aktion ausgeführt. <code>StartEngagementFromOpportunityTask</code> <code>SubmitOpportunity</code> akzeptiert jetzt <code>InvolvementType</code> und <code>Visibility</code> • Umbenennung und Erweiterung von Attributen: <code>EstimatedAwsMonthlyRevenue</code> geändert in <code>ExpectedCustomerSpend</code>. <code>ExpectedCustomerSpend</code> enthält jetzt neue Attribute wie <code>Frequency</code> und <code>TargetCompany</code>. • Neue Aktion für AWS die Zusammenfassung von Geschäftschancen: Die Sichtbarkeit von Geschäftschancen wird nun durch eine separate Aktion mit dem Namen <code>getAwsOpportunitySummary</code>. • Workflow-Updates für die Arbeit mit Opportunities: Der	

Änderungen	Beschreibung	Date
	Workflow für die Arbeit mit Ihren Opportunities wurde aktualisiert und verwendet nun die neuen Aktionen, die in dieser Version eingeführt wurden. • Workflow-Updates für die Arbeit mit Opportunities: Der Workflow für die Arbeit mit Opportunities wurde aktualisiert und ist nun EngagementInvitations benutzerfreundlich, was den Umgang mit den von AWS bereitgestellten Möglichkeiten verbessert. • Workflow-Updates für die Nachverfolgung von Updates: Der Workflow zur Nachverfolgung von Updates funktioniert jetzt mit Opportunity-Updates, was die Übersicht und Effizienz bei der Überwachung des Opportunity-Status verbessert. • Neue Filter für Auflistungsaktionen: Für Aktionen ListSolutions und ListOpportunities Aktionen sind neue Filter verfügbar. • Für ListSolutions sind die neuen	

Änderungen	Beschreibung	Date
	FilterFilterCategory, FilterIdentifier, und FilterStatus . - Für ListOpportunities sind die neuen FilterFilterCustomerComp anyName, FilterIdentifier, FilterLastModified Date, FilterLifecycleApp rovalStatus, FilterLifecycleStage, und FilterOrigin . - Entfernung des AccessControl Unterobjekts: Das AccessControl Unterobjekt im Opportunity-Modell wurde entfernt. NationalSecurity ist jetzt ein Attribut der obersten Ebene. - Attribute wurden verschoben nach GetAwsOpportunitySummary : Attribute wie AWSOpportunityTeam Insights (EngagementScore, NextBestAction), Origin und InvolvementType sind	

Änderungen	Beschreibung	Date
	jetzt über <code>GetAwsOpportunitySummary</code> statt über <code>GetOpportunity</code> verfügbar. • Einführung des <code>InvolvementType</code> Feldes: Das <code>InvolvementType</code> Feld wurde eingeführt, um zwischen Cosell- und ausschließlich sichtbaren Möglichkeiten zu unterscheiden. • Aktualisierungen der Veranstaltungsnamen: Die Ereignisse wurden aktualisiert, um den Änderungen Rechnung zu tragen. Beispiel: Die erstellte Einladung zur Interaktion mit Opportunity ist jetzt die Einladung zur Interaktion erstellt. • Aktualisierungen der Antwort-Payloads: Alle Antwort-Payloads enthalten jetzt eine <code>and/or ID-ARN</code>. • Payload-Updates anfordern : Alle Anforderungs-Payloads verwenden jetzt Identifikatoren als Eingabe, die entweder eine ID oder einen ARN akzeptieren. • Aktualisierung der Engagement-Entität:	

Änderungen	Beschreibung	Date
	Das Titelattribut in der Engagement-Entität wurde umbenannt. EngagementTitle - Entfernung des Filterpräfixes: Alle Filter enthalten das Filterpräfix nicht mehr, was den Filtermechanismus vereinfacht. - StartEngagementFromOpportunity Aktionsupdate: Das Attribut AwsInvolvement wurde geändert in AwsSubmission . - Aktualisierung der Projektdetails der Einladung: Invitation.ProjectDetails.TargetCompletionDate wurde umbenannt in Invitation.ProjectDetails.EstimatedCompletionDate . - Aktualisierung der Umsatzschätzung: Invitation.PartnerRevenueEstimate ist jetzt ExpectedCustomerSpend . Der Datentyp wurde von einem Objekt zu einem Array geändert, das ein Objekt enthält.	

