



개발자 가이드

AWS 파트너 센트럴



AWS 파트너 센트럴: 개발자 가이드

Table of Contents

.....	vi
환영합니다	1
AWS SDKs 작업	1
설정 및 인증	2
를 Partner Central AWS 계정에 연결	2
IAM 설정	2
API 호출 인증	3
사용자 지정 사용자 에이전트를 사용하여 호출에 서명	4
파트너 센트럴 에이전트 MCP 서버	7
개요	7
에이전트 기능	7
주요 이점	8
사용 예제	8
기회 생성	8
기회 복제	9
파이프라인 인사이트	9
기회 요약	9
영업 플레이 생성	9
고객 프로필 생성	10
솔루션 권장 사항	10
자금 지원 권장 사항	10
다음 단계 권장 사항	11
기회 진행	11
파트너 온보딩을 위한 에이전트 경험	11
파트너 프로필 자동화	12
판매자 설정 지침	12
PRM 규정 준수 지침	12
시작하기	13
시작하기	13
사전 조건	13
1단계: IAM 권한 설정	13
2단계: MCP 클라이언트 연결	20
MCP 헤더를 사용하여 호출에 서명	22
3단계: 설정 확인	23

4단계: 첫 번째 태스크 실행	24
보안 고려 사항	26
다음 단계	26
구성 참조	27
엔드포인트	27
IAM 권한	27
카탈로그 환경	32
세션 관리	32
파일 업로드	32
오류 코드	33
속도 제한	34
스트리밍(SSE) 이벤트 유형	34
다음 단계	35
도구 참조	35
도구 개요	36
sendMessage	36
getSession	44
오류 처리	46
APIs 정보	47
API 사용	47
계정 API 사용	47
Selling API 사용	59
Benefits API 사용	110
채널 API 사용	123
수익 측정 API 사용	125
지원되는 리전:	135
계정 API의 리전	136
Selling API의 리전	136
Benefits API의 리전	136
채널 API의 리전	136
수익 측정 API의 리전	137
권한	138
계정 API에 대한 액세스	138
Selling API에 대한 액세스	141
Benefits API에 대한 액세스	147
채널 API에 대한 액세스	149

수익 측정 API에 대한 액세스	157
할당량	166
계정 API에 대한 할당량	166
판매 API 할당량	169
Benefits API 할당량	171
채널 API 할당량	173
수익 측정 API의 할당량	176
로깅	178
계정 API 로깅	179
Selling API 로깅	182
Benefits API 로깅	183
채널 API 로깅	185
수익 측정 API 로깅	188
알림	190
AWS파트너 중앙 계정 API에 대한 알림	191
AWS Partner Central Selling API에 대한 알림	202
AWS Partner Central Benefits API에 대한 알림	216
AWS파트너 중앙 채널 API에 대한 알림	229
샌드박스에서 테스트	239
샌드박스 환경에 대한 액세스	239
샌드박스 환경에 대한 중요한 세부 정보	239
계정 API용 샌드박스에서 테스트	240
샌드박스에서 Selling API 테스트	243
샌드박스에서 Benefits API 테스트	247
채널 API용 샌드박스에서 테스트	252
수익 측정 API에 대해 샌드박스에서 테스트	256
코드 예제	261
기본 사항	262
작업	262
시나리오	328
기회에 연결된 엔터티 업데이트	328
릴리스 노트	334
문서 기록	361

AWS Partner Central API 참조가 재구성되었습니다. 지원되는 API 작업에 대한 자세한 내용은 [AWS Partner Central API 참조](#)를 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.

AWS Partner Central 개발자 안내서에 오신 것을 환영합니다.

이 개발자 안내서에서는 AWS 파트너가 서비스 APIs 사용하여 빌드, 마케팅, 판매 및 성장 활동을 통합하고 관리하는 방법을 설명합니다 AWS.

주제

- [AWS SDK에서 AWS Partner Central API 사용](#)
- [설정 및 인증](#)

AWS SDK에서 AWS Partner Central API 사용

AWS 소프트웨어 개발 키트(SDKs)는 널리 사용되는 많은 프로그래밍 언어에 사용할 수 있습니다. 각 SDK는 개발자가 선호하는 언어로 애플리케이션을 쉽게 구축할 수 있도록 하는 API, 코드 예제 및 설명서를 제공합니다.

SDK 설명서	코드 예제
AWS SDK for C++	AWS SDK for C++ 코드 예제
AWS CLI	AWS CLI 코드 예제
AWS SDK for Go	AWS SDK for Go 코드 예제
AWS SDK for Java	AWS SDK for Java 코드 예제
AWS SDK for JavaScript	AWS SDK for JavaScript 코드 예제
AWS SDK for Kotlin	AWS SDK for Kotlin 코드 예제
AWS SDK for .NET	AWS SDK for .NET 코드 예제
AWS SDK for PHP	AWS SDK for PHP 코드 예제
AWS Tools for PowerShell	AWS Tools for PowerShell 코드 예제
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) 코드 예제
AWS SDK for Ruby	AWS SDK for Ruby 코드 예제

SDK 설명서	코드 예제
AWS SDK for Rust	AWS SDK for Rust코드 예제
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP코드 예제
AWS SDK for Swift	AWS SDK for Swift코드 예제

예제 가용성

필요한 예제를 찾을 수 없습니까? 이 페이지 하단의 피드백 제공 링크를 사용하여 코드 예시를 요청하세요.

설정 및 인증

AWS Partner Central API 설정 및 인증에는 세 단계가 포함됩니다. 다음은 프로세스에 대한 개요입니다.

1. AWS Marketplace판매자 계정을 Partner Central에 연결합니다.
2. IAM을 사용하여 권한을 설정합니다.
3. 서명 버전 4(SigV4)를 사용하여 API 호출을 인증합니다.

를 Partner Central AWS 계정에 연결

를 Partner Central AWS 계정에 연결하는 것은 API를 사용하기 위한 사전 조건입니다. 자세한 내용은 [AWS Partner Central 계정과AWS Marketplace판매자 계정 연결을 참조하세요](#). Alliance-lead 또는 Cloud-administrator 권한이 있는 계정으로 Partner Central에 로그인하고 **Account Linking** 섹션으로 이동하여 프롬프트를 따라야 합니다.

IAM 설정

AWS Partner Central API를 사용하려면 호출을 시작하려면AWS Identity and Access Management(IAM) 역할 또는 IAM 사용자가 필요합니다. 자세한 내용은 [언제 IAM을 사용하나요?](#)를 참조하세요. 이에 대한AWS 계정가이드의 [IAM 역할 생성](#) 및 [IAM 사용자 생성](#) 단계를 따릅니다. 파트너 중앙 연결AWS Marketplace판매자 계정에서이 IAM 역할/사용자를 생성해야 합니다. IAM 역할/사용자 생성에는 비용이 발생하지 않습니다.

1. IAM 역할/사용자 생성

에 로그인하고 IAM 서비스로 AWS Management Console 이동한 다음 단계에 따라 IAM 역할 또는 IAM 사용자를 생성합니다.

2. 정책 할당:

필요에 따라 관리형 정책을 연결하거나 사용자 지정 정책을 생성합니다. 권한을 수정하거나 확장하려면 콘텐츠 복사하고 다른 권한 `AWSPartnerCentralOpportunityManagement`과 결합하는 대신 IAM 역할에 추가 정책을 적용합니다. 관리형 정책을 복제하지 마세요. 이렇게 하면 새 기능이 릴리스될 때 새 기능에 자동으로 액세스할 수 없으며 향후 정책을 수동으로 업데이트해야 하기 때문입니다. 액세스 정책에 대한 자세한 내용은 [액세스 제어 설명서](#)를 참조하세요.

AWS Marketplace 제안 관리

AWS Marketplace 제안을 관리하고 기회에 연결하려면 파트너는 IAM 역할에 카탈로그 APIs에 액세스할 수 있는 권한을 부여해야 합니다. 역할/사용자에게 `aws-marketplace:ListEntities` 및와 같은 권한이 있는지 확인합니다 `aws-marketplace:SearchAgreements`.

API 호출 인증

AWS Partner Central API는 인증에 서명 버전 4(SigV4)를 사용합니다. 구현 방법은 다음과 같습니다.

AWS SDK 사용

AWS SDKs 요청 서명을 자동으로 처리합니다. 자격 증명(AWS 자격 증명)을 제공하면 SDK가 나머지 작업을 수행합니다.

1. Java의 경우 [Java용 AWS SDK에 임시 자격 증명 제공을 참조하세요](#).
2. Python(Boto3)의 경우 [자격 증명을 참조하세요](#).
3. JavaScript(Node.js)의 경우 [Node.js에서 자격 증명 설정을 참조하세요](#).
4. .NET의 경우 [자격 증명 및 프로필 확인을 참조하세요](#).
5. 다른 프로그래밍 언어 및 추가 예제는 [여기 빌드할 도구를 AWS](#) 참조하세요.

AWS SDK를 사용하지 않는 인증

선택한 프로그래밍 언어에 AWS SDK를 사용할 수 없는 경우 인증에는 표준 요청을 수동으로 생성하고, 요청에 서명하고, 세션 토큰을 처리하는 작업이 포함됩니다.는 [SigV4 서명 사용에](#) 대한 포괄적인 지침

을 AWS에 제공합니다. 그러나 수동 요청 서명은 복잡성을 높이고 보안 토큰을 신중하게 관리해야 하므로 AWS SDK를 사용하는 것이 좋습니다.

awsJson1_0 프로토콜에 대한 모든 요청은 다음 헤더를 사용하는 HTTP "POST" 메서드를 사용하여 루트 URL(/)로 전송해야 합니다.

필수 헤더

awsJson1_0 프로토콜에 대한 모든 API 요청은 다음 헤더를 사용하는 HTTP "POST" 메서드를 사용하여 루트 URL(/)로 전송해야 합니다.

헤더	필수	설명
Content-Type	true	이 헤더의 정적 값은 <code>application/x-amz-json-1.0</code> 입니다.
Content-Length	true	RFC 9110#section-8.6 에서 정의한 표준 Content-Length 헤더입니다.
X-Amz-Target	요청에 대해 true	예를 들어 <code>CreatePartner</code> 서비스 작업의 값은 <code>PartnerCentralAccount</code> 입니다. <code>PartnerCentralAccount.CreatePartner</code> .

사용자 지정 사용자 에이전트를 사용하여 호출에 서명

AWS Partner Central에 API를 요청할 때 클라이언트 애플리케이션의 소스를 AWS식별하고, 사용량을 모니터링하고, 성능을 감사하는 데 도움이 되는 X-Amzn-User-Agent 헤더를 포함하는 것이 좋습니다. 이 헤더를 AWS 사용하여 호출하는 클라이언트 애플리케이션의 유형을 구분하고, 다양한 클라이언트 구현의 성공률에 대한 인사이트를 수집합니다.

사용자 지정 사용자 에이전트 헤더

헤더 이름: X-Amzn-User-Agent

용도: 상호 작용의 소스를 분류하여 API 요청을 수행하는 클라이언트 유형을 구분합니다.

형식: {Integrator}|{Product}

예제 값: AWS|AWS Partner CRM Connector

모든 요청에 이 헤더를 포함하면 AWS가 요청 패턴을 분석하고, 통합을 추적하고, 다양한 클라이언트 시스템의 API 경험을 개선할 수 있습니다.

SDKs에서 사용자 지정 헤더 사용

SDK 호출에 X-Amzn-User-Agent 헤더를 포함하려면 API 호출을 수행하기 전에 클라이언트 요청 동작을 수정할 수 있습니다. 다음은 AWS SDK for Python(Boto3)을 사용하는 판매 API 예제입니다.

```
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
```

이 예제에서는 Boto3 SDK에 이벤트를 등록하여 모든 API 요청에 X-Amzn-User-Agent 헤더를 자동으로 추가하는 방법을 보여줍니다. 각 요청 가로채기 메커니즘을 수정하여 다른 AWS SDKs에 동일한 접근 방식을 적용할 수 있습니다.

 Note

X-Amzn-User-Agent 헤더 형식이 간소화되었습니다. 새 형식을 사용하는 것이 좋습니다. 이전 형식(CompanyName|ProductName|CRMName|ProductVersion)은 이전 버전과의 호환성을 위해 계속 지원됩니다. 기존 통합은 수정 없이 계속 작동합니다.

파트너 센트럴 에이전트 MCP 서버

파트너 센트럴 에이전트 MCP 서버는 모델 컨텍스트 프로토콜(MCP)을 통해 파트너 센트럴 도구를 제공하므로 AI 에이전트와 도구가 자연어를 통해 기회 관리, 고객 인사이트 및 자금 조달 프로그램을 검색하고 상호 작용할 수 있습니다.

서버는 워크플로를 간소화하면서 제어를 유지하는 쓰기 작업에 대한 인증, 세션 관리 및 human-in-the-loop 승인을 처리합니다.

개요

Partner Central 에이전트 MCP Server는 MCP 호환 AI 클라이언트를 Partner Central 에이전트에 연결하는 완전 관리형 AWS 호스팅 서비스입니다. SigV4 인증과 함께 HTTPS를 통한 JSON-RPC 2.0을 사용하며 실시간 스트리밍 응답을 위해 Server-Sent Events(SSE)를 지원합니다.

서버는 멀티턴 대화, 문서 분석을 위한 파일 첨부 파일, 쓰기 작업이 실행되기 전에 사용자의 명시적 동의가 필요한 기본 제공 승인 워크플로를 지원합니다.

에이전트 기능

- 파이프라인 인사이트 - 위험 기회, 단계 진행, 손실 종료 분석 등 영업 파이프라인에 대한 대화형 인텔리전스 확보
- 기회 생성 - 자연어 대화, 회의 메모, 제안 또는 통화 기록을 업로드하거나 기존 기회를 복제하여 새로운 기회를 생성합니다.
- 기회 요약 - 단계, 지출, 종료일 등을 다루는 모든 거래에 대한 간결한 at-a-glance 생성합니다.
- 영업 플레이 생성 - 기회 세부 정보, 업계 컨텍스트 및 AWS 솔루션 권장 사항을 결합한 맞춤형 영업 전략 구축
- 고객 프로필 생성 - 업계, 비즈니스 모델, 지역 및 최근 개발에 대해 공개적으로 사용 가능한 정보를 사용하여 회사 프로필을 생성합니다.
- 솔루션 권장 사항 - 등록된 솔루션을 기회 요구 사항과 상호 참조하여 가장 적합한 솔루션 찾기
- 자금 권장 사항 - 사용 가능한 AWS 자금 프로그램을 기준으로 기회를 평가하고, 금액을 추정하고, 자금 요청을 생성합니다.
- 다음 단계 권장 사항 - AWS 공동 판매 표준 및 단계 진행 지침을 기반으로 우선순위가 지정된 행동 계획 받기

- 기회 진행 - 지원 문서 업로드, 관련 데이터 추출, 파이프라인 단계를 통한 진행 기회
- AI 지원 제품 목록 - 기존 디지털 자산에서 고품질AWS Marketplace제품 목록을 생성하고,AWS Marketplace표준에 대한 점수 목록 강도를 평가하며, 검색 가능성을 개선하기 위한 필드 수준 권장 사항을 받습니다. 자세한 내용은 AWS Marketplace판매자 안내서의 [AI 지원 제품 목록](#)을 참조하세요.

주요 이점

- Partner Central에 대한 대화 액세스 - 복잡한 콘솔 워크플로를 탐색하는 대신 자연어로 질문합니다.
- 더 많은 시간 판매 - 다단계 양식을 작성하는 대신 짧은 대화를 통해 기회를 생성하여 데이터 입력을 줄이고 파트너 영업 팀이 더 많은 시간을 판매할 수 있도록 합니다. 에이전트는 파트너가 더 높은 품질의 기회를 제출하고 파이프라인 위생을 개선할 수 있도록 개선 사항을 추천합니다.
- Human-in-the-loop 모든 쓰기 작업은 실행 전에 명시적 승인이 필요합니다.
- 멀티턴 대화 - 컨텍스트를 반복하지 않고 세션 내에서 쿼리를 구체화합니다.
- 파일 분석 - 에이전트가 질문과 함께 분석할 문서(PDF, DOCX, XLSX, CSV, 이미지) 첨부
- 중앙 집중식 액세스 관리 - 세분화된 권한으로 IAM 정책을 통해 액세스 제어
- 샌드박스 테스트 - 프로덕션 데이터를 터치하기 전에 격리된 샌드박스 환경에서 워크플로를 테스트 합니다.
- 스트리밍 응답 - 에이전트가 요청을 처리할 때 SSE를 통해 피드백 받기

사용 예제

모든 AI 생성 인사이트에는 추적성을 위한 세션 ID가 포함됩니다. 데이터는 로그인한 파트너의 자체 기회로 격리됩니다. 모든 콘텐츠에는 명확한 공개 레이블이 있으며 [AWS책임 있는 AI 정책](#)이 적용됩니다.

기회 생성

에이전트는 자연어 설명 또는 업로드된 문서(PDF, DOCX, XLSX, TXT)에서 새로운 기회를 생성합니다. 고객 이름, 프로젝트 범위, 예상 종료일 및 기타 필수 필드를 추출하고, 공개적으로 사용 가능한 웹 데이터에서 고객 세부 정보를 보강하고, AWS 준비 요구 사항에 따라 초안을 검증하고, 승인 후 Partner Central Selling API를 통해 기회를 생성합니다.

- “Acme Corp을 위한 기회 생성 - 데이터 웨어하우스를 Redshift로 마이그레이션하고, 목표 종료일이 Q3로 끝나고, 예상 월별 AWS 지출은 4만 달러입니다.”

- “다음은 어제 GlobalTech와의 회의에서 내 통화 메모입니다.이 트랜스크립트에서 기회를 생성하세요.”
- “이 제안 PDF를 사용하여 고객의 SAP 마이그레이션 기회를 만드세요.”

기회 복제

에이전트는 기존 고객 또는 프로젝트 세부 정보에서 새 기회를 생성하고, 제공한 새 고객 또는 프로젝트 세부 정보를 병합하고, 제출 전에 하나 이상의 차별화 필드가 변경되어 중복이 AWS 검토에서 거부되지 않는지 확인합니다.

- "신규 고객에 대한 기회 O1234567890 해지 - Globex, 동일한 워크로드, 목표 종료일 Q4 종료"
- “O9876543210과 같은 기회를 생성하되 샌드박스 대신 고객의 프로덕션 환경을 위한 기회를 생성합니다.”

파이프라인 인사이트

영업 파이프라인에 대한 대화형 인텔리전스를 얻습니다. 에이전트는 스테이지 진행 상황, 기한, 지연된 거래 및 손실 종료 패턴을 분석하여 가장 중요한 것을 표시합니다.

- “이번 주에 어떤 기회에 주의를 기울여야 하나요?”
- “다음 달에 마감할 기회는 몇 개입니까?”
- “지난 6개월 동안 기회를 잃은 가장 큰 이유는 무엇인가요?”

기회 요약

에이전트는 회사 이름, 산업, 단계, 예상 월별AWS지출, 목표 마감 날짜 및 기타 주요 세부 정보를 간결한 요약으로 합성하므로 개별 양식 필드를 스캔할 필요가 없습니다.

- "기회 O1234567890 요약 제공"
- “Acme Corp 거래의 현재 상태와 주요 세부 정보는 무엇입니까?”

영업 플레이 생성

에이전트는 기회의 세부 정보, 고객의 업계 컨텍스트 및 관련AWS솔루션 권장 사항을 ready-to-use 수 있는 영업 플레이에 결합하여 맞춤형 영업 전략을 구축합니다.

- “기회 O1234567890을 위한 영업 전략 생성”
- “이 금융 서비스 고객에게 클라우드 마이그레이션을 판매하는 가장 좋은 방법은 무엇인가요?”
- “GlobalTech 데이터 분석 거래를 위한 영업 전략 수립”

고객 프로필 생성

에이전트는 업계 분류, 비즈니스 모델(B2B/B2C/하이브리드), 지리적 입지, 회사 규모, 시장 포커스, 최근 비즈니스 개발 등 공개적으로 사용 가능한 정보를 사용하여 회사 프로필을 생성합니다. 프로필에는 “공개적으로 사용 가능한 데이터 및 AWS AI 인사이트로 생성됨” 레이블이 지정됩니다.

- “Acme Corp에 대한 고객 프로필 생성”
- “이 고객의 산업 및 비즈니스 모델에 대해 무엇을 알고 있나요?”
- “GlobalTech와의 예정된 회의에 대한 회사 개요를 함께 작성”

솔루션 권장 사항

에이전트는 등록된 솔루션을 기회 요구 사항과 상호 참조하여 솔루션 이름, 설명 및 이미 기회에 연결되어 있는지 여부를 보여줍니다.

- “O1234567890과 가장 일치하는 솔루션은 무엇입니까?”
- “데이터 분석 솔루션이 이미이 거래에 연결되어 있나요?”
- “SAP 워크로드를 마이그레이션하려는 고객에게 솔루션 권장”

자금 지원 권장 사항

에이전트는 기회 세부 정보 및 프로그램 자격 기준을 기반으로 사용 가능한 AWS 자금 조달 프로그램을 기준으로 각 기회를 평가합니다. 일치하는 항목이 발견되면 프로그램 이름, 설명 및 자세한 추론이 표시됩니다. 그런 다음 자금 금액을 추정하거나, 자동 채워진 자금 요청 초안을 생성하거나, 대화식으로 프로그램에 대해 자세히 알아볼 수 있습니다. SCA(전략적 협업 계약) 예산 가용성은 적절한 경우 표시되며 모든 작업은 IAM 권한을 준수합니다.

- “기회 O6789012345에 대한 자금 지원 프로그램을 받을 수 있나요?”
- “이 고객의 POC에 대한 자금 금액 추정”
- “이 기회를 위한 MAP 혜택 애플리케이션 생성”

다음 단계 권장 사항

에이전트는 AWS의 단계 진행 지침을 기준으로 기회를 평가하고, 현재 데이터를 적합한 기회 기준과 비교하며, 여전히 수집해야 할 정보를 정확하게 식별합니다. 그 결과 AWS 공동 판매 표준을 기반으로 우선순위가 지정된 행동 계획이 수립됩니다.

- “기회 발전 O1234567890 다음에 무엇을 해야 하나요?”
- “이 기회를 제출할 준비가 되었나요? 누락된 필드는 무엇입니까?”
- “이 거래를 잠재 고객에서 적격 고객으로 전환하기 위한 요구 사항은 무엇입니까?”

기회 진행

에이전트는 지원 문서(회의 기록, 통화 메모, 이메일 요약)를 수락하고, 관련 정보를 추출하고, 기회 필드에 매핑하고, 단계 요구 사항을 평가하고, 기회를 업데이트합니다. 격차가 남아 있으면 충족된 요구 사항과 충족되지 않은 요구 사항의 분류가 반환됩니다.

- “다음은 내 통화 메모입니다. 기회 O1234567890을 관련 세부 정보로 업데이트하세요.”
- “회의 기록을 첨부하고 있습니다. 논의한 내용을 기반으로 이 기회를 진행하십시오.”
- “이 이메일 요약을 검토하고 어떤 기회 필드를 충족하는지 알려주세요.”

파트너 온보딩을 위한 에이전트 경험

에이전트는 파트너 센트럴에 대한 파트너 온보딩을 자동화하고 Marketplace 판매자 설정 및 PRM 규정 준수에 대한 안내 지원을 제공합니다. 이 모든 것이 자연어를 통해 이루어집니다.

에이전트 기능:

- 파트너 프로필 자동화 - 웹 사이트를 스캔하고, 파트너 프로필을 자동으로 채우고, 가시성, 제휴 리드 연락처, 교육 도메인 및 계정 연결을 관리합니다.
- 판매자 설정 지침 Step-by-step 지침 KYC/BAV/SU
- PRM 규정 준수 지침 - PRM 준비 상태 평가, 수익 태그 지정을 위한 제품 코드 검색, 통합 수익 어트리뷰션을 위한 자회사 계정 연결 확인

파트너 프로필 자동화

에이전트는 회사 웹 사이트를 스캔하여 파트너 프로필을 추출하고 자동으로 채운 다음 남은 공백을 안내합니다. 프로필 필드를 업데이트하고, 가시성을 변경하고, 제휴 책임자 연락처를 관리하고, 훈련 인증 도메인을 연결하고, 계정 연결 초대를 처리할 수 있습니다.