Änderungen	Beschreibung	Date
	• Aktualisierung des Empfängerkontos: Das Feld <code>ReceiverAccount</code> wurde umbenannt <code>AccountReceiver</code> . • Neues Attribut für Einladungen zur Verlobung: Das <code>SenderContact</code> Attribut wurde für Einladungen zu Interaktionen eingeführt. • <code>Customer.Contact</code> ist jetzt ein Array statt ein Objekt. <code>OpportunityOwner</code> , <code>PartnerAccountManager</code> werden als Werte unterstützt <code>BusinessTitle</code> . • <code>PartnerOpportunityTeam</code> heißt jetzt <code>OpportunityTeam</code> . • <code>APNPrograms</code> ist jetzt <code>ApnPrograms</code> • Der Status der Einladungen zu Interaktionen wurde aktualisiert. • <code>ListEngagementInvitations</code> benötigt jetzt <code>ParticipantType</code> als <code>RECEIVER</code>.	

Änderungen	Beschreibung	Date
RC 2	• Hat eine neue Entität namens eingeführt OpportunityEngagementInvitation. • Eine neue Aktion eingeführt: ListOpportunityEngagementInvitations. • Eine neue Aktion eingeführt: GetOpportunityEngagementInvitation. • Es wurde eine neue Veranstaltung eingeführt: „Opportunity Engagement Invitation Created“. • Ersetzt AcceptOpportunity durch AcceptOpportunityEngagementInvitation. • Ersetzt RejectOpportunity durch RejectOpportunityEngagementInvitation. • „Opportunity Accepted“ wurde durch „Opportunity Engagement Invitation Accepted“ ersetzt. • „Abgelehnte Geschäftschance“ wurde durch „Einladung zur Verkaufschance abgelehnt“ ersetzt. • PrimaryNeedsFromAWS geändert zu PrimaryNeedsFromAws.	2024-08-07

Änderungen	Beschreibung	Date
	• AWSOpportunityTeam geändert zuAwsOpportunityTeam . • AWSSalesLifeCycle geändert zuAwsSalesLifeCycle . • PrimaryNeedsFromAWSChangeReason geändert zuPrimaryNeedsFromAwsChangeReason . • AWSAccountId geändert zuAwsAccountId . • ExpectedMonthlyAWSRevenue geändert zuExpectedMonthlyAwsRevenue . • AWSFundingUsed geändert zuAwsFundingUsed . • AWSMarketplaceOffers geändert zuAwsMarketplaceOffers . • Countrygeändert zuCountryCode . • PartnerOpportunityContact geändert zuPartnerOpportunityTeam . • Feld aktualisiert Contact.Title aufContact.BusinessTitle .	

Änderungen	Beschreibung	Date
	• Feld wurde aktualisiert ContactInfo aufOwnerInfo . • AWS Team wurde von einer Karte zu einer Liste geändert. • Es wurden Details zur Liste der akzeptierten hinzugefügt RejectionReasons. • Es wurden Informationen zur Verwendung des Client-Tokens bereitgestellt. • Enthält Einzelheiten zur Verwendung UpdateOpportunity. • Es wurden Hinweise zur Verwendung von AWS Produkten und Partnerlösungen hinzugefügt. • Es wurden Einzelheiten zur OpportunityEngagementInvitation Entität bereitgestellt. • Es wurde aktualisiert StateOrRegion und enthält nun eine überarbeitete Liste der akzeptierten Werte. • Es wurden mehrere Bugfixes implementiert, um die Leistung und die Fehlermeldung zu verbessern.	

Änderungen	Beschreibung	Date
RC 1	• [Aktuell] Umbenennung von ApprovalStatus in ReviewStatus: Wir haben das ApprovalStatus Feld in umbenannt, ReviewStatus um seinen Zweck besser widerzuspiegeln. Darüber hinaus führen wir den neuen Status „Ausstehende Einreichung“ ein, um darauf hinzuweisen, dass es sich um einen Entwurf einer Verkaufschance handelt. ReviewStatus Sie akzeptiert nun die folgenden Werte: Ausstehende Einreichung, Eingereicht, Wird geprüft, Genehmigt, Abgelehnt und Maßnahme erforderlich. Der bisherige Status „Entwurf“ wird durch „Ausstehende Einreichung“ ersetzt. • [Aktuell] Einführung von SubmitOpportunity Action: Eine neue Aktion SubmitOpportunity,, wurde eingeführt. Im Gegensatz zum aktuellen Verhalten , bei dem die Opportunity CreateOpportunity auch weitergeleitet wird, können Sie jetzt eine Opportunity mit dem Status „Ausstehende Einreichung“ erstellen	2024-06-21