- “웹 사이트를 보고 프로필을 작성할 수 있나요?”
- “AWS 고객이 당사를 찾을 수 있도록 프로필 공개”
- “[이메일]의 [이름]으로 제휴 책임자 연락처 업데이트”
- “EMEA 자회사 계정 연결”

판매자 설정 지침

에이전트는 Marketplace 판매자 등록end-to-end 살펴보고, 국가 및 법인 유형에 따라 올바른 세금 양식을 결정하고, 지역별 은행 및 지출 설정을 설명하고, 사용자에게 적용되는 규정 준수 요구 사항(KYC, 은행 계좌 확인, 보조 사용자 확인)을 식별합니다.

- “마켓플레이스에서 판매를 시작하고 싶습니다. 어디서 시작해야 하나요?”
- “어떤 세금 양식이 필요합니까? 독일에 있는 회사입니다.”
- “KYC가 필요합니까? 프랑스 고객에게 판매하고 있습니다.”
- “미국 외부에 있는 경우 결제를 설정하려면 어떻게 해야 하나요?”

PRM 규정 준수 지침

에이전트는 계정이 PRM에 준비되었는지 확인합니다. 즉, 계정이 연결되어 있고, 목록이 존재하며, 제품 코드에 태그를 지정할 수 있는지 확인합니다. 자회사 계정이 있는 파트너의 경우 모든 판매자 계정이 통합 수익 어트리뷰션에 올바르게 연결되어 있는지 확인합니다.

- “PRM을 준수하나요?”
- “태깅할 제품 코드는 어디에서 찾을 수 있나요?”
- “PRM을 준수하기 위해 수행해야 하는 모든 단계를 말해 주세요.”

시작하기

Partner Central 에이전트 MCP 서버를 설정할 준비가 되셨나요? step-by-step 설정 지침은 [시작하기](#) 안내서를 참조하십시오.

Partner Central Agent MCP 서버 시작하기

이 가이드에서는 사용자 지정 MCP 클라이언트를 사용하여 Partner Central Agent MCP 서버에 대한 프로그래밍 방식 액세스를 설정하는 방법을 안내합니다. 서버는 프록시 또는 IDE 플러그인 없이 SigV4 인증과 함께 직접 HTTPS를 사용합니다.

사전 조건

시작하기 전에, 다음 사항을 확인해야 합니다.

- 활성 Partner Central 계정(AWS콘솔로 마이그레이션됨)
- Partner Central에 대한 IAM 권한이 있는AWS계정
- AWS자격 증명으로 CLI 설치 및 구성
- us-east-1(버지니아 북부) 리전에 대한 액세스
- 에 대한 HTTPS 연결 `partnercentral-agents-mcp.us-east-1.api.aws`
- HTTP 클라이언트에서 TLS 1.2+ 지원
- JSON-RPC 2.0 및 SigV4 요청 서명을 지원하는 MCP 호환 클라이언트

1단계: IAM 권한 설정

Partner Central Agent MCP 서버에는 프로토콜 액세스(MCP 엔드포인트와 통신)와 데이터 액세스(Partner Central 작업 수행)의 두 가지 수준에서 IAM 권한이 필요합니다.

IAM 정책 연결

AWS관리 콘솔을 사용하여 IAM 자격 증명에 정책을 연결하려면:

1. [IAM 콘솔](#)을 엽니다.
2. 왼쪽 탐색 창에서 정책을 연결할 자격 증명에 따라 사용자, 사용자 그룹 또는 역할을 선택한 다음 특정 사용자, 그룹 또는 역할의 이름을 선택합니다.
3. 권한 탭을 선택합니다.
4. 정책 연결을 선택합니다(또는 처음인 경우 권한 추가).

5. 정책 목록에서 연결하려는 관리형 정책(예: 아래 JSON 블록에서 생성한 사용자 지정 정책)을 검색하고 선택합니다.
6. 정책 연결(또는 다음 및 권한 추가)을 선택하여 확인합니다.

권한은 즉시 적용됩니다. 여러 정책을 동일한 자격 증명에 연결할 수 있습니다.

권장 사항: 관리형 정책 사용

MCP 프로토콜 액세스 권한을 부여하는 가장 간단한 방법은 `AWSMcpServiceActionsFullAccess` 관리형 정책을 IAM 자격 증명에 연결하는 것입니다. 이 정책에는 MCP 서버와 상호 작용하는 데 필요한 모든 권한이 포함됩니다.

세분화된 제어를 위해 IAM 정책의 `aws:IsMcpServiceAction` 조건 키를 사용하여 특히 MCP 서비스 작업에 대한 권한의 범위를 지정할 수 있습니다.

MCP 프로토콜 액세스에 대한 최소 권한

MCP 서버와 상호 작용하려면 최소한 IAM 자격 증명에 다음 작업이 필요합니다.

작업	설명
<code>partnercentral:UseSession</code>	대화 세션을 생성, 업데이트 및 검색하는 데 필요합니다.

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

```
]
}
```

데이터 액세스 권한

에이전트를 통해 실제로 Partner Central 작업을 수행하려면 사용 사례에 따라 추가 권한이 필요합니다.

기회 관리:

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

자금 지원 프로그램:

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

Marketplace 액세스:

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

파트너 중앙 온보딩:

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

```
}
```

Marketplace 판매자 설정의 경우 다음을 추가합니다.

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

서비스 연결 역할 관리(재판매 권한 부여/CPPO)의 경우:

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

전체 액세스 정책

개발 및 테스트의 경우 모든 권한을 단일 정책으로 결합할 수 있습니다.

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

```

읽기 전용 정책

프로덕션 환경 또는 읽기 전용 사용 사례의 경우 읽기 작업에 대한 권한을 제한합니다.

```

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": "*"
    }
]
}'

```

2단계: MCP 클라이언트 연결

Partner Central Agent MCP 서버는 SigV4 요청 서명과 함께 직접 HTTPS를 사용합니다. 프록시 계층은 없습니다. MCP 클라이언트는 JSON-RPC 2.0 요청을 엔드포인트로 직접 전송합니다.

엔드포인트

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

Authentication

모든 요청은 다음을 사용하여 [AWS서명 버전 4](#)로 서명해야 합니다.

- 서비스 이름: partnercentral-agents-mcp
- 리전: us-east-1

MCP 연결 초기화

initialize 요청을 전송하여 프로토콜을 설정합니다.

```
{
```

```

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

```

예상 응답:

```

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

```

사용 가능한 도구 나열

```

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

```

MCP 헤더를 사용하여 호출에 서명

Partner Central 에이전트 MCP에 요청할 때는 클라이언트 애플리케이션의 소스를 AWS 식별하고, 사용량을 모니터링하고, 성능을 감사하는 데 도움이 되도록 다음 방법을 사용하여 사용자 지정 MCP 헤더를 포함하는 것이 좋습니다. 이 헤더를 AWS 사용하여 호출하는 클라이언트 애플리케이션의 유형을 구분하고 다양한 클라이언트 구현의 성공률에 대한 인사이트를 수집합니다.

방법 1: _meta 필드(프로그래밍 방식/상태 비저장 MCP)

MCP tools/call 요청을 직접 구성하는 코드의 경우 요청에 _meta 필드를 제공합니다.

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

방법 2: clientInfo(세션 기반 사용자 지정 에이전트)

세션을 설정하는 사용자 지정 MCP 클라이언트의 경우 clientInfo 필드 내에 MCP 헤더 정보를 제공합니다.

```
{
  "method": "initialize",
  "params": {
    "protocolVersion": "2024-11-05",
```

```

    "clientInfo": {
      "integrator": "<Integrator's Company Name / Direct>",
      "sourceProduct": "<Integrator's Application Name>"
    }
  }
}

```

의 필드 clientInfo:

- integrator - 회사 이름 또는 파트너의 경우 "직접"

예시: AWS

- sourceProduct - 제품/에이전트 이름

예시: AWS CRM Connector

방법 3: URL 파라미터(호스팅된 MCP만 해당)

통합자가 프로토콜 필드를 수정할 수 없는 호스팅 MCP 클라이언트에만 해당됩니다. URL 파라미터를 사용합니다.

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

3단계: 설정 확인

간단한 메시지를 보내 모든 것이 작동하는지 확인합니다. Sandbox 카탈로그를 사용하여 테스트합니다.

```

{
  "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"
  }
}

```

에이전트로부터 응답 "status": "complete"과 텍스트 응답을 받으면 설정이 올바르게 작동하는 것입니다. 응답에는 후속 메시지에 사용할 수 sessionId 있는 도 포함됩니다.

4단계: 첫 번째 태스크 실행

기회 쿼리

```

{
  "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"
    }
  }
}

```

자금 지원 자격 확인

```

{
  "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"
    }
  }
}

```

세션 기록 검색

```

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

```

파트너 온보딩

```

{
  "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"
  }
}

```

시도할 기타 온보딩 작업:

- “납세 인터뷰 프로세스를 안내해 주세요.”
- “웹 사이트를 보고 파트너 프로필을 작성할 수 있나요?”
- “마켓플레이스에서 판매할 준비가 되려면 여전히 무엇을 해야 하나요?”

보안 고려 사항

- MCP 도구 파라미터를 통해 AWS 자격 증명을 전달하지 마십시오. 인증은 전송 계층에서 SigV4 요청 서명을 통해 처리됩니다.
- 테스트 및 개발에 샌드박스 카탈로그를 사용합니다. "Sandbox" 카탈로그는 프로덕션 파트너 데이터에 영향을 주지 않는 격리된 환경을 제공합니다.
- 프로덕션에 최소 권한 IAM 정책을 적용합니다. 사용 사례를 모니터링하고 보고하려면 읽기 전용 정책을 사용합니다. 사용자가 기회를 업데이트하거나 자금 지원 애플리케이션을 제출해야 하는 경우에만 쓰기 권한을 부여합니다.
- 쓰기 작업을 주의 깊게 검토합니다. 서버는 모든 쓰기 작업에 human-in-the-loop 루프 승인을 사용합니다. 쓰기 작업이 제안되면 승인하기 전에 파라미터를 검토합니다.
- 세션 데이터는 일시적입니다. 세션은 생성 후 48시간 후에 만료됩니다. 장기 데이터 스토리지를 위해 세션에 의존하지 마세요.
- 파일 업로드는 임시 S3 버킷으로 이동합니다. 업로드된 파일은 일시적으로 저장되며 영구적으로 보존되지 않습니다. 자격 증명, 보안 암호 또는 기타 민감한 정보가 포함된 파일을 업로드하지 마십시오.

다음 단계

- [구성 참조](#) - 엔드포인트, IAM 작업, 세션 관리 및 오류 코드에 대한 전체 참조
- [도구 참조](#) - sendMessage 및 getSession 도구에 대한 세부 설명서

구성 참조

이 페이지에서는 Partner Central Agent MCP 서버에 연결하고 구성하기 위한 전체 참조를 제공합니다.

엔드포인트

속성	값
URL	<code>https://partnercentral-agents-mcp.us-east-1.api.aws/mcp</code>
리전	us-east-1 (버지니아 북부)만 해당
프로토콜	HTTPS를 통한 JSON-RPC 2.0
스트리밍	Server-Sent Events(SSE)
Authentication	AWS서명 버전 4
SigV4 서비스 이름	partnercentral-agents-mcp
TLS 요구 사항	TLS 1.2 이상

IAM 권한

달리 명시되지 않는 한 모든 IAM 작업은 `partnercentral:접두사`를 사용합니다.

MCP 프로토콜 액세스

MCP 프로토콜 액세스 권한을 부여하는 가장 간단한 방법은 `AWSMcpServiceActionsFullAccess` 관리형 정책을 IAM 자격 증명에 연결하는 것입니다. 이 정책에는 MCP 서버와 상호 작용하는 데 필요한 모든 권한이 포함됩니다. 세분화된 제어를 위해 IAM 정책의 `aws:IsMcpServiceAction` 조건 키를 사용하여 특히 MCP 서비스 작업에 대한 권한의 범위를 지정할 수 있습니다.

작업	설명
<code>partnercentral:UseSession</code>	대화 세션 생성, 업데이트 및 검색

기회 관리

작업	설명
<code>partnercentral:List*</code>	기회, 솔루션 및 기타 Partner Central 리소스 나열
<code>partnercentral:Get*</code>	개별 기회 및 기타 리소스에 대한 세부 정보 검색
<code>partnercentral:CreateOpportunity</code>	기회 생성
<code>partnercentral:UpdateOpportunity</code>	기회 필드 수정(스테이지, 종료일, 수익 등)
<code>partnercentral:SubmitOpportunity</code>	AWS검토할 기회 제출
<code>partnercentral:AssignOpportunity</code>	파트너 담당자에게 기회 할당
<code>partnercentral:AssociateOpportunity</code>	기회를 다른 리소스(솔루션 등)에 연결
<code>partnercentral:DisassociateOpportunity</code>	기회와 다른 리소스 간의 링크 제거
<code>partnercentral:StartEngagementFromOpportunityTask</code>	기회에서 참여를 시작할 수 있는 기회 제출

자금 지원 프로그램

작업	설명
<code>partnercentral:ListBenefitAllocations</code>	계정에 사용 가능한 혜택 할당 나열
<code>partnercentral:ListBenefitApplications</code>	제출된 혜택 애플리케이션 나열

작업	설명
<code>partnercentral:CreateBenefitApplication</code>	새 혜택 애플리케이션(MAP, POC, WMP) 생성
<code>partnercentral:GetBenefitApplication</code>	특정 혜택 애플리케이션의 세부 정보 검색
<code>partnercentral:UpdateBenefitApplication</code>	보류 중인 혜택 애플리케이션 업데이트
<code>partnercentral:AssociateBenefitApplicationResource</code>	리소스를 혜택 애플리케이션에 연결
<code>partnercentral:DisassociateBenefitApplicationResource</code>	혜택 애플리케이션에서 리소스 링크 제거

Marketplace 액세스

작업	설명
<code>aws-marketplace:DescribeEntity</code>	마켓플레이스 개체의 세부 정보 검색
<code>aws-marketplace:SearchAgreements</code>	마켓플레이스 계약 검색
<code>aws-marketplace:ListEntities</code>	마켓플레이스 엔터티 나열

온보딩 에이전트

파트너 계정 관리(**partnercentral**):

작업	설명
<code>partnercentral:ListPartners</code>	계정에 대한 파트너 등록 나열
<code>partnercentral:GetPartner</code>	파트너 등록 및 프로필 세부 정보 검색

작업	설명
<code>partnercentral:GetProfileVisibility</code>	현재 프로필 가시성 검색(PUBLIC/PRIVATE)
<code>partnercentral:GetAllianceLeadContact</code>	제휴 책임자 연락처 정보 검색
<code>partnercentral:GetProfileUpdateTask</code>	보류 중인 비동기 프로필 업데이트 상태 확인
<code>partnercentral:StartProfileUpdateTask</code>	파트너 프로필 업데이트 제출(비동기)
<code>partnercentral:CancelProfileUpdateTask</code>	보류 중인 프로필 업데이트 작업 취소
<code>partnercentral:PutProfileVisibility</code>	프로필 가시성을 PUBLIC 또는 PRIVATE로 변경
<code>partnercentral:PutAllianceLeadContact</code>	제휴 리드 연락처 업데이트(이름, 제목, 이메일)
<code>partnercentral:SendEmailVerificationCode</code>	이메일 기반 도메인/연락처 변경에 대한 확인 코드 전송
<code>partnercentral:AssociateAwsTrainingCertificationEmailDomain</code>	훈련 인증 추적을 위한 이메일 도메인 연결
<code>partnercentral:DisassociateAwsTrainingCertificationEmailDomain</code>	연결된 훈련 인증 이메일 도메인 제거
<code>partnercentral:GetAccountConnections</code>	활성 계정 연결 나열(예: 자회사 링크)
<code>partnercentral:GetConnectionInvitations</code>	보류 중인 전송 및 수신된 연결 초대 나열

작업	설명
<code>partnercentral:CreateConnectionInvitation</code>	다른AWS계정으로 연결 초대 전송
<code>partnercentral:CancelConnectionInvitation</code>	보류 중인 발신 연결 초대 취소
<code>partnercentral:RespondConnectionInvitation</code>	수신 연결 초대 수락 또는 거부
<code>partnercentral:CancelConnection</code>	활성 계정 연결 취소
<code>partnercentral:ManageConnectionPreferences</code>	연결된 계정 간 공유 기본 설정 가져오기 또는 업데이트

Marketplace 판매자 설정(**aws-marketplace**):

작업	설명
<code>aws-marketplace:ListEntities</code>	Marketplace 엔터티 나열(예: 판매자 엔터티)
<code>aws-marketplace:DescribeEntity</code>	Marketplace 개체의 전체 세부 정보 검색
<code>aws-marketplace:DescribeChangeSet</code>	제출된 변경 세트의 상태 확인
<code>aws-marketplace:StartChangeSet</code>	Marketplace 변경 세트 제출(예: 판매자 프로필 업데이트)

IAM - 서비스 연결 역할(**iam**):

작업	설명
<code>iam:GetRole</code>	Marketplace 재판매 권한 부여 서비스 연결 역할이 존재하는지 확인

작업	설명
iam:CreateServiceLinkedRole	AWSServiceRoleForMarketplaceResaleAuthorization 역할 만들기

카탈로그 환경

카탈로그	설명
"AWS"	프로덕션 환경. 모든 작업은 라이브 파트너 데이터에 영향을 미칩니다.
"Sandbox"	격리된 테스트 환경. 프로덕션 데이터에 영향을 미치지 않습니다. 개발 및 검증에 사용됩니다.

세션 관리

속성	값
세션 ID 형식	session- 접두사가 있는 UUID v4(예: session-550e8400-e29b-41d4-a716-446655440000)
세션 생성	가 없는 첫 번째 sendMessage 호출 시 자동 sessionId
세션 만료	세션 생성 후 48시간(비활동 기반이 아닌 절대 만료)

파일 업로드

속성	값
메시지당 최대 파일 수	3

속성	값
이미지 크기 제한	3.75MB
문서 크기 제한	4.5MB
허용된 확장	doc, docx, pdf, png, jpeg, xlsx, csv, txt
S3 버킷(프로덕션)	aws-partner-central-marketplace-ephemeral-writeonly-files
업로드 경로	s3://{bucket}/{aws-account-id}/
S3 URI 요구 사항	versionId 쿼리 파라미터를 포함해야 합니다.

업로드 워크플로

1. AWS계정 ID 접두사 아래의 해당 S3 버킷에 파일을 업로드합니다.
2. 업로드에서 versionId 반환된를 포함하여 S3 URI를 기록해 둡니다.
3. sendMessage 호출에 파일을 document 콘텐츠 블록으로 포함합니다.

S3 URI 형식의 예:

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

오류 코드

코드	이름	설명
-32001	AUTHENTICATION_FAILURE	SigV4 서명이 유효하지 않거나 자격 증명이 만료되었습니다.
-31004	TOOL_PERMISSION_DENIED	IAM 자격 증명에이 작업에 필요한 partnercentral: 작업이 없음

코드	이름	설명
-32002	ACCESS_DENIED	일반 액세스 거부됨(계정이 등록되지 않음, 리전 불일치 등)
-32004	LIMIT_EXCEEDED	속도 제한 또는 할당량을 초과했습니다. 지수 백오프를 사용하여 재시도합니다.
-30001	RESOURCE_NOT_FOUND	요청된 리소스(세션, 기회 등)가 존재하지 않음
-32600	INVALID_REQUEST	잘못된 JSON-RPC 요청 또는 잘못된 파라미터
-32603	INTERNAL_ERROR	서버 측 오류입니다. 요청을 다시 시도하세요.

속도 제한

서버는 계정당 속도 제한을 적용합니다.

연산	지속 속도	버스트
sendMessage	분당 요청 2개	10
기타 모든 작업	분당 요청 10개	20

이러한 제한을 초과하면 서버가 오류 코드-32004(LIMIT_EXCEEDED)를 반환합니다. 클라이언트에서 지터를 사용하여 지수 백오프를 구현하여 속도 제한을 정상적으로 처리합니다.

스트리밍(SSE) 이벤트 유형

stream가 sendMessage 호출true에서 로 설정된 경우 서버는 다음 이벤트 유형과 함께 Server-Sent Events를 반환합니다.

이벤트 유형	설명
stream_start	스트림 연결 설정됨
assistant-response.start	에이전트가 응답 생성을 시작했습니다.
assistant-response.delta	에이전트 응답의 증분 텍스트 청크
assistant-response.completed	에이전트가 응답 생성을 완료했습니다.
server-tool-use	에이전트가 내부 도구를 호출하고 있음(읽기 작업)
server-tool-response	내부 도구 호출 결과
tool_approval_request	에이전트가 쓰기 작업에 대한 인적 승인을 요청하고 있습니다.
stream_end	스트림 연결 닫기
done	SSE 스트림이 완료되었음을 나타내는 최종 이벤트

다음 단계

- [도구 참조](#) - sendMessage 및 getSession 도구에 대한 세부 설명서

도구 참조

파트너 중앙 에이전트 MCP 서버는 sendMessage 모든 에이전트 상호 작용과 세션 상태 getSession 검색이라는 두 가지 MCP 도구를 제공합니다. 기회 쿼리, 자금 지원 애플리케이션, 문서 분석 등 모든 Partner Central 운영은를 통해 자연어를 통해 처리됩니다sendMessage.

도구 개요

도구	설명	카테고리
sendMessage	Partner Central AI 에이전트에 메시지를 전송합니다. 텍스트, 파일 첨부 파일 및 human-in-the-loop 승인 응답을 지원합니다.	읽기/쓰기
getSession	대화 기록, 이벤트 및 메타데이터를 포함한 세션 상태를 검색합니다.	읽기 전용

sendMessage

모든 Partner Central AI 에이전트 상호 작용을 위한 기본 도구입니다. 이 도구를 사용하여 질문하고, 작업을 요청하고, 분석할 문서를 첨부하고, 쓰기 작업에 대한 승인 요청에 응답할 수 있습니다.

에이전트는 세션 내에서 대화 컨텍스트를 유지하므로 이전 컨텍스트를 반복하지 않고도 후속 질문을 할 수 있습니다.

파라미터

- **content (필수)** - 콘텐츠 블록의 배열입니다. 각 블록에는 블록 구조를 결정하는 type 필드가 포함되어야 합니다. 단일 메시지(예: 텍스트 + 문서 첨부 파일)에 여러 블록을 포함할 수 있습니다.

콘텐츠 블록 유형:

유형	필드	설명
text	type (필수), text (필수)	에이전트에게 전송된 사용자 메시지 텍스트
document	type (필수), filename (필수), s3Uri (필수)	에이전트가 분석할 파일 첨부 파일입니다. 예는 versionId

유형	필드	설명
		파라미터가 포함되어야 s3Uri 합니다.
tool_approval_response	type (필수), toolUseId (필수), decision (필수), message (선택 사항)	human-in-the-loop 승인 요청에 대한 응답

- catalog (필수) - 작업의 대상 환경입니다.

유효한 값: "AWS" (프로덕션), "Sandbox" (테스트)

- sessionId (선택 사항) - 계속할 기존 세션을 식별하는 UUID v4입니다. 새 세션을 생성하려면 생략합니다. 형식: session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxx

기본값: 새 세션이 자동으로 생성됩니다.

- stream (선택 사항) - 실시간 응답 전송을 위해 Server-Sent Events(SSE) 스트리밍을 활성화합니다.

유효값: true, false

기본값: false

응답

응답에는 다음이 포함됩니다.

필드	설명
sessionId	후속 메시지의 세션 식별자
status	응답 상태: "complete" "requires_approval" , 또는 "error"
content	에이전트의 응답 콘텐츠 블록 배열

예제

기본 문자 메시지(새 세션)

요청:

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

응답:

```
{
  "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"
  }
}
```

```
}
```

후속 메시지(기존 세션)

요청:

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

파일 첨부 파일

먼저 S3에 문서를 업로드한 다음 메시지에서 참조합니다.

```
{
  "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"
}
}
}

```

파일 업로드 제약 조건:

- 메시지당 최대 3개의 파일
- 이미지 크기 제한: 3.75MB
- 문서 크기 제한: 4.5MB
- 허용되는 확장: doc, docx, pdf, png, jpeg, xlsx, csv, txt
- 파일을 업로드해야 합니다. s3://{bucket}/{your-aws-account-id}/
- S3 URI에는 versionId 쿼리 파라미터가 포함되어야 합니다.

Human-in-the-loop 승인 워크플로

에이전트가 쓰기 작업을 수행해야 하는 경우(예: 기회 업데이트, 자금 지원 애플리케이션 제출) 제안된 작업 세부 정보가 포함된 "requires_approval" 상태를 반환합니다. tool_approval_response 콘텐츠 블록으로 응답해야 합니다.

1단계 - 에이전트가 승인을 요청합니다.

```

{
  "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"
}
}

```

2단계 - 작업 승인:

```

{
  "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"
    }
  }
}

```

2단계(대안) - 작업 거부:

```

{

```

```

    "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"
      }
    }
  }
}

```

2단계(대안) - 사용자 지정 응답으로 재정의:

```

{
  "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"
    }
  }
}

```


승인 결정 값:

결정	동작
"approve"	제안된 파라미터로 도구 실행
"reject"	도구를 실행하지 마십시오. 선택 사항으로 이유를 message 설명합니다.
"override"	를 통해 사용자 지정 응답 또는 수정된 지침 제공 message

SSE를 사용한 스트리밍

에이전트가 요청을 처리할 때 증분 응답 청크를 수신하도록 스트리밍을 활성화합니다.

요청:

```
{
  "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
    }
  }
}
```

서버는 SSE 이벤트 스트림으로 응답합니다.

```
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

전체 대화 기록, 이벤트 및 메타데이터를 포함하여 대화 세션의 현재 상태를 검색합니다. 이를 사용하여 세션 상태를 검사하거나, 과거 상호 작용을 검토하거나, 대화를 재개할 수 있습니다.

파라미터

- **sessionId** (필수) - 검색할 세션의 UUID입니다. 형식: session-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
- **catalog** (필수) - 세션이 속한 환경입니다.

유효값: "AWS", "Sandbox"

응답

Field	유형	설명
sessionId	문자열	세션 식별자
createdAt	문자열	세션 생성의 ISO 8601 타임스탬프
lastActivity	문자열	마지막 활동의 ISO 8601 타임스탬프
sequenceNumber	정수	현재 이벤트 시퀀스 번호
stateType	문자열	현재 세션 상태
events	배열	전체 대화 기록(사용자 메시지, 에이전트 응답, 도구 사용)
variables	객체	세션 변수 및 메타데이터
eventCount	정수	세션의 총 이벤트 수

예제

요청:

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

응답:

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

오류 처리

모든 오류는 JSON-RPC 2.0 오류 형식을 따릅니다.

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

오류 코드의 전체 목록과 그 의미는 [the section called “오류 코드”](#) 섹션을 참조하세요.

권장 재시도 전략

- -32004 (LIMIT_EXCEEDED): 1초부터 지수 백오프를 사용하여 재시도
- -32603 (INTERNAL_ERROR): 지수 백오프를 사용하여 최대 3회 재시도
- -32001 (AUTHENTICATION_FAILURE): 자격 증명 새로 고침 및 재시도
- 다른 모든 오류의 경우: 자동으로 재시도하지 않음 - 오류 메시지를 검사하고 요청을 수정합니다.

AWS Partner Central APIs

다음 섹션에서는 APIs에 대한 일반적인 정보를 제공합니다.

주제

- [AWS Partner Central API 사용](#)
- [지원되는 AWS 리전](#)
- [권한](#)
- [AWS Partner Central API 할당량](#)
- [AWS Partner Central API에 대한 로깅](#)
- [AWS Partner Central API에 대한 알림](#)
- [샌드박스에서 테스트](#)

AWS Partner Central API 사용

다음 섹션에서는 AWS Partner Central APIs 사용에 대한 정보를 제공합니다.

주제

- [AWS파트너 중앙 계정 API 사용](#)
- [AWS Partner Central Selling API 사용](#)
- [AWS Partner Central Benefits API 사용](#)
- [AWS파트너 중앙 채널 API 사용](#)
- [수익 측정 API 사용](#)

AWS파트너 중앙 계정 API 사용

파트너 계정 관리

파트너 계정은 등록되고 기존 AWS계정과 연결되어 전체 IAM 기반 경험을 제공합니다. 이 접근 방식을 사용하면 사용자 수준 자격 증명이 필요하지 않으므로 AWS계정 내 IAM 엔터티를 통해 모든 등록 및 작업을 관리할 수 있습니다. AWS Partner Central은 AWS서비스와 원활하게 통합할 수 있으며, 이 모델은 파트너 엔터티 관리를 제공하고, 여러 카탈로그를 지원하고, AWS계정 수준 권한 부여 시스템을 설정합니다.

파트너는 사용 가능한 파트너 계정 APIs를 사용하여 계정을 등록, 관리 및 업데이트할 수 있습니다.

파트너 등록 API

파트너는 Partner Engagement 계정을 사용하여 AWS 계정을 등록할 수 있습니다. 파트너는 API 생성을 사용하여 필요한 제휴 리드 정보를 제공하고, APN 약관을 수락하고, 고유한 법적 이름을 제공합니다.

파트너 프로필 API

파트너는 퍼블릭/프라이빗 가시성 제어, 비동기 검증 및 상태 추적을 통해 회사 세부 정보(이름, 설명, 웹 사이트, 로고)가 포함된 프로필을 관리하여 한 번에 하나의 업데이트를 허용합니다.

도메인 관리 API

파트너는 이메일 검증을 통해 비즈니스 도메인을 등록하고 확인하여 조직 자격 증명을 설정하고 직원의 교육 및 인증을 연결할 수 있습니다.

파트너 연결 API

파트너는 다른 파트너와의 공동 판매, AWS 계정 간 AWS Marketplace 제안 데이터 공유, 배포자/채널 파트너에게 제품 배포/재판매 권한 부여 등 다양한 목적으로 자신의 AWS 계정을 다른 계정과 연결합니다.

파트너 확인 API

파트너 확인은 파트너 계정 등록을 위한 필수 사전 조건입니다. 파트너가 파트너 계정을 생성하려면 먼저 비즈니스 확인 및 자격 증명 확인 프로세스를 완료해야 합니다. 이러한 확인은 사업체가 합법적으로 등록되었는지 확인하고 AWS 계정을 등록하는 사람의 신원을 확인합니다.

주제

- [파트너 등록 작업](#)
- [파트너 프로필 작업](#)
- [도메인 관리 작업](#)
- [파트너 계정 연결 작업](#)
- [파트너 확인 작업](#)

파트너 등록 작업

파트너 등록

파트너는 Partner Engagement 계정을 사용하여 AWS 계정을 등록할 수 있습니다. 파트너는 API 생성을 사용하여 필요한 제휴 리드 정보를 제공하고, APN 약관을 수락하고, 고유한 법적 이름을 제공합니다.

등록 중

1. 동기식 파트너 엔터티 생성 - 고객은 입력 검증을 수행하고 동일한 요청 내에 파트너 엔터티를 생성하는 API 생성을 호출하여 파트너 등록 프로세스를 시작합니다.
2. 제휴 리드 정보 요구 사항 - 파트너 등록 중에 고객은 AWS 파트너 네트워크(APN) 관련 통신에 대한 제휴 리드 연락처 정보를 제공해야 합니다.
3. 이용 약관 적용 - 고객은 등록 중에 APN 이용 약관에 동의해야 합니다. 등록을 완료하고 모든 기능에 액세스하려면 수락이 필요합니다. 약관은 모든 APN 프로그램 혜택 및 기능에 적용됩니다.
4. Tag-On-Create 지원 - AWS 일관된 권한 부여 경험(CAE) 및 태그 기반 권한 부여의 일환으로 고객은 파트너 엔터티 생성 시 태그를 추가할 수 있습니다.
5. 법적 이름 기반 검증 - 등록은 파트너 법적 이름을 기반으로 고유성을 적용하므로 동일한 엔터티에서 중복 등록이 발생하지 않습니다.

등록 후

1. 태그 관리 지원 - 등록 후 파트너는 기존 파트너 엔터티에 태그를 지정, 태그 해제 및 나열할 수 있습니다.
2. 파트너 엔터티 가져오기 - 파트너 엔터티가 생성된 후 고객은 List API를 사용하여 파트너 ID를 검색한 다음 Get API를 사용하여 특정 파트너 엔터티의 세부 정보를 가져올 수 있습니다.

API 요약

1. CreatePartner API: 고객은 입력 검증을 수행하고 동일한 요청 내에 파트너 엔터티를 생성하는 CreatePartner API를 호출하여 파트너 등록 프로세스를 시작합니다.
2. ListPartners API: 등록 후 고객은 List API를 사용하여 파트너 엔터티 목록을 검색할 수 있습니다.
3. GetPartner API: 등록 후 고객은 Get API를 사용하여 특정 파트너 엔터티의 세부 정보를 검색할 수 있습니다.
4. PutAllianceLeadContact API: 프라이머리 제휴 리드 연락처 정보를 업데이트합니다. 이 작업을 통해 파트너는 조직의 연속성을 유지하면서 기본 제휴 리드 지정을 이전하거나 연락처 세부 정보를 수정할 수 있습니다.
5. GetAllianceLeadContact API: 기본 제휴 리드 연락처 정보를 업데이트합니다. 이 작업을 통해 파트너는 조직의 연속성을 유지하면서 기본 제휴 리드 지정을 이전하거나 연락처 세부 정보를 수정할 수 있습니다.

파트너 프로필 작업

프로필 관리

파트너는 공개/비공개 가시성 제어, 비동기 검증 및 상태 추적을 통해 회사 세부 정보(이름, 설명, 웹 사이트, 로고)가 포함된 프로필을 관리하여 한 번에 하나의 업데이트를 허용합니다.

1. **프로필 정보** - 파트너 프로필에는 표시 이름, 설명, 웹 사이트 URL, 로고, IndustrySegments 및 PrimarySolutionType과 같은 필수 회사 세부 정보가 포함되어야 합니다.
2. **다국어 지원** - 파트너 프로필은 여러 언어를 지원합니다. 프로필의 각 현지화된 버전에는 로캘에 대한 속성이 포함되어 있습니다.
3. **프로필 가시성 제어** - 파트너는 프로필 가시성(퍼블릭 또는 프라이빗)을 구성할 수 있습니다.
4. **프로필 스토리지** - 프로필 정보는 파트너 계정 객체 내에 저장됩니다.
5. **프로필 상태 관리** - 프로필 업데이트는 게시 전에 비동기적으로 검증됩니다. 파트너는 최신 프로필 업데이트 상태(IN_PROGRESS, SUCCEED, FAILED 및 CANCELLED)를 확인할 수 있습니다. 승인된 프로필만 공개적으로 볼 수 있습니다.
6. **동시성 제어 요청** - 한 번에 하나의 프로필 업데이트 요청만 허용됩니다. 프로필 업데이트가 이미 진행 중인 경우 고객은 새 업데이트를 시작하기 전에 CancelPartnerProfileUpdateTask API를 통해 진행 중인 업데이트를 명시적으로 취소해야 합니다.
7. **작업 업데이트 취소** - 파트너는 CancelPartnerProfileUpdateTask API를 호출하여 진행 중인 프로필 업데이트를 취소할 수 있습니다. 취소는 업데이트가 IN_PROGRESS 상태인 경우에만 허용되며 새 업데이트를 시작하기 전에 완료해야 합니다.

API 요약

1. **StartPutProfileTask API**: 이 API는 파트너 프로필 업데이트를 수락하고 기본 동기 입력 검증을 수행합니다.
2. **GetProfileUpdateTask API**: 이 API를 사용하면 파트너가 현재 프로필 업데이트 상태를 가져올 수 있습니다. 이는 고객이 StartPutPartnerProfile API를 통해 프로필 업데이트를 시작한 후 사용되므로 파트너는 요청이 IN_PROGRESS, FAILED, SUCCEEDED 또는 CANCELLED인지 확인할 수 있습니다.
3. **CancelProfileTask API**: 파트너가 현재 IN_PROGRESS 상태인 프로필 업데이트 작업을 명시적으로 취소할 수 있도록 허용합니다. 여기에는 자동 심사 단계와 수동 검토 단계가 모두 포함됩니다. 요청을 취소하면 파트너가 현재 심사 프로세스가 완료될 때까지 기다리지 않고 새 프로필 업데이트를 시작할 수 있습니다.

4. PutProfileVisibility API: 이 API를 호출하면 파트너는 로컬에 관계없이 프로필의 가시성을 제어할 수 있습니다. 이를 통해 파트너는 콘텐츠를 수정하지 않고도 프로필을 공개 또는 비공개로 만들 수 있습니다.
5. GetProfileVisibility API: 파트너는 UpdatePartnerProfileVisibility API를 호출하여 로컬에 관계없이 프로필의 가시성을 제어할 수 있습니다. 이를 통해 파트너는 콘텐츠를 수정하지 않고도 프로필을 공개 또는 비공개로 만들 수 있습니다.

도메인 관리 작업

도메인 관리

파트너는 이메일 검증을 통해 비즈니스 도메인을 등록하고 확인하여 조직 자격 증명을 설정하고 직원의 교육 및 인증을 연결할 수 있습니다.

API 요약

1. SendEmailVerificationCode API: 지정된 이메일 주소로 확인 코드를 전송하여 이메일 확인 프로세스를 시작합니다. 새 연락처 생성 및 이메일 주소 업데이트 모두에 사용됩니다.
2. AssociateAwsTrainingCertificationEmailDomain API: 이메일 도메인을 파트너 계정에 대한 AWS 교육 및 인증과 연결하여 직원 인증의 자동 확인을 활성화합니다.
3. DisassociateAwsTrainingCertificationEmailDomain API: 이메일 도메인과 파트너 계정에 대한 AWS 훈련 및 인증 간의 연결을 제거합니다.

파트너 계정 연결 작업

파트너는 파트너 연결 API를 AWS 사용하여 자신의 계정 간 연결 또는 다른 파트너의 AWS 계정과의 연결을 관리할 수 있습니다. 이 프로세스에는 다음 두 단계가 포함됩니다.

1. 연결 초대 단계: 연결 요청 시작 및 응답
2. 활성 연결 단계: 설정된 연결 및 관련 비즈니스 활동 관리

연결 초대 생성 및 전송

1단계: 연결 초대 생성

파트너 연결을 설정하는 첫 번째 단계는 CreateConnectionInvitation 작업을 사용하여 연결 초대를 생성하고 보내는 것입니다. 초대를 생성할 때 다음을 지정해야 합니다.

- 카탈로그: 연결을 관리할 카탈로그(AWS또는AWS샌드박스)
- ConnectionType: 설정하려는 관계의 유형입니다. 다음 중 하나를 선택합니다.
 - **OPPORTUNITY_COLLABORATION**: 기회 협업을 위한 기회 참여 초대장을 보낼 수 있도록 다른 파트너의 계정으로 연결 요청을 보내려는 경우이 유형을 사용합니다.
 - **SUBSIDIARY**: 소유한 여러AWS Marketplace 판매자 계정을 APN에 연결했거나 파트너로 등록된 "기본" 계정에 연결하려는 경우이 유형을 사용합니다.
- ReceiverIdentifier: 수신자의 퍼블릭 파트너 프로필 ID(기회 협업 연결 유형의 경우) 또는AWS계정 ID(보조 연결 유형의 경우)
- 이메일: 통신을 위한 연락처 이메일 주소
- 이름: 발신자의 이름
- 메시지: 연결 요청의 목적을 설명하는 설명

초대가 생성되면 보류 중 상태가 되어 수신자의 작업을 기다립니다.

Important

연결 초대를 보내면 초대를 수락하는 경우 수신자에AWS계정 ID를 공개하는 데 동의하는 것입니다. 이 공개는 계정 수준 연결을 설정하는 데 필요합니다.

2단계: 전송된 초대 모니터링

ParticipantType 파라미터가 로 설정된 ListConnectionInvitations 작업을 사용하여 전송한 초대 상태 추적할 수 있습니다SENDER. 다음 작업을 수행할 수 있습니다.

- 모든 발신 초대 보기
- 상태(PENDING, ACCEPTED, REJECTED, EXPIRED)를 기준으로 필터링
- 연결 유형별 필터링
- 다른 파트너에 대한 특정 초대 확인

3단계: 초대 취소(선택 사항)

보류 중인 초대를 수락하기 전에 취소해야 하는 경우 CancelConnectionInvitation 작업을 사용할 수 있습니다. 참고:

- 보류 중 상태의 초대만 취소할 수 있습니다.

- 초대가 수락되고 연결이 설정되면 취소할 수 없습니다.
- 설정된 연결을 종료하려면 `CancelConnection` 작업을 대신 사용해야 합니다.

연결 초대 수신 및 응답

1단계: 수신 초대 나열

받은 연결 초대를 보려면 `ParticipantType` 파라미터가 로 설정된 `ListConnectionInvitations` 작업을 사용합니다. 다음을 기준으로 필터링할 수 있습니다.

- 연결 유형
- 상태(PENDING, ACCEPTED, REJECTED)
- 발신자 식별자

2단계: 초대 세부 정보 검토

`GetConnectionInvitation` 작업을 사용하여 다음을 포함하여 특정 초대의 전체 세부 정보를 봅니다.

- 발신자 이름 및 연락처 이메일
- 목적을 설명하는 초대 메시지
- 요청 중인 연결 유형
- 만료 날짜
- 발신자의 공개 프로필 정보

3단계: 초대 수락 또는 거부

초대 세부 정보를 검토한 후 다음 두 가지 옵션을 사용할 수 있습니다.

옵션 A: 초대 수락

`AcceptConnectionInvitation` 작업을 사용하여 연결을 설정합니다. 수락하는 경우:

- 활성 상태에서 새 연결이 생성됩니다.
- 초대 상태가 수락됨으로 변경됩니다.
- AWS계정 ID가 발신자에게 공개됩니다.

- 연결 유형으로 활성화된 비즈니스 활동 수행을 시작할 수 있습니다.

Important

중요 동의: 연결 초대를 수락하면 발신자에게 AWS 계정 ID를 공개하는 데 동의하는 것입니다. 이 공개는 계정 수준 연결을 설정하는 데 필요합니다.

옵션 B: 초대 거부

연결을 설정하지 않으려면 `RejectConnectionInvitation` 작업을 사용합니다. 선택적으로 거부 이유를 제공할 수 있습니다. 거부된 경우:

- 초대 상태가 거부됨으로 변경됩니다.
- 연결이 설정되지 않음
- 초대가 보류 중인 초대 목록에 더 이상 표시되지 않습니다.

자동 만료

만료 기간 내에 조치를 취하지 않으면 시스템에서 자동으로 초대를 만료됨 상태로 전환합니다. 만료된 초대는 수락하거나 거부할 수 없습니다.

활성 연결 관리

연결 보기

연결이 설정되면 양 당사자는 다음 작업을 사용하여 연결을 보고 관리할 수 있습니다.

ListConnections: 필터링 옵션을 사용하여 연결 목록을 검색합니다.

- 연결 유형 및 상태를 기준으로 필터링(예: `OPPORTUNITY_COLLABORATION:ACTIVE`)
- 상대방의 식별자를 기준으로 필터링
- 각 연결에 대한 요약 정보 보기

GetConnection: 다음을 포함하여 특정 연결의 전체 세부 정보를 검색합니다.

- 이 파트너 관계와 연결된 모든 연결 유형
- 각 연결 유형의 상태(활성 또는 종료됨)

- 원래 초대의 연락처 정보
- AWS양 당사자의 계정 IDs 및 퍼블릭 프로필 IDs
- 각 연결 유형에 대한 생성 및 종료 타임스탬프

기존 연결에 연결 유형 추가

파트너와 이미 활성 연결이 있고 새 유형의 관계를 설정하려는 경우 다른 연결 유형의 새 연결 초대를 보낼 수 있습니다. 수락 시:

- 새 연결 유형이 기존 연결에 추가됩니다.
- 두 연결 유형 모두 독립적으로 관리할 수 있습니다.
- 관계에 대해 동일한 연결 ID가 유지됩니다.

Note

지정된 카탈로그의 두 계정 간에는 언제든지 특정 유형의 활성 연결이 하나만 있을 수 있습니다.

연결 유형 종료

어느 당사자든 `CancelConnection` 작업을 사용하여 특정 연결 유형을 종료할 수 있습니다. 종료 시:

- 종료할 연결 ID 및 연결 유형 지정
- 종료 이유 제공
- 연결 유형의 상태가 `CANCELED`로 변경됩니다.
- 동일한 연결 내의 다른 연결 유형은 영향을 받지 않습니다.

연결 종료 완료

연결 내의 모든 연결 유형이 종료되면 연결 자체가 종료됨 상태로 전환됩니다. 완전히 종료된 연결:

- 다시 활성화할 수 없지만 새 연결 요청을 보내 해당 계정으로 비즈니스를 계속할 수 있습니다.
- 기록 보관을 위해 시스템에 남아 있습니다.
- `GetConnection` API를 사용하여 볼 수 있음

- 관계를 다시 설정하려면 새 초대가 필요합니다.

연결 이벤트 모니터링

파트너는 Amazon EventBridge를 사용하여 연결 관련 이벤트를 모니터링하여 다음 정보를 파악해야 합니다.

- 새로운 수신 연결 초대
- 보낸 초대의 상태 변경(ACCEPTED, REJECTED, EXPIRED)
- 상대방이 시작한 연결 종료

이벤트를 수신하면 적절한 API 가져오기 작업을 사용하여 최신 세부 정보를 가져옵니다.

- GetConnectionInvitation 초대 업데이트
- GetConnection 연결 업데이트용

가능한 이벤트

- 연결 초대 생성됨: 수신되는 연결 초대를 수신 계정에 알립니다.
- 연결 초대 취소됨: 발신자가 수신 연결 초대를 취소할 때 수신자 계정에 알립니다.
- 연결 초대 만료: 작업 없이 연결 초대가 만료되면 발신자와 수신자 계정 모두에 알립니다.
- 연결 초대가 거부됨: 수신자가 연결 초대를 거부하면 발신자 계정에 알립니다.
- 연결 초대 수락: 연결 초대가 수락되고 연결이 설정되면 발신자 계정에 알립니다.
- 연결 취소됨: 모든 연결이 완전히 종료되면 연결된 당사자 모두에게 알립니다.

연결 상태 이해

연결 초대 상태

State	설명	로 전환할 수 있음
PENDING	생성 시 초기 상태, 수신기 작업 대기 중	ACCEPTED, REJECTED, EXPIRED, CANCELED
ACCEPTED	수신기 수락됨, 연결이 생성됨	없음(터미널 상태)

State	설명	로 전환할 수 있음
REJECTED	수신자 거부됨, 선택적 거부 이유 포함	없음(터미널 상태)
EXPIRED	만료 기간 내에 작업이 없어 시스템 만료	없음(터미널 상태)
CANCELED	발신자가 수락 전에 초대를 철회함	없음(터미널 상태)

연결 유형 상태

State	설명
ACCEPTED	연결 유형이 작동 중이고 연결된 비즈니스 활동이 활성화됨
CANCELED	한 당사자가 종료한 연결 유형

모범 사례

1. 명확한 커뮤니케이션: 연결 초대장에 목적과 예상되는 협업을 설명하는 상세하고 명확한 메시지를 제공합니다.
2. 적시 응답: 자동 만료를 방지하기 위해 연결 초대를 즉시 모니터링하고 응답합니다.
3. 정기 검토: 활성 연결을 정기적으로 검토하여 현재 비즈니스 요구 사항과 관련이 있는지 확인합니다.
4. 적절한 종료: 비즈니스 관계를 종료할 때 CancelConnection 작업을 명확한 이유와 함께 사용하여 전문적인 커뮤니케이션을 유지합니다.
5. 이벤트 모니터링: 연결 상태 변경에 대한 알림을 자동으로 수신하도록 EventBridge 규칙을 설정하여 중요한 업데이트에 대한 알림을 받을 수 있습니다.
6. 연결 유형 관리: 각 유형은 서로 다른 비즈니스 기능을 지원하므로 연결을 설정하기 전에 필요한 연결 유형을 고려합니다.
7. 보안 인식: 연결을 설정하면 당사자 간의 AWS 계정 IDs 드러납니다. 신뢰할 수 있는 파트너와만 연결합니다.

연결 유형 및 용도

연결 유형	용도	사용 사례
OPPORTUNITY_COLLABORATION	다른 파트너와의 공동 판매 기회에 대한 협업을 활성화합니다.	다른 파트너가 기회에 액세스하거나 그 반대로 액세스하도록 하려면이 연결 유형을 사용합니다. 이 연결을 통해 연결된 파트너에게 기회 참여 초대를 보낼 수 있습니다.
SUBSIDIARY	여러AWS Marketplace 판매자 계정과 기본 파트너 계정 간의 연결을 설정합니다.	파트너 및 판매자로 등록된 기본 계정에 연결하려는AWS Marketplace 판매자 계정이 여러 개 있는 경우를 사용합니다.