Änderungen	Beschreibung	Date
	, ohne sofort eine Lösung oder ein AWS Produkt zu verknüpfen. Um die Einreichung abzuschließen, verwenden Sie die AssociateOpportunity Aktion, um eine Lösung oder ein Produkt zu verlinken, gefolgt von SubmitOpportunity. Diese Trennung sorgt für eine bessere API-Antwortleistung und eine flexible Vorbereitungsphase vor der Einreichung. • [Breaking] Anforderung eines Katalogparameters und Verfall des Sandbox-Endpunkts: Der Endpunkt 'Gamma' (partnercentral-selling-gamma.us-east-1.amazonaws.com) wird nach 7/30 nicht mehr verfügbar sein. Für alle API-Aktionen ist jetzt ein Katalogparameter erforderlich, um anzugeben, ob der Vorgang in der Sandbox oder im Katalog ausgeführt wird. AWS Benachrichtigungsregeln filtern jetzt nach Katalog statt nach EnvironmentName.	

Änderungen	Beschreibung	Date
	• [Breaking] Umbenennung von OpportunityIdentifier in Identifier: Actions AssignOpportunity, und RejectOpportunity jetzt Identifier verwenden AcceptOpportunity, anstatt mit dem Identifier konsistent OpportunityIdentifier zu bleiben. UpdateOpportunity • Änderungen vom Typ Enum zu String für APNProgram und UseCase: Wir werden CustomerUseCase,, und Project.CustomerUseCase vom Aufzählungstyp in den String-Typ konvertieren Project.APNPrograms Marketing.ActivityUseCases, um die Risiken zu verringern, die mit der Änderung von Werten verbunden sind. Diese Felder akzeptieren nur Werte aus vordefinierten Listen, die jedoch nicht nativ in den SDKs verfügbar sind. • Verbesserte Fehlerbehandlung: Die Fehlerbehandlung innerhalb der APIs wurde erheblich verbessert, um das Debuggen zu vereinfachen und Probleme schneller zu lösen. Die neue	

Änderungen	Beschreibung	Date
	Fehlerbehandlung strukturiert Fehler in zwei Stufen: 1. Überprüfung der Anfrage fehlgeschlagen: Es werden Datentypen und Format geprüft, oder 2. Geschäftsvalidierung fehlgeschlagen: Alle Probleme in einer Payload werden in einer Antwort aufgelistet, wobei das fehlerhafte Feld angegeben wird. Dieses konsolidierte Feedback ermöglicht es Ihnen, Fehler effizienter zu beheben und zu korrigieren. Fehlercodes und Formate wurden standardisiert. • Quell-ID des Kunden: Partner können den X-Amzn-User-Agent Header in ihre Anfragen aufnehmen , sodass die Quelle des Datenverkehrs leichter identifiziert werden kann. Verwenden Sie das Format [Lösungsname einschließlich Lösungsanbieter] [Version] Zum Beispiel:AWS Partner CRM Connector v2.0.1 • Bugfix, bereits bereitgestellt, keine Änderung am SDK] Opportunity Creation und	

Änderungen	Beschreibung	Date
	AWS Account-ID: Partner können jetzt eine Opportunity starten und gleichzeitig die AWS Konto-ID von Null auf einen gültigen Wert ändern, ohne dass Fehler auftreten. Bisher hinderte ein Fehler Partner daran, diese Aktionen separat durchzuführen. • Bugfix, bereits bereitgestellt, keine Änderung am SDK] Solution Association with New Opportunities: Partner können eine Lösung mit einer neu erstellten Opportunity verknüpfen. Bisher gingen die Lösungsinformationen verloren, wenn Partner eine neue Geschäftschance schufen und gleichzeitig eine Lösung zuordneten. Dieses Problem wurde behoben. • Bugfix, bereits bereitgestellt, keine Änderung am SDK] Anzeige der Opportunity-Eigentümergebenheiten: Das Problem, dass bei einigen Opportunities die Daten des Besitzers in einem kombinierten LastName "" - Feld statt in den erwarteten Feldern „E-Mail“, "FirstNam	

Änderungen	Beschreibung	Date
	e„und" LastName "angezeigt wurden, wurde behoben. • Bugfix, bereits bereitgestellt, keine Änderung am SDK] Customer.Account.WebsiteUrl Feld optional: Das Customer.Account.WebsiteUrl Feld wurde optional gemacht, wenn das AccessControls.NationalSecurity Feld auf „Ja“ steht, wodurch das API-Verhalten an das bestehende Verhalten der Benutzeroberfläche angepasst wird. Wenn das AccessControls.NationalSecurity Feld „Nein“ oder Null ist, ist das WebsiteUrl Feld zur Geschäftsvalidierung erforderlich. • Bugfix, bereits bereitgestellt, keine Änderung an den Anforderungen des SDK]AWS Konto-ID für Beratungspartner: Die Inkonsistenz in der Benutzeroberfläche, in der das Feld AWS Konto-ID für Beratungspartner als optional angezeigt wurde, wurde behoben. Es ist jetzt ein bedingtes Pflichtfeld,	