파트너 확인 작업

파트너 확인 관리

파트너 확인은 파트너 계정 등록을 위한 필수 사전 조건입니다. 파트너가 파트너 계정을 생성하려면 먼저 비즈니스 확인 및 자격 증명 확인 프로세스를 완료해야 합니다. 이러한 확인은 사업체가 합법적으로 등록되었는지 확인하고AWS계정을 등록하는 사람의 신원을 확인합니다.

파트너는 사용 가능한 확인 APIs를 활용하여 확인 프로세스를 시작하고 모니터링할 수 있습니다.

확인 중

1. 비즈니스 확인 시작 - 파트너는 법적 이름, 등록 ID, 국가 코드 및 법인 설립 관할 구역을 포함한 비즈니스 등록 세부 정보를 사용하여 StartVerification API를 호출하여 비즈니스 확인을 시작합니다.
2. 자격 증명 확인 시작 - 파트너는 등록자 정보로 StartVerification API를 호출하여 자격 증명 확인을 시작합니다. API는 등록자가 정부에서 발급한 ID를 사용하여 자격 증명 확인 워크플로를 완료하는 보안 완료 URL을 반환합니다.
3. 확인 상태 모니터링 - 파트너는 GetVerification API를 사용하여 확인 프로세스의 현재 상태 및 세부 정보를 검색합니다. API는 확인 유형, 상태, 타임스탬프 및 확인별 세부 정보를 반환합니다.
4. 확인 완료 - 파트너가 파트너 계정 생성을 진행하기 전에 비즈니스 확인 및 자격 증명 확인 모두 SUCCEEDED 상태에 도달해야 합니다. 계정 등록 전에 확인 실패 또는 만료를 해결해야 합니다.

사후 확인

1. 파트너 계정 생성 - 확인에 성공하면 파트너는 파트너 계정 설명서에 설명된 CreatePartner API를 사용하여 파트너 계정을 생성합니다.
2. 확인 레코드 검색 - 파트너는 시작 중에 반환된 확인 ID가 있는 GetVerification API를 사용하여 언제든지 확인 세부 정보를 검색할 수 있습니다.

API 요약

1. **StartVerification** API: 파트너는 StartVerification API를 호출하여 확인 프로세스를 시작합니다. 비즈니스 확인을 위해 API는 비즈니스 등록 세부 정보를 검증합니다. 자격 증명 확인을 위해 API는 등록자가 자격 증명 확인을 완료할 수 있도록 시간 제한 완료 URL을 생성합니다.
2. **GetVerification** API: 확인을 시작한 후 파트너는 GetVerification API를 사용하여 확인 상태 및 세부 정보를 검색합니다. API는 현재 상태, 타임스탬프 및 확인 관련 정보를 반환합니다.

AWS Partner Central Selling API 사용

이 AWS Partner Central API 참조는 [AWS파트너가](#) 고객 관계 관리(CRM) 시스템을 Partner Central과 통합하는 데 도움이 되도록 설계되었습니다. API는 Partner Central과의 상호 작용을 자동화하므로 공동 비즈니스 활동에 효과적으로 참여할 수 있습니다.

API는 표준 AWS API 기능을 제공합니다. API [작업을](#) 사용하거나 프로그래밍 언어 또는 플랫폼에 맞는 AWS SDK를 사용하여 액세스합니다. 자세한 내용은 [시작하기AWS](#) 및 [빌드할 도구를 참조하세요 AWS](#).

AWS Partner Central API에서 제공하는 기능

1. 기회 관리: , CreateOpportunity, UpdateOpportunity, ListOpportunitiesGetOpportunity, 등의 API 작업을 AWS 사용하여와 공동 판매 기회를 쉽게 관리할 수 있습니다 AssignOpportunity.
2. AWS추천 관리: ListEngagementInvitations, GetEngagementInvitationStartEngagementByAcceptingInvitation, 및 등의 작업을 AWS 사용하여 공유된 추천을 쉽게 수신할 수 있습니다 RejectEngagementInvitation.
3. 엔터티 연결: AWS제품, 파트너 솔루션, AWS Marketplace솔루션, AWS Marketplace제품 및 AWS Marketplace비공개 제안과 같은 관련 엔터티를 AssociateOpportunity 작업 및를 사용하여 기회와 연결합니다 DisassociateOpportunity.

4. AWS 기회 세부 정보 보기: GetAWSOpportunitySummary 작업을 사용하여 기회와 연결된 기회에 대한 AWS 실시간 요약 검색합니다.
5. 솔루션 나열: 파트너를 사용하여 제공하는 솔루션을 나열하기 위한 목록 APIs를 제공합니다 ListSolutions.
6. 이벤트 구독: 파트너는 기회 생성됨, 기회 업데이트됨, 참여 초대 수락됨, 참여 초대 거부됨, Amazon EventBridge를 사용하여 참여 초대 생성됨과 같은 이벤트를 수신하여 기회에 대한 실시간 업데이트를 구독할 수 있습니다.

지원되는 리전 및 엔드포인트

AWS Partner Central API는 미국 동부(버지니아 북부) 리전에서 사용할 수 있습니다.

리전	엔드포인트
us-east-1	partnercentral-selling.us-east-1.api.aws

파트너는 안전한 샌드박스 환경에서 API 통합을 테스트하고 검증할 수 있습니다. 이를 통해 라이브 데이터에 영향을 주지 않고 API 작업을 테스트할 수 있습니다. 자세한 내용은 [AWS Partner Central Selling API용 샌드박스에서 테스트](#) 단원을 참조하십시오.

API 엔터티 판매

AWS Partner Central 엔터티는 파트너와 간의 공동 판매 계약에 사용되는 주요 비즈니스 구성 요소를 나타냅니다. 이러한 엔터티는 기회, 솔루션, 제품 및 제안과 관련된 정보를 캡슐화하여 공동 판매 활동의 원활한 협업 및 관리를 가능하게 합니다. 다음 섹션에서는 Partner Central Selling API의 코어 엔터티에 대한 설명을 제공합니다.

Opportunity

기회는 기업이 식별하고 적극적으로 추구하는 잠재적 판매 또는 거래입니다. 자격을 갖춘 잠재 고객이거나 회사의 제품 또는 서비스가 해결할 수 있는 특정 요구 사항이 있는 잠재 고객입니다. 기회는 일반적으로 영업 파이프라인 또는 CRM 시스템에서 추적되며 향후 수익의 기반을 형성합니다. 효과적인 기회 관리에는 초기 검증부터 종료까지 영업 프로세스를 통해 리드를 양성하는 것이 포함됩니다. 자세한 내용은 기회 및 [데이터 유형 작업을 참조하세요](#).

파트너 솔루션

파트너가 생성하고 제공하는 소프트웨어 제품 또는 컨설팅 관행인 AWS 파트너 솔루션(AWS Partner Central에서는 제안이라고 함)을 나타냅니다. 파트너 솔루션은 고객이 특정 비즈니스 문제를 해결하거나 AWS 서비스를 사용하여 특정 목표를 달성할 수 있도록 지원합니다. 자세한 내용은 [솔루션이란 무엇입니까?](#)를 참조하세요.

AWS 제품

특정 AWS 서비스 또는 제품을 나타냅니다. 확장 가능하고 안정적이며 비용 효율적인 인프라 솔루션을 제공하도록 설계된 다양한 제품과 서비스를 AWS 제공합니다. 파트너는 [Partner Central의 대량 가져오기 페이지](#)(가져오기 시작 > AWS 제품 및 솔루션)에서 최신 AWS 제품 목록을 얻을 수 있습니다. 자세한 내용은 [AWS 제품에 대해 알아보기](#)거나 [모든 AWS 제품 목록을 참조하십시오](#).

AWS Marketplace 비공개 제안

AWS Marketplace 비공개 제안은 AWS Marketplace 판매자가 개별 AWS 고객에게 특정 요금 및 조건을 제공할 수 있는 기능입니다. 판매자는 비공개 제안을 통해 사용자 지정 가격, 결제 일정 및 최종 사용자 라이선스 조건을 협상할 수 있습니다. AWS 고객은 특정 요구 사항을 충족하는 소프트웨어 솔루션을 얻는 동시에 표준 제품에 비해 더 유리한 조건 또는 요금의 혜택을 누릴 수 있습니다. 비공개 제안 프로세스에는 판매자가 맞춤형 약관으로 고유한 제안을 생성한 다음 지정된 AWS 고객과 비공개로 공유하여 검토 및 수락합니다. 자세한 내용은 [의 비공개 제안을 AWS Marketplace](#) 참조하세요.

참여 초대

참여 초대는 파트너가 특정 추천에 대해 협업하도록 AWS 하는의 공식 요청을 말합니다. 이렇게 하면 AWS와 파트너가 협력하여 기회를 추진할 수 있습니다. 파트너는 초대를 수락하거나 거부할 수 있습니다.

주제

- [기회 작업](#)
- [의 기회 작업 AWS](#)
- [기회 업데이트 작업](#)
- [멀티파트너 기회 작업](#)
- [기회 연결, 연결 해제 및 할당](#)
- [리드 작업](#)
- [거래 규모 조정 인사이트 활용](#)

- [모범 사례](#)

기회 작업

기회란 무엇입니까?

영업 프로세스의 초기 단계에서 영업 담당자는 관심 있는 개인(리드라고 함)이 고객이 될 가능성이 있는지 평가합니다. 이 평가 및 검증 단계를 검증이라고 합니다. 리드가 적극적으로 간주되고 고객으로 전환할 가능성이 더 높은 것으로 간주되면 기회로 분류됩니다.

기회 작업

파트너는 AWS Partner Central Selling API를 사용하여 CRM 시스템 내에서 생성된 기회를 관리하고 AWS Partner Central과 동기화할 수 있습니다. 이를 통해 파트너는 시작부터 종료까지 기회를 추적하고 관리할 수 있습니다.

기회 생성

기회 관리의 첫 번째 단계는 CreateOpportunity 작업을 사용하여 기회를 만드는 것입니다. 이렇게 하면 Lifecycle.ReviewStatus 설정된 기회가 생성됩니다 Pending Submission. 그러나 검증을 AWS 위해 아직 기회가 제출되지 않았습니다.

기회가 생성된 후에는 AssociateOpportunity 작업을 사용하여 하나 이상의 파트너 솔루션, AWS Marketplace 솔루션 또는 AWS Marketplace 제품을 기회와 연결해야 합니다. 이 작업은 기회가 판매하려는 것을 명확하게 정의합니다. ListSolutions API를 사용하여 계정에서 사용 가능한 솔루션의 전체 목록을 보고 솔루션 1~10개를 연결할 수 있습니다.

선택적으로 AssociateOpportunity 작업을 사용하여 관련 AWS 제품을 기회와 연결할 수도 있습니다. 이 단계는 AWS 영업 팀이 기회와 함께 판매될 것으로 예상되는 AWS 제품을 이해하는 데 도움이 됩니다.

필요한 솔루션 또는 제품이 연결되고 선택적으로 AWS 제품이 연결되면

StartEngagementFromOpportunityTask 작업을 사용하여 기회에 대한 참여를 시작할 수 있습니다. 이는 참여를 시작할 기회를 제출할 때입니다.

검토 프로세스

StartEngagementFromOpportunityTask 작업을 사용하여 기회에 대한 참여를 시작한 후 기회는 AWS 검증 단계에 들어가고 Lifecycle.ReviewStatus는 Submitted로 설정됩니다. 검토 프로세스가 완료될 때까지 기회를 변경할 수 없습니다.

이 검증 단계에서는 기회 세부 정보가 정확하고 완전한지 AWS 확인합니다. 검증이 진행되는 동안 Lifecycle.ReviewStatus는 In-Review로 설정됩니다.

파트너의 변경 사항이나 추가 세부 정보가 필요한 경우 AWS는 Lifecycle.ReviewStatus로 설정하고 Action Required필요한 업데이트는 Lifecycle.ReviewComments 필드를 통해 전달됩니다.

기회가 검증을 통과하면 Lifecycle.ReviewStatus 변경 Approved되어 공동 판매 활동을 위한 기회가 마련됩니다.

파트너는 Amazon EventBridge를 사용하여 Opportunity Updated 이벤트를 모니터링해야 합니다. 이렇게 하면 상태 변경 또는 피드백을 알립니다 AWS. 이벤트를 수신하면 파트너는 GetOpportunity API 작업을 사용하여 최신 기회 세부 정보를 가져오고 Lifecycle.ReviewStatus 필드를 확인할 수 있습니다.

검증 문제 해결

Lifecycle.ReviewStatus가 In-Review로 설정된 경우 Action Required파트너는 로 강조 표시된 문제를 해결해야 합니다 AWS. 이를 해결하기 위해 파트너는 UpdateOpportunity API 작업을 사용하여 기회를 업데이트할 수 있습니다.

Action Required 상태 중에는 특정 필드만 편집할 수 있습니다. 이러한 필드는 다음과 같습니다.

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

필요한 사항을 변경한 후 기회가 검증 단계에 다시 들어가고 기회의이 Approved 또는 In-Review로 설정될 때까지 프로세스가 반복 Lifecycle.ReviewStatus됩니다 Disqualified.

승인 후 업데이트

기회가가 되면 Approved파트너는 UpdateOpportunity 작업을 사용하여 필요에 따라 기회를 계속 업데이트하여 원활한 공동 판매 활동을 촉진할 수 있습니다.

파트너는 Amazon EventBridge를 통해 Opportunity Updated 이벤트를 계속 모니터링하여 변경 사항에 대한 업데이트를 유지해야 합니다.AWS업데이트 추적에 대한 자세한 내용은 "기회 업데이트 작업" 섹션을 참조하세요. 파트너는 비즈니스 검증 규칙에 따라 선택 필드를 업데이트할 수도 있습니다.

의 기회 작업AWS

1.AWS기회 받기

AWS영업 임원이 파트너를 CRM 시스템의 기회에 연결하면 파트너와 기회가 공유AWS됩니다. 이를 기회라고 하며AWS, 파트너의 자체 계정에서 생성된 기회와는 다릅니다.

AWS기회에 파트너가 연결되면AWS기회의 데이터 하위 집합이 포함된 참여 초대AWS생성합니다. 파트너는 참여 초대 생성 이벤트를 받게 됩니다.

2. 참여 초대 검토

참여 초대에는 Project.Title, , Project.CustomerUseCase, 및 몇 가지 추가 필드Lifecycle.StageProject.CustomerBusinessProblem와 같은 필수 정보가 포함되어 있습니다. 파트너는이 데이터를 사용하여 기회를 추구할지 여부를 결정할 수 있습니다.

그러나AWS기회의 다음 필드는 참여 초대에 포함되지 않습니다.

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

3. 참여 초대 처리

참여 초대 생성 이벤트를 수신하면 파트너는 GetEngagementInvitation 작업을 사용하여 기회의 세부 정보를 검색할 수 있습니다AWS.

파트너가 기회를 추구하지 않기로 결정한 경우 필요한와 함께 RejectEngagementInvitation 작업을 사용하여 초대를 거부할 수 있습니다RejectionReason. 거부되면 기회에 대한 액세스 권한이 손실됩니다.

초대를 수락하고 진행하려면 파트너가 `StartEngagementByAcceptingInvitationTask` 작업을 사용해야 합니다. 이는 다음 작업을 순차적으로 수행하는 비동기 작업입니다.

1. 참여 초대를 수락합니다.
2. 기회의 데이터를 AWS 사용하여 파트너 계정에 새 기회를 생성합니다.
3. 파트너 계정에서 고객을 식별하는 데 필요한 추가 세부 정보를 포함합니다.

완료되면 기회 생성 이벤트가 해당 기회 ID와 함께 트리거됩니다. 그런 다음 파트너는 `GetOpportunity` 작업에서 ID를 사용하여 기회에 대한 전체 세부 정보를 검색할 수 있습니다.

파트너가 이벤트를 사용하지 않는 경우 동일한 페이로드를 `StartEngagementByAcceptingInvitationTask` 다시 호출하여 최신 상태를 확인할 수 있습니다.

4. 수락 후 AWS 기회 관리

기회가 파트너의 계정에 있으면 시스템의 다른 기회처럼 관리하고 업데이트할 수 있습니다. 파트너는 와 같은 작업을 사용하여 필요한 사항을 변경할 수 `UpdateOpportunity` 있습니다.

AWS 업데이트를 추적하고 기회 업데이트를 관리하는 방법에 대한 자세한 내용은 [기회 업데이트 작업](#) 섹션을 참조하세요.

기회 업데이트 작업

기회 업데이트

파트너는 `UpdateOpportunity` 작업을 사용하여 업데이트되는 필드와 업데이트 시기를 제어하는 특정 규칙을 사용하여 기회를 업데이트할 수 있습니다.

1. `Lifecycle.ReviewStatus`가 `Submitted` 또는 `In-Review`인 경우 업데이트할 수 없습니다.
2. 제출 전에 파트너는 업데이트를 수행할 수 있지만 `StartEngagementFromOpportunityTask` 작업이 호출되지 않는 한 이를 처리하지 않습니다.
3. 기회가 `Submitted` 또는 `In-review` 상태이면 모든 업데이트가 차단됩니다.
4. 기회가 `Action Required` 상태인 경우 업데이트할 필드를 선택합니다. 이러한 필드는 다음과 같습니다.
 - `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. 검토 프로세스 후(예: Lifecycle.ReviewStatus가 로 설정된 경우Approved) 다음 필드는 업데이트할 수 없습니다.
- Customer.Account.Address.Country
 - Customer.Account.Address.PostalCode
 - Customer.Account.Industry
 - Customer.Account.WebsiteUrl
 - Project.CustomerBusinessProblem
 - PartnerOpportunityIdentifier
 - Project.Title
6. 다른 모든 필드의 경우 UpdateOpportunity 작업을 사용하여 업데이트할 수 있습니다. 그러나 특정 프로그램 또는 기회 유형에 대한 비즈니스 규칙에 따라 추가 제한이 적용될 수 있습니다. 자세한 내용은 필드 수준 검증을 참조하세요.
7. UI와 API를 통해 이루어진 모든 업데이트에 대해 기회 업데이트 이벤트가 생성됩니다.

에서 기회에AWS대한 업데이트 수신

AWS는 일반적으로AWS기회를 업데이트하며, 업데이트가 이루어질 때마다 기회 업데이트 이벤트가 생성됩니다.

에서 최신 업데이트를 검색하려면AWS파트너는 두 개의 개별 작업을 호출해야 합니다.

1. GetOpportunity 파트너의 기회에 대한 세부 정보를 검색합니다.

2. GetAWSOpportunitySummaryAWS기회 데이터의 실시간 요약을 검색합니다.

의 대부분의 정기 업데이트AWS는 GetAWSOpportunitySummary 응답을 통해 사용할 수 있습니다. 그러나 파트너의 기회에서 속성을 직접 업데이트할AWS수 있습니다.

의 업데이트를 사용하려면AWS

1. GetAWSOpportunitySummary 작업을 호출하여AWS기회의 최신 세부 정보를 검색합니다.
2. 변경 사항을 파트너의 기회에 반영해야 하는 경우 UpdateOpportunity 작업을 사용하여 관련 데이터를 파트너의 기회에 복사합니다.

파트너는이 프로세스를 직접 업데이트 메커니즘으로 자동화하거나 수동 검토 프로세스를 구현하여 데이터를 검증하고 업데이트할 수 있습니다.

멀티파트너 기회 작업

AWS Partner는 활성 참가자AWS로서와 함께 기회에 대해 함께 작업할 수 있습니다.

주제

- [참여 및 스냅샷](#)
- [에 대한 사용자 지정 정책 생성 ResourceSnapshotJobRole](#)
- [파트너를 기회에 초대](#)
- [참여 초대 세부 정보 검색](#)
- [참여 초대에 응답](#)
- [멀티파트너 기회 검토 및 업데이트](#)
- [스냅샷을 사용하여 파트너 업데이트 수신](#)
- [참여 멤버 보기](#)
- [리소스 스냅샷 작업 상태 모니터링](#)

참여 및 스냅샷

참여는 여러AWS Partner계정 간의 협업과 고객 기회를AWS위해AWS가 소유한 리소스입니다. 참여는 개별 기회 소유권과 제어를 유지하면서 파트너와 협업 간의 안전한 정보 공유를 촉진합니다. 참여에는 활성 참가자AWS로 사용하여AWS및 파트너 모두에서 발생하는 기회가 포함됩니다. 파트너는AWS

또는 다른 파트너를 사용하여 참여에 초대할 수 있습니다. Engagement Invitations. 초대된 파트너가 수락하면 동일한 참여 내에서 자신의 기회를 얻게 됩니다.

참여 구성원은 특정 시점, 기회와 같은 기본 리소스의 변경 불가능한 특정 필드 사본 등 스냅샷을 통해 진행 상황을 공유합니다. 스냅샷은 참여 내에서 생성되어 멤버와 공유됩니다. 기본 리소스가 변경되면 소유자는 업데이트를 반영하는 새 스냅샷 개정을 생성할 수 있습니다. 리소스 변경 시 새 개정을 자동으로 생성하는 고객 소유 작업인 스냅샷 작업을 AWS가 제공합니다. 공유된 스냅샷은 참여 구성원이 영구적으로 액세스할 수 있습니다.

에 대한 사용자 지정 정책 생성 ResourceSnapshotJobRole

다중 파트너 기회를 공동 작업하려면 참여의 다른 파트너와 기회의 스냅샷을 공유해야 합니다. 최신 기회 세부 정보에 대한 액세스를 유지하려면 라는 사용자 지정 역할을 생성해야 합니다 ResourceSnapshotJobRole. 이 역할을 사용하면 시스템에서 사용자를 대신하여 기회의 스냅샷을 생성하고 동일한 참여의 다른 파트너로부터 스냅샷을 검색할 수 있습니다. 이 역할이 없으면 기회에 대한 업데이트는 참여의 다른 파트너에게 표시되지 않으며 기회 데이터의 오래된 스냅샷을 계속 볼 수 있습니다.

Note

이외의 이름을 사용할 수 있습니다 ResourceSnapshotJobRole. 다른 이름을 사용하는 경우 다음 정책에 ResourceSnapshotJobRole 있는 모든 인스턴스를 정책 이름으로 바꿉니다.

이 역할을 생성하려면 IAM 콘솔로 이동하여 다음 사용자 지정 신뢰 정책을 사용하여 ResourceSnapshotJobRole 역할을 생성합니다.

JSON

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

```
    ]
  }
}
```

이 역할을 생성한 후에는 [AWSPartnerCentralSellingResourceSnapshotJobExecutionRolePolicy](#) AWS 관리형 정책을 연결하거나 다음 권한 정책을 연결해야 합니다.

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/*"
      ]
    }
  ]
}
```

{account}를 AWS 계정 ID로 바꿉니다.

ResourceSnapshotJobRole 역할을 생성하고 권한을 연결한 후에는 조직의 ResourceSnapshotJobRole를 설정해야 합니다. [PutSellingSystemSettings](#) 작업을 사용하여 (AWS 계정에서 식별한) ResourceSnapshotJobRole 회사의 로 생성한 역할을 설정합니다.

```
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}를AWS계정 IDResourceSnapshotJobRole로 바꾸고를 생성한 역할의 이름으로 바꿉니다. 리소스 스냅샷 작업은이 역할을 암시적으로 수임하여 기회에 액세스하고 참여의 다른 파트너와 공유할 스냅샷을 생성합니다.

파트너를 기회에 초대

파트너는 참여 초대를 시작하여 두 파트너AWS모두에서 시작된 기회에 대해 협업할 수 있습니다. 최대 9명의 파트너를 기회에 초대할 수 있습니다.

파트너를 기회에 초대하려면

1. AWS파트너 중앙 판매 가이드의 [파트너 찾기 및 연결](#)에 있는 단계를 따릅니다. 파트너를 기회에 초대하려면 먼저 파트너와 연결해야 합니다. 이 연결은 서로의 계정 IDs에 대한 상호 액세스 권한을 부여하여 초대가 의도한 파트너 계정에 도달하도록 합니다.
2. [StartEngagementFromOpportunityTask](#) 작업을 사용하여 참여를 생성하고 기회를 연결합니다. 이 비동기 작업은 다음 작업을 순차적으로 수행합니다.
 1. 참여를 생성합니다.
 2. 기회를 참여와 연결합니다.
 3. 작업을 [CreateEngagementInvitation](#)에 호출합니다AWS.
 4. 기회를 제출합니다AWS.
 5. ResourceSnapshotJob를 시작하여 스냅샷을 생성합니다.
3. 다른 파트너를 초대하여 가입시킵니다.
 - a. 기회와 연결된 참여 ID를 가져오려면 이전 단계에서 TaskIdentifier 반환된 로 필터링된 [ListEngagementFromOpportunityTasks](#) 작업을 사용합니다.
 - b. 수신 파트너의 참여 ID와 계정 ID를 사용하여 로 초대를 보냅니다CreateEngagementInvitation.

초대가 성공하면 수신 파트너에게 참여 초대 생성 이벤트가 전송됩니다.

참여 초대 세부 정보 검색

참여 초대 생성 이벤트를 받으면 [GetEngagementInvitation](#) 작업을 사용하여 초대 세부 정보를 검색합니다. 응답에는 참여와 관련된 고객 기회에 대한 필수 정보가 포함됩니다.

초대 세부 정보, 특히 `InvitationMessage`, `Project.TitleProject.CustomerUseCase`, 및 `Project.CustomerBusinessProblem` 필드를 검토합니다. 이 정보는 고객 기회와 공동 작업에 대한 매력적인 파트너의 기대치에 대한 컨텍스트를 제공합니다.

이러한 세부 정보를 사용하여 참여 초대를 수락하거나 거부하여 기회를 모색할지 여부를 평가합니다.

참여 초대에 응답

파트너가 참여 초대를 보내면 초대를 알리는 참여 초대 생성 이벤트가 생성됩니다. 초대를 수락하거나 거부할 수 있습니다. 이 결정은 멀티파트너 기회에 대한 귀하의 참여를 결정합니다.

참여 초대를 거부하려면:

[RejectEngagementInvitation](#) 작업을 사용하여 초대를 거부합니다. 결정을 설명하는 `RejectionReason` 파라미터를 제공해야 합니다. 거부되면 초대 세부 정보에 액세스할 수 없으며 초대 거부 이벤트가 전송 파트너에게 초대를 거부했음을 알립니다.

참여 초대를 수락하려면:

초대를 수락하고 멀티파트너 기회를 진행하려면 [StartEngagementByAcceptingInvitationTask](#) 작업을 사용합니다. 이 비동기 작업은 다음 작업을 순차적으로 수행합니다.

1. 참여 초대를 수락합니다.
2. 전송 파트너의 기회에 대한 데이터를 사용하여 파트너 계정에 새 기회를 생성합니다. 기회 생성 이벤트를 받게 됩니다.
3. 계정에서 고객을 식별하는 데 필요한 추가 세부 정보를 포함합니다.
4. 귀하가 초대를 수락했고 참여에 추가되었음을 파트너에게 알리는 참여 초대 수락 이벤트를 게시합니다.

만료된 참여 초대

15일 이내에 참여 초대에 응답하지 않으면 초대가 만료되며 멀티파트너 기회에 참여할 수 없습니다. 참여 초대 만료 이벤트는 전송 파트너에게 초대가 만료되었음을 알립니다.

 Note

그래도 협업하려면 전송 파트너가 새 참여 초대를 시작해야 합니다.

멀티파트너 기회 검토 및 업데이트

파트너가 기회에 대한 참여를 시작하면 수신 파트너의 계정에서 새로 생성된 기회의 검토 상태는 전송 파트너의 기회 상태에 따라 결정됩니다.

제출된 검토 상태의 기회:

전송 파트너의 기회가 로 Lifecycle.ReviewStatus 설정된 경우 Submitted다음 프로세스가 발생합니다.

1. 수신 파트너의 계정에 생성된 새 기회는 로 Lifecycle.ReviewStatus 설정됩니다Submitted.
2. Submitted 상태는 전송 파트너의 기회가 될 때까지 유지됩니다Approved.
3. 전송 파트너의 기회가 검증되고 Lifecycle.ReviewStatus가 로 설정되면 Approved다른 파트너의 기회가 자동으로 Approved 상태를 상속합니다.

이 프로세스는 멀티파트너 기회 전반에 걸쳐 일관된 기회 세부 정보를 보장하므로 여러 검토가 필요하지 않습니다.

완료되면 기회 생성 이벤트가 해당 기회 ID와 함께 트리거됩니다. 파트너는이 ID를 GetOpportunity 작업과 함께 사용하여 전체 기회 세부 정보를 검색할 수 있습니다.

검토 상태가 승인된 기회:

전송 파트너가 로 Lifecycle.ReviewStatus 설정된 기회에 대한 참여를 시작하는 경우 Approved다음 프로세스가 발생합니다.

1. 수신 파트너의 계정에 생성된 새 기회는 로 Lifecycle.ReviewStatus 설정됩니다Approved.
2. 기회 생성 이벤트는 해당 기회 ID와 함께 트리거됩니다.
3. 파트너는 기회 ID와 함께 [GetOpportunity](#) 작업을 사용하여 전체 기회 세부 정보를 검색할 수 있습니다.

그러나 승인된 기회에는 전송 파트너의 기회에서 복사되지 않은 일부 파트너별 세부 정보가 없을 수 있습니다. 수신 파트너는 기회 단계가 Qualified 또는 이후 단계로 업데이트될 때 [UpdateOpportunity](#) 작업을 사용하여 이러한 세부 정보를 업데이트해야 합니다.

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

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

수신 파트너의 계정에서 기회가 생성되면 파트너의 시스템 내에서 다른 기회처럼 관리하고 업데이트할 수 있습니다. 파트너는 [UpdateOpportunity](#) 작업을 사용하여 변경하거나 기회 참여에 대한 자세한 정보를 제공할 수 있습니다.

스냅샷을 사용하여 파트너 업데이트 수신

참여 내에서 파트너는 기회를 독립적으로 유지하고 업데이트합니다. 누군가 기회 추가와 같은 참여 리소스를 수정하면 참여 리소스 스냅샷 생성 이벤트가 게시됩니다.

변경 사항에 대한 최신 정보를 확인하고 다른 파트너의 기회에 대한 최신 정보에 액세스하려면 다음 작업을 사용할 수 있습니다.

- [ListEngagementResourceAssociations](#) -이 작업을 사용하여 기회와 연결된 참여 ID를 검색합니다.
- [ListResourceSnapshots](#) -이 작업을 사용하여 참여와 관련된 모든 기회 스냅샷의 포괄적인 목록을 검색하여 모든 파트너에서 사용 가능한 스냅샷의 개요를 제공합니다.
- [GetResourceSnapshot](#) -이 작업을 사용하여 특정 기회 스냅샷에 대한 실시간 요약물을 얻을 수 있으므로 다른 파트너의 기회에 직접 액세스하지 않고도 up-to-date 정보를 볼 수 있습니다.

다른 파트너의 기회에서 자체적으로 반영해야 하는 관련 변경 사항 또는 업데이트를 식별하는 경우 [UpdateOpportunity](#) 작업을 사용합니다. 이 작업을 통해 관련 데이터를 기회에 선택적으로 통합할 수 있으므로 참여 전반에 걸쳐 조정과 일관성을 보장할 수 있습니다.

참여 멤버 보기

멀티파트너 기회에 참여하면 최대 10명의 파트너가 참여할 수 있습니다. 새 파트너가 참여 초대를 수락할 때마다 참여 멤버 추가 이벤트가 게시되어 모든 현재 멤버에게 변경 사항을 알립니다.

참여 내에서 협업하는 모든 구성원을 보려면 다음 단계를 따르세요.

1. [ListEngagementResourceAssociations](#) 작업을 사용하여 기회와 연결된 참여 ID를 검색합니다.
2. 1단계의 ID를 [ListEngagementMembers](#) 작업에 제공하여 참여 구성원의 파트너 세부 정보를 가져옵니다.

Note

참여의 구성원만 ListEngagementMembers 작업을 호출할 수 있습니다.

리소스 스냅샷 작업 상태 모니터링

멀티파트너 기회로 작업할 때는 다른 파트너를 위한 기회의 스냅샷을 생성하고 참여 중인 다른 파트너의 기회 스냅샷을 검색할 수 있는 역할을 생성해야 합니다. 이 역할이 없으면 기회에 대한 업데이트는 참여 중인 다른 파트너가 사용할 수 없으며, 해당 파트너는 기회의 오래된 스냅샷을 계속 볼 수 있습니다.

참여에서 멀티파트너 기회와 연결된 리소스 스냅샷 작업의 상태를 추적하려면 다음 단계를 따르세요.

1. [ListEngagementResourceAssociations](#) 작업을 사용하여 기회와 연결된 참여 ID를 검색합니다.
2. 1단계의 ID를 [ListResourceSnapshotJobs](#) 작업에 제공하여 참여에서 호출자가 소유한 모든 스냅샷 작업 목록을 생성합니다. 응답에서 작업 ID를 검색합니다.
3. 작업 ID를 [GetResourceSnapshotJob](#) 작업에 제공하여 작업 상태를 추적하고 실행 중인지 확인합니다.

ResourceSnapshotJob 작업은 비동기 작업에 대한 지표를 Amazon CloudWatch에 게시합니다. CloudWatch는 데이터를 읽기 가능하며 실시간에 가까운 지표로 처리하여 서비스의 성능과 상태를 모니터링하는 데 도움이 됩니다.

다음과 같은 지표를 사용할 수 있습니다.

- 결함 - 작업 실행 중에 내부 문제를 계산합니다. 이 지표를 사용하여 서비스 안정성을 모니터링하고 장애 임계값에 대한 경보를 설정합니다.

- 오류 - 작업 처리 중에 고객 관련 문제를 추적합니다. 이 지표는 고객 입력 또는 구성 문제를 식별하는 데 도움이 됩니다.
- 간접 호출 - 성공한 시도와 실패한 시도를 모두 포함하여 총 작업 간접 호출 수를 나타냅니다. 이를 사용하여 시간 경과에 따른 서비스 사용 추세를 추적할 수 있습니다.

Note

지표는 ResourceSnapshotJob 서비스에 활동이 있는 경우에만 CloudWatch에 보고됩니다. 특정 지표에 대한 작업이 없거나 데이터가 없는 경우 해당 지표는 보고되지 않습니다.

ResourceSnapshotJob 작업의 원활한 작동을 보장하려면:

- 지표를 사용하여 서비스가 예상대로 작동하는지 확인합니다.
- CloudWatch 경보를 생성하여 지표를 모니터링하고 지표가 허용 범위를 초과할 때 작업(예: 알림 전송)을 트리거합니다.
- 사전 모니터링을 설정하여 문제를 식별하고 해결합니다.

CloudWatch 지표 사용에 대한 자세한 내용은 [Amazon CloudWatch 사용 설명서](#)를 참조하세요.

기회 연결, 연결 해제 및 할당

기회는 기회 수명 주기의 모든 단계에서 파트너 솔루션, AWS 제품, AWS Marketplace 솔루션, AWS Marketplace 제품, AWS Marketplace 제안 및 AWS Marketplace 제안 세트와 연결되거나 연결 해제될 수 있습니다.

기회를 다른 엔터티와 연결

연결된 개체는 RelatedEntityIdentifiers 객체 내의 GetOpportunity 메서드에 검색됩니다. 를 사용하여 업데이트할 RelatedEntityIdentifiers 수 있습니다 AssociateOpportunity. 이 필드는 UpdateOpportunity 또는 CreateOpportunity 메서드를 사용하여 업데이트할 수 없습니다.

Solutions

StartEngagementFromOpportunityTask 작업을 사용하여 와의 참여를 AWS 시작하기 전에 하나 이상의 파트너 솔루션을 최대 10개까지 연결해야 합니다. AWS 추천에는 파트너 솔루션이 포함될 수도 있고 포함되지 않을 수도 있습니다.

기존 솔루션을 보려면 ListSolutions 작업을 사용합니다.

파트너는 [AWS Partner Central](#)의 Build 섹션에서 솔루션을 생성, 업데이트 및 관리할 수 있습니다.

Important

기회 단계를 커밋됨 또는 시작됨으로 진행하려면 유효한 솔루션 또는 AWS Marketplace 제품 목록을 기회와 연결해야 합니다.

Note

Solutions 엔터티 유형을 사용하는 레거시 파트너 솔루션(S-XXXX 식별자)은 여전히 지원됩니다. 그러나 AwsMarketplaceSolutions 개체 유형은 향후 더 이상 사용되지 않으므로 Solution 개체 유형을 기회와 연결하는 것이 좋습니다.

AWS 제품

최대 20개의 AWS 제품을 기회와 연결할 수 있습니다. 사용 가능한 AWS 제품 목록을 보려면 [AWS GitHub에서 호스팅되는 제품](#) 목록을 사용합니다. AWS 제품과의 연결은 AssociateOpportunity 작업을 통해서만 수행됩니다. 제품을 교체하거나 제거하려면 DisassociateOpportunity 작업을 사용합니다.

Solutions

기회는 판매자 계정의 솔루션과 연결할 수 있습니다. 가 로 RelatedEntityType 설정된 AssociateOpportunity 작업을 사용합니다. AwsMarketplaceSolutions.

엔터티는 퍼블릭 또는 제한 상태여야 합니다. 서비스는 초안 또는 비활성 엔터티를 거부합니다.

솔루션을 연결하려면 ARN이 필요합니다. 다음 예제에서는 솔루션의 ARN 형식을 보여줍니다.

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

파트너 AWS Marketplace 계정 연결을 통해 기본 계정에 연결된 자회사 계정의 솔루션을 연결할 수도 있습니다. 자회사 계정 ID를 포함하여 전체 ARN을 입력합니다.

ListSolutions 작업은 각 솔루션 요약 AwsMarketplaceSolutionArn에 대해 반환하고 ARN별 필터링을 지원합니다.

AWS Marketplace제품

기회는 판매자 계정의AWS Marketplace제품과 연결할 수 있습니다. 가 로 RelatedEntityType 설정된 AssociateOpportunity 작업을 사용합니다AwsMarketplaceProducts.

엔터티는 퍼블릭 또는 제한 상태여야 합니다. 서비스는 초안 또는 비활성 엔터티를 거부합니다.

제품을 연결하려면 ARN이 필요합니다. ARN에는 제품 유형(예: , AmiProductSaaSProduct, ContainerProduct)이 포함됩니다. 다음 예제에서는 AMI 제품의 ARN 형식을 보여줍니다.

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

파트너AWS Marketplace계정 연결을 통해 기본 계정에 연결된 자회사 계정의 제품을 연결할 수도 있습니다. 자회사 계정 ID를 포함하여 전체 ARN을 입력합니다.

AWS Marketplace제안 및 제안 세트

AWS Marketplace제안

기회는AWS Marketplace비공개 제안과 연결될 수 있습니다. 사용 가능한 제안을 보려면AWS Marketplace카탈로그 API의 ListEntities를 사용합니다. 현재는AWS Partner Central에 연결된AWS Marketplace판매자 계정의 제안만 연결할 수 있습니다.

비공개 제안을 연결하려면 ARN이 필요합니다. 샘플:

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

AWS Marketplace제안 세트

기회는AWS Marketplace제안 세트와도 연결할 수 있습니다. 제안 세트를 사용하면 포괄적인 솔루션 패키징을 위해 여러 관련 마켓플레이스 제안을 그룹화할 수 있습니다. 사용 가능한 제안 세트를 보려면 AWS Marketplace카탈로그 API의 ListEntities를 사용합니다.

제안 세트를 연결하려면 ARN이 필요합니다. 샘플:

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

중요: 기회는 제안 하나 또는 제안 세트 하나와 연결될 수 있지만 둘 다와 연결될 수는 없습니다. 이러한 엔터티 유형 중 하나만 한 번에 기회와 연결할 수 있습니다.

다른 엔터티와 기회 연결 해제

DisassociateOpportunity 작업을 사용하여 다음 엔터티 유형에서 기회를 연결 해제합니다.

- Solutions
- AWS제품
- AWS Marketplace솔루션
- AWS Marketplace제품
- AWS Marketplace제안
- AWS Marketplace제안 세트

기회 상태에 따라 이러한 관련 객체의 연결을 해제할 때 다른 검증 규칙이 적용됩니다.

기회 할당

AssignOpportunity를 사용하여 기회 소유자를 변경합니다. Partner Central 사용자를 기회 소유자로 설정할 수 있습니다.

리드 작업

리드란 무엇입니까?

B2B(Businessbusiness-to-business) 판매에서 리드는 회사의 제품 또는 서비스에 관심을 보였지만 아직 판매 기회로 완전히 인정받지 못한 잠재 고객을 나타냅니다. 리드는 영업 파이프라인의 초기 단계이며 활발한 공동 판매 기회로 전환하기 전에 성장과 검증이 필요합니다AWS.

가AWS파트너와 공유하는 리드는 웨비나 참석, 캠페인 참여 또는 파트너 솔루션 문의와 같은 고객 참여 신호를 기반으로 합니다. 이러한 리드에는 파트너가 잠재적 기회를 우선시하고 검증하는 데 도움이 되는 고객 회사 정보, 비즈니스 문제 및 참여 세부 정보와 같은 중요한 컨텍스트가 포함됩니다.

리드 초대 작업

파트너는 잠재 고객이 파트너 솔루션 또는 서비스에 대한 관심을 표현할AWS때 로부터 리드 초대를 받습니다. 리드 초대 프로세스를 통해 파트너는 참여를 약속하기 전에 리드를 평가하여 효율적인 리소스 할당과 높은 전환율을 보장할 수 있습니다.

리드 초대 수신

AWS는 리드 초대를 생성하고 Selling API를 통해 파트너와 공유합니다. 새 리드 초대를 사용할 수 있게 되면는 리드 컨텍스트가 포함된 참여 초대를 적격 파트너에게AWS보냅니다.

파트너는 Amazon EventBridge를 사용하여 Engagement Invitation Created 이벤트를 모니터링해야 합니다. 이 이벤트 알림에는 로 설정된 payloadType 필드가 포함되어 LeadInvitation 있으므로 파트너가 리드 초대와 기회 초대를 구별할 수 있습니다. 이벤트를 수신하면 파트너는 초대 세부 정보를 검색하여 리드를 평가할 수 있습니다.

리드 초대 나열

파트너는 ListEngagementInvitations API 작업을 사용하여 모든 리드 초대를 볼 수 있습니다. 이 작업은 호출 보안 주체가 발신자 또는 수신자인 초대를 검색합니다.

특별히 리드 초대를 필터링하려면 파트너는 값이 인 payloadType 필터를 사용해야 합니다. LeadInvitation. 이 필터는 결과에서 기회 초대를 제외하고 리드 초대만 반환되도록 합니다.

리드 초대는 다음 상태가 될 수 있습니다.

- 보류 중: 파트너 수락 또는 거부 대기 중
- 수락됨: 파트너가 초대를 수락하고 전체 리드 세부 정보에 액세스할 수 있음
- 거부됨: 파트너가 초대를 거부했습니다.
- 만료됨: 초대가 작업 없이 만료 날짜를 초과했습니다.

리드 초대 평가

리드 초대를 수락하기 전에 파트너는 GetEngagementInvitation API 작업을 사용하여 자세한 초대 정보를 검색해야 합니다. 이를 통해 정보에 입각한 수락 또는 거부 결정을 내리는 데 필수적인 컨텍스트를 얻을 수 있습니다.

리드 초대 페이로드에는 다음이 포함됩니다.

고객 정보(사용 가능한 사전 승인):

- 회사 이름, 산업 및 국가
- 웹사이트 URL
- 시장 세그먼트(Enterprise, Large, Medium, Small, Micro)
- AWS성숙도 수준(평가, 단일 계정, 다중 계정)

고객 상호 작용:

- 소스 유형 및 캠페인 정보

- 고객 조치(양식 작성, 웨비나 참석 등)
- 고객이 제공한 비즈니스 문제 설명
- 사용 사례 범주
- 연락처 제목(BANT 검증을 위한 권한 컨텍스트 제공)

Important

고객 연락처 정보(이름, 이메일, 전화번호)는 초대 단계에서 표시되지 않습니다. 연락처 세부 정보는 초대를 수락하여 고객 프라이버시를 보장하고 리드 팸핑 동작을 방지한 후에만 사용할 수 있습니다.

리드 초대 수락

파트너가 리드를 찾기로 결정하면 AcceptEngagementInvitation API 작업을 사용하여 초대를 수락해야 합니다. 이 작업은 파트너를 참여에 추가하고 고객 연락처 정보를 포함한 전체 리드 세부 정보에 대한 액세스 권한을 부여합니다.

수락 후:

- 파트너가 참여의 구성원으로 추가됩니다.
- 고객 연락처 정보가 참여 리드 컨텍스트를 통해 표시됩니다.
- 리드는 AWS 시스템에서 파트너 적격 리드(PQL)로 분류됩니다.
- 리드는 파트너의 활성 리드 목록에 표시됩니다.
- 파트너는 리드를 양성하고 고객 응대를 확립하기 시작할 수 있습니다.

파트너는 각 초대에 AcceptEngagementInvitation 대해를 호출하여 프로그래밍 방식으로 여러 리드 초대를 수락할 수 있으므로 고품질 리드를 효율적으로 대량 처리할 수 있습니다.

리드 초대 거부

리드가 파트너의 기능, 용량 또는 비즈니스 포커스와 일치하지 않는 경우 파트너는 RejectEngagementInvitation API 작업을 사용하여 초대를 거부할 수 있습니다. 리드 초대를 거부할 때 파트너는 리드 라우팅 및 매칭을 AWS 개선하는 데 도움이 되는 거부 이유를 제공해야 합니다. 일반적인 거부 이유는 다음과 같습니다.

- 지리적 범위 부족

- 사용 사례에 대한 기술 전문 지식 부족
- 현재 용량 제약 조건
- 고객 업계 불일치
- 예산 불일치

거부 후:

- 리드가 파트너의 보류 중인 초대 대기열에서 제거됩니다.
- 고객 연락처 정보는 파트너와 공유되지 않습니다.
- 리드는 AWS 시스템에서 파트너 거부 리드(PRL)로 분류됩니다.
- 참조용으로 파트너의 거부된 초대 목록에 초대가 표시됩니다.

고객 동의 요구 사항이 충족되면 거부된 리드는 다른 파트너에게 재할당할 수 있습니다.

수락된 리드 관리

리드 초대를 수락한 후 파트너는 전체 리드 정보에 액세스하고, 고객 관계를 발전시키고, 검증 단계를 통해 리드를 추적할 수 있습니다.

리드 세부 정보 보기

파트너는 GetEngagement API 작업을 사용하여 전체 리드 세부 정보에 액세스합니다. 참여에는 고객과 고객의 상호 작용에 대한 포괄적인 정보가 포함된 리드 컨텍스트가 포함됩니다 AWS.

리드 컨텍스트에는 다음이 포함됩니다.

전체 고객 프로필:

- 원래 초대 의 모든 회사 정보
- 모든 고객 상호 작용에 대한 전체 연락처 세부 정보(이름, 이메일, 전화, 직책)
- AWS 성숙도 수준 및 시장 세그먼트

상호 작용 기록:

- 여러 고객 접점을 단일 참여로 통합
- 각 상호 작용에는 소스 정보, 고객 작업, 비즈니스 문제 및 연락처 세부 정보가 포함됩니다.

- 상호 작용은 전체 고객 여정 보기를 제공하기 위해 시간순으로 나열됩니다.

검증 상태:

- 현재 리드 검증 상태(부적격, 적격, 부적격 또는 사용자 지정 상태)
- 파트너는 검증 프로세스를 진행하면서이 상태를 업데이트할 수 있습니다.

이 포괄적인 보기를 통해 파트너는 각 상호 작용을 별도의 리드로 처리하는 대신 여러AWS캠페인 및 터치포인트에서 고객의 완전한 참여 기록을 이해할 수 있습니다.

활성 리드 나열

파트너는 contextType 필터가 로 설정된 ListEngagements API 작업을 사용하여 모든 활성 리드를 검색할 수 있습니다Lead. 그러면 파트너가 멤버이고 참여에 리드 컨텍스트가 포함된 참여가 반환됩니다.

목록 응답에는 다음과 같은 주요 정보가 포함됩니다.

- 참여 ID 및 제목
- 고객 회사 이름
- 현재 참여 점수
- 자격 상태
- 상호 작용 수
- 타임스탬프 생성 및 수정

파트너는이 목록을 사용하여 대시보드를 구축하고, 리드 파이프라인 상태를 추적하고, 주의가 필요한 리드를 식별할 수 있습니다.