Änderungen	Beschreibung	Date
	wie in der Dokumentation beschrieben. • Bugfix, Bereitstellung am 6/19, keine Änderung am SDK] Phase of a Co-sell Opportunity auf Closed Lost: Partner können die Phase einer Co-sell Opportunity jetzt erfolgreich auf Closed Lost aktualisieren, sofern ein Closed Lost Reason angegeben wird. Dadurch wurde der vorherige Fehler behoben, dass beim Schließen der Opportunity ein Pflichtfeld fehlt: Der Grund für die Schließung der Verkaufschance ist erforderlich. • [Dokumentation] Verbesserungen an der Dokumentation: Es wurden mehrere Verbesserungen an der Dokumentation vorgenommen, um den Änderungen Rechnung zu tragen. • Bekannte Probleme mit einer noch ausstehenden Lösung und einer Liste bevorstehender Funktionen: • Bei der Ausführung von AssociateOpportunity Aktionen mit Lösungen oder Produkten werden	

Änderungen	Beschreibung	Date
	die Fehlermeldungen von Angeboten falsch angezeigt. • Fähigkeit, Chancen aus der Ferne zu schaffen und den AWS Validierungsprozess in Sandbox zu simulieren • Postal Code gibt Null zurück, wenn es das Format XXXXX-XXXX für USA hat. • Die Optionen „Nächste Schritte“ und „Verlauf der nächsten Schritte“ sind nicht für Angebote verfügbar, die als „Ich benötige keine Unterstützung von“ gekennzeichnet sind AWS. • Bisherige Verkaufschancen mit Beschreibungen von Kundenproblemen mit mehr als 255 Zeichen konnten nicht abgerufen werden. • Möglichkeit, Opportunities zu löschen, deren Einreichung noch aussteht. • LeadSource ist im Datenmodell nicht verfügbar.	

Änderungen	Beschreibung	Date
Betaversion 5	• Geändert LifeCycle .ApprovalStatus zu LifeCycle.ReviewStatus • Geändert Insights. ReviewComments zu LifeCycle.ReviewComments • Hinzugefügt LifeCycle .ReviewStatusReason (neues Attribut) • Der neue Wert „Ausstehend“ wurde hinzugefügt PartnerAcceptance.Status • Veröffentlichte SDKs für Dotnet, Java, Javascript, Python, Ruby. • Das API-Referenzhandbuch wurde aktualisiert, um die oben genannten Änderungen widerzuspiegeln.	2024-04-19

Änderungen	Beschreibung	Date
Beta 4	• Sandbox-Tests hinzugefügt: Möglichkeit, die Benachrichtigung über Opportunity-Ereignisse in der Sandbox zu testen. • Dokumentation hinzugefügt: Beispielregeln für die Benachrichtigung über Opportunity-Ereignisse wurden eingeführt. • SDK-Versionen hinzugefügt: Veröffentlichte SDKs für Java (V2), Javascript (V2), DotNet (V3) und Python (V3). • Aktualisiertes Datumsformat: Es wurde aktualisiert, um den ISO-Standards zu entsprechen.	2024-03-04

Änderungen	Beschreibung	Date
Betaversion 3	• Änderungsprotokolldatei hinzugefügt: Es wurde eine Änderungsprotokolldatei eingeführt, um Aktualisierungen und Änderungen besser verfolgen zu können. • Aktualisiertes API-Referenzhandbuch (PDF): Eine neue Version des API-Referenzhandbuchs im PDF-Format wurde veröffentlicht. • Aktualisiertes API-Referenzhandbuch (Webversion): Die Webversion des API-Referenzhandbuchs wurde aktualisiert. • DotNet SDK hinzugefügt: Das DotNet SDK wurde veröffentlicht, wodurch unsere Unterstützung für verschiedene Programmierumgebungen erweitert wurde. • Aktualisiertes Python Boto 3 SDK (V2): Version 2 des Python Boto 3 SDK veröffentlicht, die neue Funktionen und Verbesserungen einführt.	2024-01-24