Prospecting Insights로 리드 강화

파트너는 아웃리치에 투자하기 전에 우선 순위를 더 잘 정하고 자격을 부여할 수 있는AWS인사이트로 수락된 리드를 강화할 수 있습니다. 보강은 잠재 고객 발굴 작업을 통해 비동기적으로 수행되며, 이는 검증 결정을 지원하기 위해AWS파생 신호로 리드 참여를 강화합니다.

잠재 고객 발굴 작업 시작

파트너는 StartProspectingFromEngagementTask API 작업을 사용하여 보강을 시작합니다. 이 작업은 단일 요청에서 최대 100개의 참여 식별자를 허용하므로 파트너가 리드를 일괄적으로 보강할

수 있습니다. 작업은 비동기적으로 실행되며 파트너가 진행 상황을 추적TaskArn하는 데 사용하는 TaskId 및를 즉시 반환합니다. 초기 TaskStatus는 PENDING, IN_PROGRESS, COMPLETED또는 중 하나입니다FAILED.

파트너는 선택적으로 TaskName를 제공하여 작업에 레이블을 지정하고 ClientToken를 제공하여 재 시도 간에 멍등성을 보장할 수 있습니다.

결과에 대한 폴링

잠재고객 발굴은 비동기식이므로 파트너는 시작 시 TaskId 반환된와 함께 GetProspectingFromEngagementTask API 작업을 사용하여 완료를 폴링합니다. 응답은 전체 작업 상태와 제출된 각 식별자에 대한 참여당 결과를 보고합니다.

각 참여는 독립적으로 처리되므로 동일한 작업의 다른 사용자에게 영향을 주지 않고 개별 참여는 성공하거나 실패할 수 있습니다. 각 참여의 결과에는 다음이 포함됩니다.

- 참여 식별자: 처리된 참여입니다.
- 상태: PENDING, IN_PROGRESSCOMPLETED, 또는 FAILED.
- Prospecting context identifier: 참여가 성공적으로 처리될 때 채워집니다. 이 식별자는 참여에 추가된 강화된 잠재 고객 발굴 컨텍스트를 참조하며 후속 작업에 사용할 수 있습니다.
- 사유 코드 및 메시지: 참여 실패 시에만 채워지며, 열거형 실패 코드와 사람이 읽을 수 있는 설명과 제안된 복구 단계를 제공합니다.

여러 작업 모니터링

파트너는 여러 리드에서 보강을 추적하기 위해 ListProspectingFromEngagementTasks API 작업을 사용합니다. 이 작업은 호출자의 계정에서 시작한 모든 잠재 고객 발굴 작업을 나열하고 구성 가능한 정렬과 함께 작업 식별자, 작업 이름 또는 시작 시간 범위별 선택적 필터를 지원합니다. 응답은 페이지가 매겨집니다. 각 응답의 NextToken 값을 사용하여 후속 페이지를 검색합니다.

보강이 성공적으로 완료되면AWS인사이트가 잠재 고객 컨텍스트로 참여에 추가됩니다. 파트너는를 사용하여 보강된 세부 정보를 검색GetEngagement하고 이를 사용하여 리드의 검증 상태를 설정하고 변환으로 진행할 대상을 결정할 수 있습니다.

리드 정보 업데이트

파트너는 리드를 양성하고 추가 정보를 수집하면서 UpdateEngagementContext API 작업을 사용하여 리드 세부 정보를 업데이트할 수 있습니다. 이 작업을 통해 파트너는 참여 내에서 리드 컨텍스트를 수정할 수 있습니다.

파트너는 다음을 업데이트할 수 있습니다.

검증 상태: 파트너는 내부 검증 프로세스를 진행하면서 검증 상태를 업데이트해야 합니다.

- 정규화되지 않음: 리드를 수락한 후 기본 상태이며, 리드에 평가가 필요함을 나타냅니다.
- 적격: 리드가 자격 기준을 충족하고 기회로 전환할 가능성이 높음
- 부적격: 리드가 기준을 충족하지 않거나 파트너의 솔루션에 적합하지 않음
- 사용자 지정 상태: 파트너는 내부 프로세스와 일치하도록 자체 검증 상태를 정의할 수 있습니다.

리드 메타데이터: 파트너는 검증 및 양육 프로세스와 관련된 추가 정보를 추가하거나 업데이트할 수 있습니다.

검증 상태를 업데이트하면 파트너가 리드 진행 상황을 추적하고, 정확한 파이프라인 보고서를 생성하고, 기회로 전환할 준비가 된 리드를 식별할 수 있습니다.

AWS업데이트 모니터링

AWS는 새로운 고객 참여 신호 또는 정교한 참여 점수를 기반으로 리드 정보를 업데이트할 수 있습니다. 가 참여를AWS업데이트하면 파트너는 Amazon EventBridge를 통해 Engagement Updated 이벤트를 수신합니다.

이러한 업데이트에는 다음이 포함될 수 있습니다.

- 새로운 고객 활동을 기반으로 참여 점수 개선
- 새 캠페인 또는 터치포인트의 추가 고객 상호 작용
- 업데이트된 고객 정보

파트너는 이러한 이벤트를 모니터링하고를 호출GetEngagement하여 최신 리드 세부 정보를 검색해야 합니다. 이를 통해 파트너는 리드의 우선 순위 지정 및 개발을 위한 최신 정보를 얻을 수 있습니다.

Engagement Updated 이벤트에는 파트너가 리드(리드 컨텍스트를 사용한 참여)에 대해 특별히 필터링할 수 있는 contextTypes 필드가 포함됩니다.

리드를 기회로 변환

리드가 자격을 갖추고 심각한 구매 의도를 보이면 파트너는 이를 공식 공동 판매 협업의 기회로 전환할 수 있습니다AWS. 이 변환은 리드 양육(주로 파트너 중심)에서 기회 관리(와 협업)로의 전환을 나타냅니다AWS.

권장 접근 방식: 편의 API

파트너는 StartOpportunityFromEngagementTask 편의 API 작업을 사용하여 기회 생성 및 연결 프로세스를 간소화할 수 있습니다. 이 작업 API는 여러 작업을 자동으로 오케스트레이션합니다.

1. 리드 컨텍스트의 예비 정보로 초안 기회를 생성합니다.
2. 리소스 스냅샷을 통해 소스 참여에 기회를 연결합니다.
3. CustomerProject 컨텍스트를 참여에 추가합니다.

이 편리한 방법은 필요한 API 호출 수를 줄이고 적절한 어트리뷰션 추적을 보장합니다. 이 접근 방식을 사용하는 파트너는 여전히 다음을 수행해야 합니다.

- 를 사용하여 기회 세부 정보 작성 UpdateOpportunity
- 를 사용하여 파트너 솔루션 연결 AssociateOpportunity
- 준비가 StartEngagementFromOpportunityTask 되면 를 사용하여 기회 제출

편의 API는 통합 복잡성을 최소화하려는 자동화된 lead-to-opportunity 워크플로를 사용하는 파트너에게 특히 유용합니다.

대체 접근 방식: Step-by-step API

초안 기회 생성

리드를 변환하는 첫 번째 단계는 CreateOpportunity API 작업을 사용하여 초안 기회를 생성하는 것입니다. 이렇게 하면 Lifecycle.ReviewStatus 설정된 기회가 생성됩니다 Pending Submission. 이 단계에서는 검증을 AWS위해에 아직 기회가 제출되지 않았습니다.

파트너는 다음을 포함하여 리드 양육 중에 수집된 정보로 기회를 채워야 합니다.

- 고객 계정 세부 정보
- 프로젝트 정보 및 비즈니스 문제
- 예상 고객 지출
- 대상 종료일
- 검증 중에 수집된 기타 관련 세부 정보

초안 기회를 통해 파트너는 제출하기 전에 전체 정보를 준비 AWS하여 승인률을 높이고 검증 속도를 높일 수 있습니다.

기회를 리드 참여와 연결

초안 기회를 생성한 후 파트너는 CreateResourceSnapshot API 작업을 사용하여 이를 소스 리드 참여에 연결해야 합니다. 이 단계는 다음에 매우 중요합니다.

- **속성 추적:** 리드 초대부터 기회 종료까지 소싱 체인을 구축하여 마케팅 캠페인 및 리드 소스에 대한 정확한 ROI 계산을 가능하게 합니다.
- **변환 지표:** AWS 및 파트너가 lead-to-opportunity 변환율을 측정하고 성공적인 리드 소스를 식별할 수 있습니다.
- **컨텍스트 보존:** 모든 원래 상호 작용 및 참여 신호를 포함하여 전체 고객 여정 기록을 유지합니다.

리소스 스냅샷을 생성할 때 파트너는 다음을 지정합니다.

- 참여 ID(소스 리드 참여)
- 기회 식별자
- 리소스 유형(기회)

이 작업은 CustomerProject 컨텍스트를 참여에 자동으로 추가하여 리드가 활성 기회로 전환되었음을 알립니다. 이제 참여에는 원래 리드 컨텍스트(보존 기록)와 새로운 CustomerProject 컨텍스트(활성 기회를 나타냄)가 모두 포함됩니다.

기회 세부 정보 완료

기회가 생성되고 연결되면 파트너는 제출 전에 추가 기회 요구 사항을 완료해야 합니다. 자세한 내용은 AssociateOpportunity API 작업을 참조하세요.

프로젝트 세부 정보 업데이트: 파트너는 UpdateOpportunity API 작업을 사용하여 리드 검증 중에 수집된 프로젝트 정보, 고객 비즈니스 문제, 예상 지출 및 기타 관련 세부 정보를 세분화할 수 있습니다.

기회 제출

필요한 모든 정보가 완료되면 파트너는 StartEngagementFromOpportunityTask 편의 API를 사용하여 AWS 검증 기회를 제출합니다. 이렇게 하면 기회가 초안 상태에서 AWS 검토 프로세스로 전환됩니다.

검증 요구 사항: 제출하기 전에 참여에 CustomerProject 컨텍스트가 있어야 합니다. 이 컨텍스트는 CreateResourceSnapshot를 사용하여 기회를 참여에 연결할 때 자동으로 생성됩니다. 이 컨텍스트가 누락된 경우 제출이 실패합니다.

제출 후:

- 기회의가 로 Lifecycle.ReviewStatus 변경됩니다. Submitted
- 리드는AWS시스템에서 파트너 영업 적격 리드(PSQL)로 분류됩니다.
- AWS는 검증을 시작하여 기회 세부 정보가 정확하고 완전한지 확인합니다.
- 검토 프로세스가 완료될 때까지 기회를 변경할 수 없습니다.

그런 다음 기회는 "기회 작업" 설명서에 설명된 표준 검증 워크플로를 따르며 검토 중, 조치 필요, 승인 됨 또는 부적격과 같은 상태를 진행합니다.

거래 규모 조정 인사이트 활용

거래 규모 조정이란 무엇입니까?

거래 규모 조정은AWS파트너가 기회를 생성하거나 업데이트할 때 AI를 사용하여 자동화된 거래 규모 추정 및AWS서비스 권장 사항을 제공하는 Partner Central의 APN 고객 참여(ACE) 기회 내의 기능입니다.

AI 예상 MRR 추정치

거래 규모 조정 인사이트는 파트너의 과거AWS기회, 사용 사례 및 비즈니스 설명을 기반으로 예측된 월별 반복 수익(MRR) 범위의 중앙값을 제공합니다. 파트너는 거래에 대한 자세한 정보를 수집하고 판매 주기를 진행하므로 총 MRR을 검토하고 업데이트해야 합니다.

AI 권장AWS제품

AI 권장 제품은 고객 비즈니스 문제 설명 및 기회 세부 정보를 기반으로 식별됩니다. AI는 기술 요구 사항 및 사용 사례에 맞는 관련AWS제품을 권장합니다. 파트너는 이러한 제안을 검토하고 고객의 특정 요구 사항에 맞게 제품 선택을 사용자 지정해야 합니다.

요금 계산기 URLs을 사용한 향상된 인사이트

또한 파트너는AWS요금 계산기 URLs 가져와 서비스 선택을 자동으로 채우고 마이그레이션 가속화 프로그램(MAP) 자격 지표, 최적화 권장 사항 및 잠재적 비용 절감 분석을 포함한 향상된 인사이트를 받을 수 있습니다.

소스로 구분된 데이터 이해

거래 크기 조정은 기회 가치에 대한 포괄적인 가시성을 제공하기 위해 두 개의 개별 소스에서 인사이트를 제공합니다.

- AWS소스: 유사한 과거 기회의 패턴을 기반으로 하는 AI 권장 제품
- 파트너 소스:AWS요금 계산기에서 가져온 자체 견적

소스 간 차이 처리

파트너가 제공한 추정치가AWS권장 사항과 크게 다른 경우 파트너는 다음을 수행해야 합니다.

1. 기회 세부 정보를 검토하여 모든 관련 정보가 캡처되었는지 확인합니다.
2. 요금 계산기 URL 구성이 실제 요구 사항과 일치하는지 확인
3. 유사한 과거 기회를 통해 얻은 데이터 포인트로AWS권장 사항을 고려하세요.
4. 특정 고객 컨텍스트 및 요구 사항에 따라 정보에 입각한 의사 결정

Deal Sizing Insights에 액세스

파트너는 GetAwsOpportunitySummary API 작업을 통해 거래 크기 조정 데이터에 액세스합니다.이 작업은 거래 크기 조정 예측, 제품 권장 사항 및 최적화 인사이트가 포함된 확장 AwsOpportunitySummary 객체를 반환합니다.

거래 크기 조정 데이터를 사용할 수 있게 되는 경우

충분한 고객 및 프로젝트 세부 정보로 기회가 생성된 후 거래 규모 조정 인사이트를 사용할 수 있습니다. 시스템은 기회 속성을 분석하고 다음을 생성합니다.

1. AI 예상 MRR: 기회 특성에 따라 자동으로 계산됨
2. AWS제품 권장 사항: 고객의 비즈니스 문제 및 기술 요구 사항과 관련된 서비스
3. 향상된 인사이트(요금 계산기 URL이 제공된 경우): MAP 자격, 최적화 권장 사항 및 비용 절감 분석

Note

일부 기회에는 거래 규모 조정 인사이트를 사용할 수 없습니다. 자격은 파트너 기록 및 제공된 기회 세부 정보의 완전성과 같은 요인에 따라 달라집니다.

거래 크기 조정 입력 제공

파트너는 CreateOpportunity 및 UpdateOpportunity API 작업을 통해 거래 규모 조정에 대한 입력을 제공합니다. 시스템은 기회 속성을 사용하여 추천을 생성합니다.

총 계약 금액(TCV) 제공

월별 반복 수익이 아닌 총 계약 가치로 추정하는 파트너는 거래 가치를 직접 제공할 수 있습니다. AWS는 AI 기반 변환 모델을 사용하여 TCV를 AWS MRR로 자동 변환합니다.

TCV에서 거래 규모 조정의 경우 ExpectedContractDuration를 Months (유효 값 범위는 1~144개월) TargetCompany로, 를 Self 및 Frequency로 제공합니다 None.

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

AWS는 AWS MRR 추정치를 도출하고에서 두 항목을 모두 반환합니다GetOpportunity.

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

}

위의 예에서 ExpectedCustomerSpend 첫 번째 배열에는 AWS MRR 추정치(TargetCompany: "AWS", Frequency: "Monthly")가 포함되어 있으며 두 번째 배열은 파트너의 총 계약 금액(TargetCompany: "Self",)입니다Frequency: "None".

TCV 검증 규칙

모든 TCV 검증 오류는 INVALID_VALUE와 함께 오류 코드를 반환합니다Reason: "BUSINESS_VALIDATION_FAILED". 구별 인자는 fieldName 및 입니다Message.

조건	Field	메시지
를 사용하지 않고 자체 입력 ExpectedContractDuration	project.expectedContractDuration	자체 항목이 있는 경우 ExpectedContractDuration이 필요합니다.
ExpectedContractDuration 자체 항목 없음(생성만 해당)	project.expectedCustomerSpend	ExpectedContractDuration이 있는 경우 자체 항목이 필요합니다.
여러 자체 항목	project.expectedCustomerSpend	하나의 자체 항목만 허용됩니다.
기간이 있는 경우 유효하지 않은 TargetCompany	project.expectedCustomerSpend.targetCompany	TargetCompany는 'AWS' 또는 'Self'여야 합니다.
AWS와 자체 간의 통화 불일치	project.expectedCustomerSpend.currencyCode	CurrencyCode는 항목 간에 일치해야 합니다.
자체 입력 + EstimationUrl 함께	project.expectedCustomerSpend.estimationUrl	자체 입력 및 EstimationUrl이 공존할 수 없음

Note

에서 ExpectedContractDuration가 자체 항목 없이 존재하지만 파트너 거래 값이 스토리지에 있는 UpdateOpportunity 경우 시스템은 자체 항목을 자동으로 재구성합니다. 오류가 발생하지 않습니다.

TCV와 수동 MRR 간 전환

TCV 모드에서 수동 MRR로 다시 전환하려면 ExpectedContractDuration 로 명시적으로 설정하고 AWS 항목만 null 제공합니다.

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

Note

페이로드 ExpectedContractDuration에서 생략하면(필드를 전송하지 않음) "변경 없음"을 의미합니다. 기존 TCV 데이터가 보존됩니다. 명시적 전송은 "TCV 모드 지우기"를 null 의미합니다.

TCV 및 요금 계산기 URLs

파트너가 Self 항목(TCV 모드)과를 모두 제공하는 경우 EstimationUrlAPI는 검증 오류를 반환합니다. 자체 항목과 EstimationUrl은 동일한 요청에 공존할 수 없습니다.

수동 MRR 견적 제공

파트너는 `Project.ExpectedCustomerSpend[].Amount` 필드를 사용하여 MRR 추정치를 수동으로 입력할 수 있습니다.

Note

`CurrencyCode` 필드는 USD 및를 허용합니다EUR. AWS 파티션이 `aws-eusc` (AWS European Sovereign Cloud)인 경우 통화 코드는 여야 합니다EUR.

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

이 필드는 모든 파트너가 사용할 수 있으며 동일한 값을 설정하여 AI 예측을 수락하거나 파트너가 제공한 값으로 재정의하는 데 사용할 수 있습니다.

요금 계산기 URLs 제공

파트너는 선택적으로 `Project.ExpectedCustomerSpend[].EstimationUrl` 필드를 통해AWS 요금 계산기 URLs을 제공하여 서비스 구성을 자동으로 가져오고 MAP 자격 지표, 최적화 권장 사항 및 비용 절감 분석을 포함한 향상된 인사이트를 받을 수 있습니다.

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

```

```

        "TargetCompany": "AWS"
    }
]
}
}

```

유효한 요금 계산기 URL이 제공되면 시스템은 계산기 구성에서 MRR 금액을 자동으로 계산하고 금액 필드와의 파트너 소스에 자세한 제품 수준 지출 데이터를 모두 채웁니다. `AwsProductsSpendInsightsBySource`. 파트너는 `EstimationUrl`을 제공할 때 `Amount`를 null로 설정해야 합니다. 시스템이 자동으로 계산합니다.

잘못된 형식 URLs과 확인할 수 없는 URLs은 `ValidationException`로 거부됩니다.

`Amount`와 `EstimationUrl`이 함께 작동하는 방식

기회에 대한 월별 반복 수익(MRR) 추정치를 제공하는 방법에 유연성이 있습니다. API는 수동 금액과 AWS요금 계산기 URLs의 다양한 조합을 지능적으로 처리하여 기회를 항상 성공적으로 생성할 수 있도록 합니다.

수동 양만 사용

`Amount` 필드만 입력하면 API가 수동 견적을 저장하고 `EstimationUrl`을 null로 설정합니다. 이 접근 방식은 고객 대화 또는 자체 계산에서 특정 MRR 추정치가 있을 때 유용합니다.

요청 예제:

```

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

```

AWS요금 계산기 URL만 사용

`EstimationUrl` 필드만 제공하는 경우:

- 유효한 URL - API는 계산기 구성에서 MRR 양을 계산하고 URL과 계산된 양을 모두 저장합니다.

- 잘못된 URL - API는 URL이 잘못된 이유에 대한 세부 정보와 함께 `ValidationException`을 반환합니다.

요청 예제:

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

오류 응답의 예(잘못된 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"
}
```

Amount와 EstimationUrl 모두 제공

두 필드를 모두 제공하면 API는 기회가 성공적으로 생성되도록 우선 순위를 지정합니다.

1. 두 값이 모두 변경되지 않은 경우 - 이러한 필드에 대한 업데이트가 수행되지 않습니다.
2. URL이 유효하지 않음 - API는 제공된 양을 저장하고 EstimationUrl을 null로 설정하여 기회 생성을 계속할 수 있습니다.
3. URL이 유효하고 계산된 금액이 제공된 금액과 일치 - 두 값이 모두 저장됩니다.
4. URL이 유효하지만 계산된 금액이 제공된 금액과 일치하지 않음 - API는 제공된 금액을 저장하고 오류를 반환하지 않고 EstimationUrl을 null로 설정합니다.

Note

두 값이 모두 기존 값과 일치하지 않으면 API는 먼저 URL에서 금액을 계산하려고 시도합니다. 불일치 또는 검증 문제가 있는 경우 수동으로 제공된 금액이 항상 우선합니다.

주요 이점: 잘못된 URLs 기회 생성을 차단하지 않음

이 접근 방식을 사용하면 유효하지 않거나 액세스할 수 없는 AWS요금 계산기 URLs 인해 기회를 생성하거나 업데이트할 수 없습니다. 수동으로 제공된 금액이 항상 충돌 또는 검증 문제가 있을 때 우선하므로 기회 데이터를 완전히 제어할 수 있습니다.

확장 AwsOpportunitySummary 데이터 모델 이해

이 섹션에서는 GetAwsOpportunitySummary API 작업에서 반환되는 응답 구조를 설명합니다.

Note

AI 예측 MRR이에 반환됩니다. `Project.ExpectedCustomerSpend[].Amount`

GetAwsOpportunitySummary API 작업은 소스 어트리뷰션으로 포괄적인 지출 분석을 제공하는 새로운 `Insights.AwsProductsSpendInsightsBySource` 구조를 통해 거래 크기 조정 인사이트를 반환합니다.

소스 분리: AWS vss 파트너 데이터

거래 규모 조정은 소스별로 인사이트를 분리하여 투명성을 제공하고 이중 계산을 방지합니다.

- AWS소스:의 AI 제품 권장 사항AWS
- 파트너 소스: 파트너가 제공한 요금 계산기 URLs에서 가져온 지출 데이터 및 제품 세부 정보

이러한 분리를 통해 파트너는 다음을 수행할 수 있습니다.

1. 견적을AWS권장 사항과 비교
2. 크기 조정 가정 검증
3. 실수로 두 소스를 결합하여 합계 팽창 방지
4. 데이터 출처에 대한 가시성 유지

구조 개요

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

필드 참조

AwsProductsSpendInsightsBySource 맵

소스로 구분된 지출 인사이트는 AWS 권장 사항 및 파트너 추정치에 대한 독립적인 분석을 제공합니다.

Insights.AwsProductsSpendInsightsBySource 필드는 최대 2개의 키를 포함하는 맵입니다.

Key(키)	유형	설명
AWS	AwsProductsSpendInsights	의 권장 제품을 포함한 AI 생성 인사이트 AWS
파트너	AwsProductsSpendInsights	세부 서비스 비용 및 최적화를 포함하여 요금 계산기 URLs에서 파생된 파트너 제공 인사이트

Note

사용 가능한 데이터가 있는 소스만 응답에 포함됩니다. 새로운 기회는 일반적으로 AWS 소스만 채워진 상태에서 시작됩니다.

AwsProductInsights 객체

총 금액, 최적화 절감, 프로그램 범주 분석, 자세한 제품 수준 인사이트를 포함하여 단일 소스(AWS 또는 파트너)에 대한 포괄적인 지출 분석.

각 소스 객체에는 다음 필드가 포함됩니다.

Field	유형	설명
CurrencyCode	문자열	ISO 4217 통화 코드. 지원되는 값은 "USD" 및 "EUR"입니다.
빈도	문자열	고객이 예상 금액을 얼마나 자주 소비할 것으로 예상되는지 나타냅니다. 빈도 필드에는 월별 값만 허용되며, 이는 반복되는 월별 지출을 나타냅니다.
TotalAmount	문자열	최적화 전이 소스의 총 예상 지출
TotalOptimizedAmount	문자열	권장 최적화 적용 후 총 예상 지출
TotalPotentialSavingsAmount	문자열	최적화 구현을 통해 얻을 수 있는 절감액 정량화
TotalAmountByCategory	객체	AWS 프로그램 및 현대화 경로에 매핑된 지출 금액
AwsProducts	배열	비용 및 최적화 권장 사항을 포함한 제품 수준 세부 정보

현재 상태 제한: AWS소스의 경우 제품별 지출 권장 사항을 사용할 수 없는 경우 TotalAmount, TotalOptimizedAmount 및 TotalPotentialSavingsAmount를 생략할 수 있습니다. 파트너는 이러한 경우 총AWS MRR 추정치Project.ExpectedCustomerSpend[].Amount를 참조해야 합니다.

TotalAmountByCategory 객체

지출 금액을AWS프로그램 및 전략적 이니셔티브에 매핑합니다.

범주 키	설명
MAP 적격	마이그레이션 가속화 프로그램 자금 지원을 받을 수 있는 서비스에 지출
CAT 적격	비용 할당 및 추적을 지원하는 서비스
경로 - 클라우드 네이티브로 이동	클라우드 네이티브 현대화에 부합하는 서비스
경로 - 관리형 서비스로 이동	관리형 서비스 채택을 지원하는 서비스
경로 - 오픈 소스로 이동	오픈 소스 서비스 대안
경로 - 컨테이너로 이동	컨테이너 기반 서비스 옵션
경로 - 서버리스로 이동	서버리스 컴퓨팅 서비스
경로 - 데이터베이스로 이동	데이터베이스 현대화 서비스
경로 - 분석으로 이동	분석 및 데이터 처리 서비스

Note

개별 제품이 여러 범주에 속할 수 있으므로 총 범주 금액이 TotalAmount를 초과할 수 있습니다.

AwsProducts 배열

프로그램 자격 지표(MAP, 현대화 경로), 비용 견적 및 최적화 권장 사항이 포함된AWS서비스 목록입니다.

각 제품 객체에는 다음이 포함됩니다.

Field	유형	필수	설명
ProductCode	문자열	예	AWS기회 연결에 사용되는 Partner Central 제품 식별자
ServiceCode	문자열	No	요금 계산기 서비스 코드(원본 계산기 URL에 대한 링크)
Categories	배열	예	이 제품이 사용할 수 있는 프로그램 및 경로 범주 목록
Amount	문자열	No	최적화 전 기준 서비스 비용(AWS소싱된 권장 사항이 null일 수 있음)
OptimizedAmount	문자열	No	최적화 적용 후 서비스 비용(AWS소싱된 권장 사항이 null일 수 있음)
PotentialSavingsAmount	문자열	No	최적화를 통한 서비스 별 비용 절감(AWS소싱된 권장 사항이 null일 수 있음)
최적화	배열	아니요	이 제품에 대한 특정 최적화 권장 사항 목록

Null 처리: 제품별 지출 권장 사항을 사용할 수 없는 경우 AWS소스 제품, 금액, OptimizedAmount 및 PotentialSavingsAmount 필드가 null일 수 있습니다. 시스템은 `Project.ExpectedCustomerSpend[0].Amount` 대신 `총 MRR`을 제공합니다.

최적화 객체

각 최적화 권장 사항에는 다음이 포함됩니다.

Field	유형	Description
설명	문자열	최적화 전략에 대한 사람이 읽을 수 있는 설명
SavingsAmount	문자열	이 최적화를 구현하여 얻을 수 있는 비용 절감 정량화

예제

예제 1: AWS 권장 사항 전용(현재 상태)

이 예제는 AI 권장 사항만 사용할 수 있고 제품별 지출 금액을 아직 권장할 수 없는 경우의 일반적인 응답을 보여줍니다.

```
{
  "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"
    ]
}
}

```

요점: AWS 소스는 범주가 있지만 제품당 금액이 없는 권장 제품을 보여줍니다. 를 통해 사용 가능한 총 MRR 추정치입니다 `Project.ExpectedCustomerSpend[].Amount`.

예제 2: 파트너 요금 계산기 가져오기

이 예제는 파트너가 유효한 요금 계산기 URL을 제공한 경우 향상된 인사이트를 보여줍니다.

```

{
    "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": []
}
}

```

요점: 파트너 소스에는 제품별 금액이 포함된 전체 지출 세부 정보가 포함되어 있습니다. 최적화 권장 사항은 실행 가능한 비용 절감 전략을 제공합니다. 범주 분석에는 MAP 자격(총 32,500 USD 중 22,500 USD)이 표시됩니다. 제품에는 요금 계산기 구성에 대한 ServiceCode 연결이 포함됩니다.

예제 3: 두 소스 모두 존재(비교 시나리오)

이 예제에서는 AWS 권장 사항과 파트너 추정치가 모두 함께 표시되어 비교가 가능한 방법을 보여줍니다.

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


요점: 두 소스 모두 있음:AWS권장 사항(총 28,000 USD) 및 파트너 계산기(총 32.5,000 USD).AWS 권장 사항에는 4개의 제품이 표시되고 파트너 계산기에는 4개의 제품이 표시됩니다(중첩 3개). 파트너를 통해 사용 가능한 세부 추정치AWS합계와AWS권장 사항을 비교할 수 있습니다Project.ExpectedCustomerSpend[].Amount.

모범 사례

Null 값 처리

1. EstimationUrl 필드: 대부분의 파트너에 대해 EstimationUrl이 null일 것으로 예상합니다. 이는 표준 동작이므로 오류로 처리해서는 안 됩니다. 수동 MRR 항목 옵션을 기본 경로로 표시합니다.
2. 누락된 범주:AWS소스에 TotalAmountByCategory가 없는 경우 이는 제품별 권장 사항을 사용할 수 없음을 나타냅니다. 가격 계산기 URLs.

최적화 권장 사항

1. 실행 가능한 인사이트 표시: 최적화 설명과 절감 금액을 눈에 띄게 표시하여 파트너가 비용 절감 기회를 이해하는 데 도움이 됩니다.
2. 총 절감액: 전체 최적화 가능성을 보여주기 위해 제품 간에 PotentialSavingsAmountPotentialSavingsAmount를 합산합니다.
3. 영향별 우선순위 지정: 파트너가 가장 영향력이 높은 권장 사항에 집중할 수 있도록 SavingsAmount 별로 최적화를 정렬합니다.

업데이트 모니터링

1. 정기적으로 폴링: 기회 생성 워크플로 중에 GetAwsOpportunitySummary 주기적으로(권장: 5분마다) 폴링합니다.
2. 비동기 생성 처리: 기회 생성 후 거래 크기 조정 인사이트를 즉시 사용할 수 없을 수 있습니다. 적절한 로드 상태를 구현하고 로직을 재시도합니다.

API 통합 패턴

1. 기존 통합 활용: 이미 호출하는 파트너는 응답에 추가 필드GetAwsOpportunitySummary만 사용하면 되므로 새 API 호출이 필요하지 않습니다.
2. 성능 저하: Deal Sizing 데이터를 아직 사용할 수 없거나 불완전한 시나리오를 처리하도록 UIs를 설계합니다.

데이터 품질 및 정확성

1. 요금 계산기 URLs 검증: null EstimationUrl 응답을 방지하기 위해 제출 전에 URLs이 유효한지 확인합니다.
2. 풍부한 기회 세부 정보 제공: 더 완전한 기회 정보(고객 세부 정보, 프로젝트 설명, 기술 요구 사항)는 더 정확한 AI 권장 사항을 제공합니다.
3. 차이 검토: 파트너 추정치가 AWS 권장 사항과 크게 다를 경우 기회 세부 정보를 검토하여 모든 관련 컨텍스트가 캡처되었는지 확인합니다.

모범 사례

이벤트에 대응

AWS Partner Central API에서 이벤트를 처리할 때 중복 이벤트를 처리할 수 있는 처리 로직이 멍등성 인지 확인합니다. 각 이벤트에 대해 즉각적인 [GetOpportunity](#) 호출을 수행하는 대신 애플리케이션의 요구 사항에 따라 세부 정보를 일괄 처리하거나 선택적으로 가져오는 것이 좋습니다. 중단 없는 작업의 경우 [할당량](#)을 주의하십시오.

낙관적 잠금 구현

낙관적 잠금은 동시 업데이트 중에 의도하지 않은 데이터 재정의의 위험을 방지합니다. 다음은 일반적인 시나리오입니다.

1. 파트너는 CRM 시스템에서 기회를 검색합니다.
2. 사용자는 AWS Partner Central에서 기회를 A 업데이트합니다.
3. 사용자는 CRM 통합을 통해 동일한 기회를 동시에 B 업데이트합니다.
4. 데이터가 변경되면 CRM 시스템은 데이터를 업로드하려고 시도하지만 충돌을 반환합니다. `ConflictException`.
5. 사용자가 오류를 검토하고 충돌하는 데이터를 수동으로 해결합니다.

이 시나리오를 방지하려면 모든 [UpdateOpportunity](#) 요청에 이전 [CreateOpportunity](#), [UpdateOpportunity](#) 및 [GetOpportunity](#) 작업에서 가져올 수 있는 LastModifiedDate 파라미터가 포함되어야 합니다. [UpdateOpportunity](#) 가 시스템과 LastModifiedDate 일치하는 경우에만 업데이트가 성공합니다. 그렇지 않은 경우 [GetOpportunity](#)를 LastModifiedDate 사용하여 최신을 가져오고 업데이트를 다시 시도해야 합니다.

CRM과AWS Partner Central 간의 데이터 동기화

시스템을 Partner Central의 최신 데이터와 동기화하는 것이 중요합니다. 다음은 시스템이 최신 데이터를 반영하도록 하기 위한 두 가지 전략입니다.

이벤트 사용(권장)

1. [ListOpportunities](#)를 사용하여 데이터를 로드합니다.
2. 기회 이벤트를 구독합니다.
3. 새로운 기회나 변화에 대응합니다.
 - Opportunity Created, Opportunity Updated 또는 Engagement Invitation Accepted 이벤트를 수신하면 GetOpportunity를 사용하여 최신 데이터를 가져옵니다.
 - Engagement Invitation Rejected 이벤트를 수신하면 해당 기회를 제거합니다.

ListOpportunities 폴링 사용

1. [ListOpportunities](#)를 사용하여 데이터를 로드합니다.
2. 일일 [읽기 할당량](#)이 소진되지 않도록 폴링 빈도를 선택합니다.
3. LastModifiedDate 저장된 데이터의 최신를 식별하여 데이터가 시작되도록 합니다AWS.
4. [ListOpportunities](#)를 호출할 때 AfterLastModifiedDate 필터의 타임스탬프를 사용합니다.

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

5. AWS는 타임스탬프에 표시된 값 이후에 생성되거나 업데이트된 기회를 반환합니다.
6. 를 사용하여 반환된 모든 페이지를 반복NextToken하고 [GetOpportunity](#)를 사용하여 시스템 데이터를 업데이트합니다.

```
{
  "NextToken": "AAMA-EFRSN...PZa942D",
  "FilterList": [
```

```

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

```

AWS Partner Central Benefits API 사용

AWS파트너 중앙 혜택 API는 파트너가 모든AWS파트너 프로그램에서 혜택을 검색, 신청 및 관리하는 방법을 간소화합니다. 이 서비스는 혜택 관리를 직관적인 단일 인터페이스로 중앙 집중화하여 운영을 간소화하면서 파트너에게 보상을 제공하는 투명한 성능 향상을 만듭니다.

API는 표준AWS API 기능을 제공합니다. API [작업](#)을 사용하거나 프로그래밍 언어 또는 플랫폼에 맞는AWS SDK를 사용하여 액세스합니다. 자세한 내용은 [시작하기AWS](#) 및 [빌드할 도구를 참조하세요 AWS](#).

혜택이란 무엇입니까?

AWS파트너 네트워크(APN)에서 혜택은 파트너가 비즈니스 성장과 고객 성공을 지원하기AWS위해에서 얻을 수 있는 이점 또는 기능을 나타냅니다. 혜택은 파트너가AWS관행을 확장하는 데 도움이 되는 가치적인 가치를 제공하면서 자격과 기여를 기반으로 파트너에게 보상을 제공하도록 설계되었습니다.

이점은 다음과 같은 다양한 형태로 나타날 수 있습니다.

- 금융 인센티브 - 마이그레이션 가속화 프로그램(MAP) 또는 마케팅 개발 자금(MDF)과 같은 크레딧, 현금 지급 및 자금 지원 프로그램
- 액세스 이점 - 새 서비스, 도구 또는 특수 리소스에 대한 액세스 미리 보기
- 인식 - 디지털 배지, 인증, 역량 지정 및 파트너 티어 상태
- 기술 리소스 - 솔루션 아키텍처 지원, 기술 지원 및 ProServe 참여
- 마케팅 지원 - 공동 마케팅 기회, 추천 파트너 상태 및 성공 스토리 간행물

사용 가능한 이점 검색

파트너는 어떤 혜택을 받을 수 있는지 알아보고 자격 요구 사항을 이해하여 혜택 여정을 시작합니다. 혜택 API는 혜택 검색 APIs를 통해 포괄적인 검색 기능을 제공합니다.

사용 가능한 이점 나열

파트너는 ListBenefits API 작업을 사용하여 사용 가능한 모든 이점을 볼 수 있습니다. 이 작업은 파트너가 관련 기회를 빠르게 찾는 데 도움이 되는 선택적 필터링 기능이 있는 이점 카탈로그를 검색합니다.

파트너는 이점을 효율적으로 발견하기 위해 다음을 기준으로 필터링할 수 있습니다.

- 프로그램 - MAP(마이그레이션 가속화 프로그램), MDF(마케팅 개발 자금), ISV 워크로드 마이그레이션, 혁신 샌드박스, 개념 증명 또는 파트너 이니셔티브 자금과 같은 특정 파트너AWS프로그램을 기준으로 필터링
- 이행 유형 - , CREDITS, CASH, DISCOUNT, ACCESS RECOGNITION또는 혜택 제공 방법을 기준으로 필터링 RESOURCE
- 상태 - 혜택 상태를 기준으로 필터링: ACTIVE 또는 DEPRECATED

ListBenefits API는 혜택 ID, 이름, 설명, 관련 프로그램 및 이행 유형과 같은 필수 정보가 포함된 혜택 요약을 반환합니다. 이 요약 보기를 통해 파트너는 사용 가능한 이점을 빠르게 스캔하고 비즈니스 목표와 가장 관련성이 높은 이점을 식별할 수 있습니다.

필터링 접근 방식의 예:

재정적 인센티브에 중점을 둔 파트너는 이행 유형을 기준으로 필터링CREDITS하고 자금 조달 기회를 CASH 확인할 수 있습니다. 기술 지원을 원하는 파트너는 이행 유형별로 필터링RESOURCE하거나 기술 지원 혜택 및 도구 액세스를 ACCESS 찾을 수 있습니다.

세부 혜택 정보 보기

파트너가 관심 있는 이점을 식별하면 GetBenefit API 작업을 사용하여 전체 세부 정보를 검색할 수 있습니다. 이는 다음을 포함한 포괄적인 정보를 제공합니다.

- 자격 기준 - 파트너가 혜택을 받기 위해 충족해야 하는 세부 요구 사항
- 혜택 요청 스키마 - 신청 시 파트너가 제공해야 하는 특정 정보 및 설명서
- 권한 부여 세부 정보 - 파트너가 승인 시 받게 될 내용
- 관련 프로그램 - 혜택이 지원하는AWS파트너 프로그램

혜택 요청 스키마는 혜택 애플리케이션의 구조와 필수 필드를 정의하므로 특히 중요합니다. 파트너는 혜택 애플리케이션을 생성하기 전에이 스키마를 주의 깊게 검토하여 필요한 모든 정보를 미리 수집해야 합니다.

⚠ Important

각 혜택에 대한 자격 결정은 클라이언트 또는 UI 계층에서 처리합니다. 자금 지원 관련 혜택의 경우 자격은 일반적으로 파트너의 스코어카드 성과 및 프로그램 참여를 기반으로 합니다.

주제

- [혜택 애플리케이션 작업](#)
- [혜택 할당 작업](#)

혜택 애플리케이션 작업

혜택 애플리케이션은 특정 혜택에 대한 파트너의 요청을 모델링합니다. 혜택별 조건에 따라 요청을 평가하고 처리하는 데 필요한 모든 정보를 캡처합니다. 혜택 애플리케이션 수명 주기는 초안 생성부터 최종 승인 또는 거부까지 여러 상태로 진행됩니다.

혜택 애플리케이션 생성

파트너는 CreateBenefitApplication API 작업을 사용하여 혜택 애플리케이션을 생성하여 혜택 요청 프로세스를 시작합니다. 생성 시 애플리케이션이 PENDING_SUBMISSION 상태가 되어 파트너가 검토를 위해 제출하기 전에 전체 정보를 준비할 수 있습니다.

혜택 애플리케이션을 생성할 때 파트너는 다음을 제공해야 합니다.

- 혜택 식별자 - 요청 중인 혜택의 ID 또는 ARN
- 이행 유형 - 파트너가 혜택을 받을 것으로 예상하는 방식(CREDITS, CASH, DISCOUNT, ACCESS, RECOGNITION 또는 RESOURCE)
- 혜택 애플리케이션 세부 정보 - 혜택의 요청 스키마에 정의된 혜택별 정보가 포함된 JSON 문서
- 클라이언트 토큰 - 중복 제출을 방지하기 위한 고유한 맥동성 토큰

파트너는 선택적으로 다음을 제공할 수 있습니다.

- 이름 및 설명 - 애플리케이션의 사람이 읽을 수 있는 식별자
- 파트너 연락처 -이 혜택 요청을 관리하는 사람의 연락처 정보(최대 1개의 연락처)
- 파일 첨부 - 프로젝트 계획, SOWs, 비용 증명 또는 고객 만족도 설문조사와 같은 지원 문서(최대 10개의 파일)

- 관련 리소스 - 관련 기회 또는 기존 혜택 할당에 대한 링크(최대 10개의 리소스)
- 태그 - 리소스 구성 및 추적을 위한 키-값 페어(최대 200개의 태그)

CreateBenefitApplication API는 소프트 검증을 수행하여 기본 필드 유형 및 패턴만 확인합니다. 전체 비즈니스 로직 검증은 제출 중에 수행되므로 파트너는 불완전한 애플리케이션을 저장하고 나중에 다시 돌아와 완료할 수 있습니다.

모범 사례: 파트너는의 혜택 요청 스키마GetBenefit를 사용하여 애플리케이션을 생성하기 전에 필요한 정보를 정확하게 이해해야 합니다. 이렇게 하면 누락되거나 잘못된 데이터로 인해 제출이 실패할 가능성이 줄어듭니다.

혜택 애플리케이션 업데이트

파트너는 UpdateBenefitApplication API 작업을 사용하여 초안 혜택 애플리케이션을 수정할 수 있습니다. 이를 통해 파트너는 제출 전에 정보를 구체화하거나, 설명서를 추가하거나, 오류를 수정할 수 있습니다.

혜택 애플리케이션을 업데이트할 때 파트너는 다음을 제공해야 합니다.

- 식별자 - 업데이트할 혜택 애플리케이션의 ID 또는 ARN입니다.
- 개정 - 낙관적 잠금을 위한 현재 개정 번호
- 혜택 애플리케이션 세부 정보 - 완전하고 업데이트된 JSON 문서

파트너는 특정 필드만 변경되더라도 전체 혜택 애플리케이션 객체를 제출해야 합니다.

모범 사례는 먼저를 사용하여 최신 애플리케이션 세부 정보를 검색하고 필요한 필드를 GetBenefitApplication수정한 다음 업데이트된 전체 페이로드를에 제출하는 것입니다UpdateBenefitApplication.

낙관적 잠금: 개정 필드는 애플리케이션이 마지막으로 검색된 이후 변경되지 않은 경우에만 업데이트가 적용되도록 합니다. 개정이 현재 데이터베이스 값과 일치하지 않으면 충돌 오류와 함께 업데이트가 거부됩니다. 이렇게 하면 파트너가 실수로 다른 프로세스 또는 시스템의 변경 사항을 덮어쓰지 않습니다.

업데이트는 애플리케이션이 PENDING_SUBMISSION 상태인 동안에만 수행할 수 있습니다. 제출된 애플리케이션은 검토 프로세스에 들어가며이 API를 통해 더 이상 업데이트할 수 없습니다.

리소스와 혜택 애플리케이션 연결

파트너는 AssociateBenefitApplicationResource API 작업을 사용하여 관련 리소스에 혜택 애플리케이션을 연결할 수 있습니다. 이렇게 하면 이점과 사용 중인 비즈니스 컨텍스트 간에 중요한 연결이 생성됩니다.

지원되는 리소스 유형은 다음과 같습니다.

- 기회 - 혜택 애플리케이션을의 특정 고객 기회에 연결합니다. 이는 고객 참여와 관련된 MAP 자금 또는 POC 크레딧과 같은 기회별 이점에 특히 유용합니다.
- BENEFIT_ALLOCATION - 기존 혜택 할당에 연결하여 파트너가 혜택을 연결하거나 이전 혜택을 활용하는 방법을 시연할 수 있습니다.

리소스 연결은 제출 전에만 수행할 수 있습니다. 혜택 애플리케이션이 제출되면(상태가 로 변경됨IN_REVIEW) 추가 연결은 허용되지 않습니다. 파트너는 각 혜택 애플리케이션에 최대 10개의 리소스를 연결할 수 있습니다.

파트너는 리소스를 연결 해제하기 위해 DisassociateBenefitApplicationResource API 작업을 사용합니다. 연결과 마찬가지로 연결 해제는 PENDING_SUBMISSION 상태에서에서만 발생할 수 있습니다.

Important

파트너는 혜택 애플리케이션에 액세스하고 연결 중인 특정 리소스를 읽을 수 있는 적절한 권한이 있어야 합니다. API는 연결 작업 중에 이러한 권한을 검증합니다.

혜택 애플리케이션 제출

혜택 애플리케이션이 완료되고AWS검토할 준비가 되면 파트너는 SubmitBenefitApplication API 작업을 사용하여 제출합니다. 제출은에서 로 상태 전환을 트리거PENDING_SUBMISSIONIN_REVIEW하고 혜택별 승인 워크플로를 시작합니다.

제출 시 API는 다음을 포함한 포괄적인 검증을 수행합니다.

- 필수 필드 검증 - 혜택 애플리케이션 세부 정보의 모든 필수 필드가 있는지 확인합니다.
- 필드 형식 검증 - 필드 값이 예상 패턴 및 데이터 유형과 일치하는지 확인합니다.
- 비즈니스 규칙 검증 - 혜택별 비즈니스 로직 및 제약 조건 적용

- 리소스 검증 - 연결된 모든 리소스가 유효하고 액세스 가능한지 확인합니다.
- 파일 검증 - 모든 파일 첨부 파일이 처리를 성공적으로 완료했는지 확인합니다.

검증에 실패하면 API는 수정이 필요한 필드를 나타내는 자세한 오류 코드 및 메시지와 함께 `ValidationException`을 반환합니다. 파트너는 제출을 다시 시도 `UpdateBenefitApplication`하기 전에 사용하여 이러한 문제를 해결해야 합니다.

파일 처리 요구 사항: 연결된 파일이 아직 `PENDING` 상태인 경우 혜택 애플리케이션을 제출할 수 없습니다. 파트너는 제출하기 전에 파일 처리가 완료될 때까지 기다려야 합니다(에 대한 상태 변경 `SUCCEEDED`). 파일 처리에 실패하면(상태 `FAILED`) 파트너는 수정된 버전을 업로드해야 합니다.

성공적으로 제출한 후:

- 애플리케이션 상태가 `IN_REVIEW` 로 변경됩니다.
- 혜택 소유자의 팀에 새 제출에 대한 알림이 전송됩니다.
- 파트너는 더 이상 표준 업데이트 APIs
- 애플리케이션이 비즈니스 승인, 기술 승인 및 재무 승인 단계를 포함할 수 있는 정의된 승인 워크플로에 들어갑니다.

파트너는 애플리케이션을 검색하고 현재 승인 단계를 나타내는 단계 필드(예: "비즈니스 승인", "기술 승인", "재무 승인")를 모니터링하여 제출 진행 상황을 추적할 수 있습니다.

제출된 애플리케이션 관리

제출된 파트너는 혜택 애플리케이션을 관리할 수 있는 옵션이 제한적이지만 API는 일반적인 시나리오에 대한 특정 작업을 제공합니다.