Änderungen	Beschreibung	Date
Betaversion 2	• API-Referenzhandbuch (Webversion) hinzugefügt: Eine neue Version des API-Referenzhandbuchs (Webversion) wurde veröffentlicht. Hinweis: Für diese Version muss nach dem Extrahieren ein lokaler Webserver ausgeführt werden, um eine einwandfreie Funktionalität zu gewährleisten.	2024-01-07
Betaversion 1	• Boto3 (Erstveröffentlichung) hinzugefügt: Die erste Version von Boto3, einem Python-SDK für unsere Dienste, wurde gestartet. • Leitfaden für das Beta-Programm hinzugefügt: Es wurde ein Leitfaden für Teilnehmer unseres Betatestprogramms eingeführt. • API-Referenzhandbuch (PDF) hinzugefügt: Die erste Version unseres API-Referenzleitfadens wurde im PDF-Format hochgeladen, das eine umfassende Dokumentation für unsere API enthält.	2023-12-15

Änderungen	Beschreibung	Date
Betaversion 0	Diese erste Version bietet eine Reihe von Funktionen zur Verwaltung und Optimierung von Partnerengagements und Geschäftschancen in AWS Partner Central.	2023-11-15

Dokumentverlauf

Im Folgenden finden Sie eine Liste der Dokumentationsversionen für diese API-Referenz.

Änderung	Beschreibung	Datum
Dokumentation für APIs zur Messung von Partnererlösen hinzugefügt	AWS Partner Central hat neue APIs für Partner Revenue Measurement (PRM) hinzugefügt.	30. Juni 2026
Die Dokumentation für ACE Resonate wurde aktualisiert	AWS Partner Central hat die API-Antwort für GetAwsOpportunitySummary Maßnahmen zur Unterstützung von Cosell Motion aktualisiert.	16. Juni 2026
Dokumentation für Partner Central MCP Server hinzugefügt	AWS Der MCP-Server für Partner Central Agents stellt über das Model Context Protocol (MCP) Tools für Partner Central bereit.	16. März 2026
RC 5-Veröffentlichung	Die Candidate 5-Version der AWS Partner Central-API wurde veröffentlicht. Es wurde eine neue Konto-API in AWS Partner Central eingeführt, mit der Partner ihre Kontoinformationen, Registrierungen, Profile, Domainverwaltung und Partnerverbindungen verwalten können. Es wurde eine neue Benefits API in AWS Partner Central eingeführt, die es Partnern ermöglicht, über eine zentrale	30. November 2025

Schnittstelle Leistungen in allen AWS Partnerprogrammen zu finden, zu beantragen und zu verwalten. Verbesserte Aktionen zur Interaktion mit der Verkaufs-API zur Unterstützung des Lead-Management-Workflows. Es wurden umfassende Nutzungshandbücher und API-Dokumentation für Account- und Benefit-APIs mit detaillierten Workflows und Integrationsmustern hinzugefügt. Verbesserte Verkaufs-API-Dokumentation für das Engagement-Management mit Lead-Kontext.

RC 4-Veröffentlichung

Die Version Candidate 4 der AWS Partner Central API wurde veröffentlicht. Einführung der Channel-API zur Verwaltung von Kanalbeziehungen und der Zusammenarbeit zwischen Partnern, sodass sich Partner bei Transaktionen mit Endkunden über Billing Transfer für Programmvorteile qualifizieren können. Die Dokumentation wurde neu organisiert, um mehrere Partner Central-APIs zu unterstützen. AWS

19. November 2025

Es wurde eine Dokumentation für Möglichkeiten mit mehreren Partnern hinzugefügt	Multi-partner Opportunities (in der Vorschauversion) ist ein integriertes Erlebnis in AWS Partner Central, das es AWS Partnern ermöglicht, andere Partner zu finden und mit ihnen in Kontakt zu treten und gemeinsam Kundendeals zu verkaufen.	4. Dezember 2024
RC 3-Veröffentlichung	Die Candidate 3-Version der AWS Partner Central-API wurde veröffentlicht	17. Oktober 2024
RC 2-Version	Candidate 2-Version der AWS Partner Central-API veröffentlicht	7. August 2024
RC 1-Version	Kandidaten-1-Version der AWS Partner Central-API veröffentlicht	21. Juni 2024
Version Beta 5	Beta 5 der AWS Partner Central API veröffentlicht	19. April 2024
Beta 4-Version	Beta 4 der AWS Partner Central API veröffentlicht	4. März 2024
Beta 3-Version	Beta 3 der AWS Partner Central API wurde veröffentlicht	24. Januar 2024
Beta 2-Version	Beta 2 der AWS Partner Central API veröffentlicht	7. Januar 2024
Veröffentlichung von Beta 1	Beta 1 der AWS Partner Central API wurde veröffentlicht	15. Dezember 2023

[Beta 0-Version](#)

AWS Erste Version der
Partner Central API

15. November 2023