혜택 애플리케이션 리콜

파트너가 오류를 발견하거나 제출 후 변경해야 하는 경우 `RecallBenefitApplication` API 작업을 사용하여 애플리케이션을 리콜할 수 있습니다. 리콜은 애플리케이션을 `PENDING_SUBMISSION` 상태로 다시 전환하여 업데이트 및 재제출을 허용합니다.

애플리케이션을 리콜할 때 파트너는 다음을 제공해야 합니다.

- 식별자 - 리콜할 혜택 애플리케이션의 ID 또는 ARN입니다.
- 이유 - 리콜에 대한 선택적 설명(최대 1,000자)

사유 필드를 사용하면 추적 가능성을 높이고 리콜로 이어지는 일반적인 문제를 AWS 이해하여 향후 혜택 애플리케이션 프로세스를 개선할 수 있습니다.

리콜 후 파트너는 UpdateBenefitApplication를 사용하여 필요한 변경을 수행한 다음 준비가 SubmitBenefitApplication 되면 이를 사용하여 다시 제출할 수 있습니다.

Important

리콜은 일반적으로 초기 검토 단계에서만 사용할 수 있습니다. 이후 승인 단계로 진행했거나 승인된 애플리케이션은 리콜 대상이 될 수 없습니다.

혜택 애플리케이션 수정

제출 후 사소한 수정의 경우 파트너는 AmendBenefitApplication API 작업을 사용하여 전체 애플리케이션을 재현하지 않고 특정 필드를 업데이트할 수 있습니다. 이는 AWS 검토자가 검토 프로세스 중에 설명 또는 수정을 요청할 때 특히 유용합니다.

수정은 파트너가 다음을 지정하는 JSON 패치 스타일 접근 방식을 사용합니다.

- 경로 - 업데이트할 필드를 식별하는 JSONPath 표현식(예: `$.CreditDisbursementDetails.AwsAccountIdForCredits`)
- 값 - 필드의 새 값입니다.
- 작업 - 수행할 작업(현재 REPLACE는 만 지원됨)

파트너는 단일 API 호출에서 최대 10개의 수정 사항을 제출할 수 있습니다. 각 수정에는 낙관적 잠금을 위한 업데이트된 개정 번호가 포함되어야 합니다.

사소한 수정에는 수정을 사용해야 합니다. 상당한 변경의 경우 파트너는 RecallBenefitApplication를 사용하여 포괄적인 업데이트를 위해 애플리케이션을 초안 상태로 되돌려야 합니다.

혜택 애플리케이션 취소

파트너가 더 이상 도움이 필요하지 않거나 애플리케이션을 철회하려는 경우 CancelBenefitApplication API 작업을 사용하여 취소할 수 있습니다. 취소하면 애플리케이션이 CANCELED 상태로 전환되어 애플리케이션 프로세스가 영구적으로 종료됩니다.

애플리케이션을 취소할 때 파트너는 다음을 제공해야 합니다.

- 식별자 - 취소할 혜택 애플리케이션의 ID 또는 ARN입니다.
- 이유 - 취소에 대한 선택적 설명(최대 1,000자)

취소되면 애플리케이션을 다시 활성화할 수 없습니다. 파트너가 나중에 혜택이 필요하다고 결정하는 경우 새 혜택 애플리케이션을 생성해야 합니다.

취소의 일반적인 이유는 다음과 같습니다.

- 고객 기회가 손실되거나 지연됨
- 파트너 용량 제약 조건 변경됨
- 비즈니스 우선 순위 이동
- 의도한 목적에 더 이상 혜택이 필요하지 않음

혜택 애플리케이션 세부 정보 보기

파트너는 GetBenefitApplication API 작업을 사용하여 혜택 애플리케이션에 대한 전체 정보를 검색할 수 있습니다. 이를 통해 다음과 같은 애플리케이션을 포괄적으로 볼 수 있습니다.

애플리케이션 메타데이터:

- 고유 식별자(ID) 및 Amazon 리소스 이름(ARN)
- 관련 혜택 식별자
- 애플리케이션 이름 및 설명
- 현재 상태(PENDING_SUBMISSION, IN_REVIEW, ACTION_REQUIRED, APPROVEDREJECTED, 또는 CANCELED)
- 현재 처리 단계

상태 정보:

- 상태 이유 - 현재 상태에 대한 사람이 읽을 수 있는 설명
- 상태 이유 코드 - 특정 문제 또는 요구 사항을 나타내는 구조화된 코드(예: "Checklist Incomplete", "Project Plan Missing", "Design Win Missing")

애플리케이션 콘텐츠:

- 전체 혜택 애플리케이션 세부 정보 JSON 문서

- 파트너 연락처 정보
- 처리 상태의 파일 첨부 파일
- 연결된 리소스(기회 또는 할당)
- 리소스 태그

감사 정보:

- 생성 타임스탬프
- 마지막 수정 타임스탬프
- 현재 개정 번호

상태 이유 코드는 애플리케이션이 ACTION_REQUIRED 상태일 때 특히 유용합니다. 이러한 코드는 파트너가 애플리케이션을 진행하기 위해 해결해야 하는 사항에 대한 구체적이고 실행 가능한 지침을 제공합니다.

혜택 애플리케이션 나열

파트너는 ListBenefitApplications API 작업을 사용하여 모든 혜택 애플리케이션을 볼 수 있습니다. 이렇게 하면 강력한 필터링 기능이 포함된 애플리케이션 요약의 페이지가 매겨진 목록이 반환됩니다.

파트너는 다음을 기준으로 애플리케이션을 필터링할 수 있습니다.

- 프로그램 - 특정 프로그램(MAP, MDF, 샌드박스, POC 등)에 대한 애플리케이션 보기
- 이행 유형 - 전송 방법(CREDITS, CASHACCESS, 등)을 기준으로 필터링
- 혜택 식별자 - 특정 혜택에 대한 애플리케이션 보기
- 상태 - 애플리케이션 상태(PENDING_SUBMISSION, , IN_REVIEW, APPROVEDREJECTED, CANCELED)를 기준으로 필터링
- 단계 - 승인 단계(비즈니스 승인, 기술 승인, 재무 승인)를 기준으로 필터링
- 연결된 리소스 ARNs - 특정 기회 또는 할당에 연결된 애플리케이션 찾기

목록 응답에는 다음과 같은 필수 요약 정보가 포함됩니다.

- 애플리케이션 ID, ARN 및 이름
- 관련 혜택 ID

- 프로그램 및 이행 유형
- 현재 상태 및 단계
- 타임스탬프 생성 및 수정
- 관련 리소스
- 현재 개정 번호
- 혜택 애플리케이션 세부 정보에서 선택한 필드(복리후생 소유자가 정의)

파트너는 생성 날짜, 상태, 프로그램 또는 혜택 식별자를 기준으로 오름차순 또는 내림차순으로 정렬하는 옵션과 함께 정렬 파라미터를 사용하여 결과 정렬을 구성할 수 있습니다.

대시보드 구축: ListBenefitApplications API는 파트너 대시보드 생성을 지원하도록 설계되었습니다. 파트너는 애플리케이션을 필터링하고 정렬하여 다음과 같은 보기를 구축할 수 있습니다.

- 작업이 필요한 애플리케이션(ACTION_REQUIRED 상태)
- 최근에 제출된 애플리케이션(생성 날짜, IN_REVIEW 상태별로 정렬됨)
- 할당 보류 중인 승인된 애플리케이션(APPROVED 상태)
- 프로그램별 애플리케이션(특정 프로그램으로 필터링됨)

혜택 할당 작업

혜택 할당은 파트너에게 부여된 혜택을 나타냅니다. 혜택 애플리케이션이 승인되면 파트너에게 실제 크레딧, 자금, 액세스 또는 기타 혜택 이행을 제공하는 하나 이상의 혜택 할당을 AWS 생성합니다.

혜택 할당 이해

혜택 할당은 다양한 유형의 이행을 나타낼 수 있습니다.

- 금융 결제(CASH) - 결제 추적을 통해 파트너에게 직접 금융 결제
- AWS 크레딧(CREDITS) - 사용량 추적이 있는 AWS 계정에 적용되는 크레딧 코드
- 사용 가능한 할당 - 사용률 추적이 포함된 사전 승인된 자금 금액 또는 지급
- Access Grants(ACCESS) - 시스템, 도구 또는 리소스에 대한 액세스
- 인식(인식) - 디지털 배지, 인증 또는 상태 지정
- 리소스(리소스) - 기술 지원, 자문 서비스 또는 ProServe 참여

각 혜택 할당에는 상태 필드를 통해 추적되는 수명 주기가 있습니다.

- ACTIVE - 할당을 사용할 수 있습니다.
- 비활성 - 할당을 일시적으로 사용할 수 없음
- FULFILLED - 할당이 완전히 사용되거나 전달되었습니다.

또한 혜택 할당에는 혜택이 적용되는 시점(StartsAt)과 만료 시점(ExpiresAt)을 정의하는 시간 정보가 포함되어 있으므로 파트너는 혜택을 활용하는 기간을 이해할 수 있습니다.

혜택 할당 나열

파트너는 ListBenefitAllocations API 작업을 사용하여 모든 혜택 할당을 볼 수 있습니다. 그러면 포괄적인 필터링 기능이 포함된 할당 요약의 페이지가 매겨진 목록이 반환됩니다.

파트너는 다음을 기준으로 할당을 필터링할 수 있습니다.

- 혜택 식별자 - 특정 혜택에 대한 할당 보기
- 혜택 애플리케이션 식별자 - 특정 애플리케이션의 할당 보기
- 이행 유형 - 할당 유형(CREDITS, CASHACCESS, 등)을 기준으로 필터링
- 상태 - 할당 상태(ACTIVE, INACTIVE, FULFILLED)를 기준으로 필터링

목록 응답에는 다음이 포함된 할당 요약이 포함됩니다.

- 할당 ID, ARN 및 이름
- 관련 혜택 ID 및 혜택 애플리케이션 ID(해당하는 경우)
- 이행 유형
- 현재 상태
- 생성 타임스탬프
- 시작일

이 목록 보기를 통해 파트너는 할당 추적 대시보드를 구축하고 다음을 식별할 수 있습니다.

- 즉시 사용할 수 있는 활성 할당
- 만료에 가까워지는 할당
- 기록 추적을 위한 이행된 할당
- 프로그램 또는 이행 유형별 총 할당 값

세부 할당 정보 보기

파트너는 GetBenefitAllocation API 작업을 사용하여 전체 할당 세부 정보를 검색할 수 있습니다. 이는 할당 유형에 따라 달라지는 이행별 세부 정보를 포함하여 할당에 대한 포괄적인 정보를 제공합니다.

코어 할당 정보:

- 고유 식별자(ID) 및 Amazon 리소스 이름(ARN)
- 할당 이름 및 상태
- 상태 이유(현재 상태 설명)
- 관련 혜택 ID 및 혜택 애플리케이션 ID
- 이행 유형
- 생성, 업데이트, 시작 및 만료 타임스탬프

이행 세부 정보(유형별):

FulfillmentDetail 필드에는 이행 유형에 따라 다양한 정보 구조가 포함되어 있습니다.

현금 지급 세부 정보(CASH 유형)

직접 현금 결제의 경우 이행 세부 정보는 다음과 같습니다.

- 지급 금액 - 금액 값 및 통화 코드를 포함하여 지급된 총 자금 금액입니다.
- 발급 세부 정보(해당하는 경우) - 구매 주문 또는 발급 이벤트에 대한 정보:
 - 발급 ID - 지급의 고유 식별자
 - 발행 금액 - 현지 통화 금액
 - 발행 시점 - 지급 타임스탬프

이 구조는 단일 할당에 대해 여러 발급이 발생할 수 있는 단일 지출 및 사전 승인된 자금 조달 시나리오를 모두 지원합니다.

크레딧 세부 정보(CREDITS 유형)

AWS크레딧의 경우 이행 세부 정보는 다음과 같습니다.

- 할당된 금액 - 할당된 총 크레딧 금액

- 발행 금액 - 발행된 총 크레딧 금액(단계적 발행에 할당된 것보다 적을 수 있음)
- 크레딧 코드 - 개별 크레딧 코드 세부 정보의 배열:
 - AWS계정 ID - 크레딧이 적용되는 계정
 - 값 - 크레딧 코드의 통화 값
 - AWS크레딧 코드 - 실제 크레딧 코드 문자열
 - 상태 - 크레딧 코드 상태(ACTIVE, INACTIVE, FULFILLED)
 - 발행 시점 - 크레딧 코드가 발행된 시점
 - 만료 시간 - 크레딧 코드가 만료되는 경우

이 세부 크레딧 추적을 통해 파트너는 여러 계정의 크레딧 사용률을 모니터링하고 사용 가능하거나 부분적으로 사용되거나 완전히 사용된 크레딧을 이해할 수 있습니다.

소비자 할당 세부 정보(미리 승인된 금액/월릿)

사전 승인된 자금 풀의 경우 이행 세부 정보에는 다음이 포함됩니다.

- 할당 금액 - 할당의 총 금액
- 남은 금액 - 미사용 금액은 여전히 사용 가능
- 사용 금액 - 이미 사용된 금액
- 발급 세부 정보(해당하는 경우) - 할당을 위한 구매 주문 정보

사용 가능한 할당을 통해 파트너는 남은 잔액을 실시간으로 추적하여 적격 활동에 필요한 만큼 자금을 회수할 수 있습니다.

액세스 세부 정보(ACCESS 유형)

액세스 권한 부여의 경우 이행 세부 정보에는 다음이 포함됩니다.

- 설명 - 부여되는 액세스 및 액세스 사용 방법에 대한 자세한 설명

액세스 할당은 미리 보기 서비스, 특수 도구 또는 제한된 리소스에 대한 액세스를 제공할 수 있습니다. 설명 필드는 파트너가 부여된 액세스를 활성화하고 사용하는 방법에 대한 지침을 제공합니다.

새 혜택 애플리케이션에 할당 적용

혜택 할당의 `ApplicableBenefitIds` 필드는 이 할당을 활용할 수 있는 다른 이점을 식별합니다. 이를 통해 파트너가 기존 할당을 사용하여 추가 혜택을 신청할 수 있는 혜택 체인이 가능합니다.

예를 들어 파트너는 다음을 수행할 수 있습니다.

1. 사전 승인된 MAP 자금 할당 받기
2. 개별 고객 마이그레이션 프로젝트를 위한 혜택 애플리케이션 생성
3. 이러한 애플리케이션을 사전 승인된 할당과 연결
4. 프로젝트가 승인되면 할당에서 자금을 뺍습니다.

이 접근 방식은 사전 승인된 자금이 있는 파트너의 혜택 애플리케이션 프로세스를 간소화하여 개별 프로젝트 요청에 대한 검토 주기를 줄입니다.

AWS파트너 중앙 채널 API 사용

AWS파트너 중앙 채널 APIs 사용하면 승인된AWS재판매 비즈니스를 프로그래밍 방식으로 관리할 수 있습니다. 이러한 APIs 사용하면AWS솔루션 공급자,AWS배포자 및AWS배포 판매자가 다음을 수행할 수 있습니다.

- 재판매 프로그램에 사용되는AWS관리 계정을 Partner Central에 보고
- AWS계정에 초대장을 보내 파트너 연결 수락
- 재판매된 최종 고객AWS계정을 온보딩하고 파트너 할인을 적용하는 데 필요한 관계 생성

AWS파트너 중앙 채널 APIs 사용하면 다음을 위한 사용자 지정 자동화를 구축할 수 있습니다.

- 새 파트너 소유AWS계정을 APN 재판매 프로그램에 온보딩
- 새로운 최종 고객 계정을 재판매 프로그램에 온보딩
- 고객 온보딩 및 할인 검증 가속화

프로그램 관리 계정(PMAs) 작업

프로그램 관리 계정(PMA)은AWS관리 계정을 채널 프로그램 등록과 연결합니다. 프로그램 관리 계정은 채널 핸드셰이크를 사용하여 셀프 서비스를 활성화합니다. PMA를 생성하고 활성화하면 채널 프로그램 혜택 및 할인이 연결된AWS관리 계정으로 전달되는 모든 인보이스에 적용됩니다. 다음 APIs 사용하여 재판매 및 채널 할인을 처리하는AWS계정을 보고하고 관리합니다.

- CreateProgramManagementAccount: 고객 결제 관리를 위한 새 계정 추가
- UpdateProgramManagementAccount: 기존 계정 수정

- DeleteProgramManagementAccount: 재판매 프로그램에서 계정 제거
- ListProgramManagementAccounts: 기존 계정 및 해당 상태 보기

채널 핸드셰이크 작업

채널 핸드셰이크를 사용하면 중요한 채널 관리 작업에 대한 안전하고 상호적인 동의를 얻을 수 있습니다. AWS Partner Central은 채널 핸드셰이크를 사용하여 프로그램AWS관리 계정(PMAs) 활성화, 서비스 기간 설정, 서비스 기간 조기 종료 등 세 가지 목적으로 관리 계정의 동의를 확인합니다. 다음과 같은 세 가지 유형이 있습니다.

- PROGRAM_MANAGEMENT_ACCOUNT: 프로그램 관리 계정 핸드셰이크는 채널 프로그램 혜택을 적용하기 위해 PMAs를 활성화할 때 동의를 확인합니다.
- START_SERVICE_PERIOD: 서비스 기간 생성 핸드셰이크는 최소 공지 기간 또는 고정 기간으로 결제 전송 약정에 대한 상호 계약을 설정합니다.
- REVOKE_SERVICE_PERIOD: 서비스 기간 종료 핸드셰이크를 사용하면 양 당사자가 동의할 때 활성 서비스 기간을 조기에 종료할 수 있습니다. 모든 핸드셰이크 유형을 적용하려면 대상AWS관리 계정에서 수락해야 합니다.

다음 APIs 사용하여 PMA 활성화 및 서비스 기간 관리를 위한 채널 핸드셰이크를 관리합니다.

- CreateChannelHandshake: PMA 초대, 서비스 기간 설정 또는 서비스 기간 종료를 위한 핸드셰이크 생성
- AcceptChannelHandshake: 대상AWS계정의 핸드셰이크 수락(핸드셰이크 유형에 따라 파트너 또는 고객이 사용할 수 있음)
- RejectChannelHandshake: 대상AWS계정에서 핸드셰이크 거부
- CancelChannelHandshake: 수락, 거부 또는 만료되기 전에 보류 중인 핸드셰이크 취소
- ListChannelHandshakes: 유형 및 상태별로 필터링하여 핸드셰이크 추적 및 보기

채널 관계 작업

관계를 통해 최종 고객, 내부 조직 또는 다운스트림 판매자를 관리할 수 있습니다. 각 관계에는 다음이 포함됩니다.

- 연결된AWS조직의AWS관리 계정
- 채널 할인 검증에 필요한AWS파트너 네트워크 메타데이터

관계를 생성하면 연결된 AWS 조직의 모든 사용량이 관련 채널 프로그램 혜택을 받습니다. 다음 APIs 사용하여 판매자와 최종 고객을 보고합니다.

- CreateRelationship: 신규 최종 고객, 배포 판매자 또는 내부 AWS 계정 온보딩
- GetRelationship: 특정 관계 세부 정보 보기
- ListRelationships: 모든 관계 보기
- UpdateRelationship: 기존 관계 수정
- DeleteRelationship: 관계 제거

수익 측정 API 사용

주제

- [수익 속성 IDs 작업](#)
- [Marketplace 수익 공유 작업](#)

수익 속성 IDs 작업

수익 속성 ID는 AWS Marketplace 제품 수익을 특정 AWS Marketplace 제안 및/또는 AWS 파트너 중앙 기회에 매핑하는 거래 수준 식별자입니다. 제품 수준 파트너 수익 측정(PRM) 구현을 기반으로 구축됩니다. PRM은 Marketplace 제품의 총 귀속 수익을 캡처하고, 수익 속성 ID는 지정한 월별 비용 할당 백분율에 따라 해당 수익을 특정 거래에 매핑합니다.

수익 속성 서비스는 수익 속성 IDs와 월별 비용 할당 항목을 비동기적으로 관리하기 위해 APIs를 노출합니다.

수익 속성 ID란 무엇입니까?

수익 속성 ID에는 두 개의 계층이 있습니다.

- ra- 접두사 ID(예: ra-aabbccdde001) 및 ARN으로 식별되는 파트너 이름이 지정된 파트너 설명 리소스의 속성입니다. 각 어트리뷰션의 범위는 Catalog (AWS 프로덕션용, 테스트 Sandbox용) 및 파트너의 계정으로 지정됩니다.
- 각 항목마다 하나 이상의 월별 비용 할당 항목이 단일(제안 ID 또는 기회 ID, 결제 월) 조합을 비용 할당 백분율에 매핑합니다. 항목은 할당 작업 패턴을 통해 비동기적으로 배치로 관리됩니다.

수익 속성 ID는 두 가지 방법으로 사용할 수 있습니다.

- 기존 제품 수준 PRM 구현에 대한 거래 수준 오버레이입니다. 수익 속성 ID를 생성하고, 해당 Marketplace 제안 및/또는 ACE 기회를 연결하고, 이미 측정된 제품 수익을 특정 거래에 AWS 매핑합니다. 기존 리소스 태그 또는 사용자 에이전트 문자열을 변경할 필요가 없습니다.
- 리소스 태그 값(aws-apn-id = <RA ID>) 또는 사용자 에이전트 문자열(APN_1.1/pc_<RA ID>\$)로 직접 사용되는 독립 실행형 식별자로, 처음부터 특정 거래에 대한 소비 속성을 지정합니다 AWS.

수익 속성 IDs 작업

파트너는 수익 속성 서비스를 통해 수익 속성 IDs와 월별 비용 할당 항목을 관리할 수 있습니다. 수명 주기는 어트리뷰션 레코드 흐름(생성, 업데이트, 검색, 나열)과 할당 흐름(배치 작업 시작, 결과 폴링, 개별 항목 검색, 어트리뷰션 항목 나열)의 두 가지 독립 흐름을 통해 진행됩니다.

수익 속성 ID 생성

첫 번째 단계는 CreateRevenueAttribution API 작업을 사용하여 수익 속성 ID를 생성하는 것입니다. 반환된 Id 및 리소스 태그 값으로 즉시 사용하거나 사용자 에이전트 문자열에서 AWS 사용 속성을 지정할 Arn 수 있습니다.

수익 속성 ID를 생성할 때 파트너는 다음을 제공해야 합니다.

- **Catalog** - AWS 프로덕션용 또는 테스트 Sandbox용.
- **Name** - 사람이 읽을 수 있는 속성 이름입니다. 카탈로그 및 파트너의 계정 내에서 고유해야 합니다. 최대 128자입니다.

파트너는 선택적으로 다음을 제공할 수 있습니다.

- **Description** - 속성에 대한 자유 텍스트 설명입니다. 최대 1,024자입니다.
- **MarketplaceProduct** - AWS이 속성과 연결할 Marketplace 제품입니다. ProductIdentifier (25자 Marketplace 제품 ID) 및 TenancyMode1 (MULTI_TENANT 또는)를 입력합니다 SINGLE_TENANT. 생성 시 생략하면 고객이 구매한 Marketplace 제품에 관계없이 수익 속성 ID로 측정된 모든 소비에 속성이 적용됩니다. 이는 단일 배포가 둘 이상의 Marketplace 목록에 걸쳐 있을 때 유용합니다.
- **Tags** - 리소스 조직을 위한 최대 200개의 키-값 페어. 태그 키는 요청 내에서 고유해야 합니다.

모범 사례: 거래가 특정 Marketplace 목록에 고정될 MarketplaceProduct.ProductIdentifier 때마다 제공합니다. 이를 통해는 호출하는 파트너 계정 또는 파트너 계정 연결(PAC)을 통해 연결된 자 회사 계정이 제품을 소유하고 있는지 AWS 확인하고 응답에 확인된 ProductCode 및 ProductType

(예: SaaS, AMI, ML)를 표시할 수 있습니다. 승인된 계정이 제품을 소유하지 않은 경우 API는 이유와 `ValidationException` 함께 반환합니다 `PRODUCT_NOT_FOUND_OR_NOT_OWNED`.

응답은 새 Id (형식 `ra-[a-z0-9]{13}`), , Arn 확인된 `MarketplaceProduct` 속성 및 초기 Version 숫자(1부터 시작하여 각 후속 업데이트 시 증분)를 반환합니다.

월별 비용 할당 항목 추가

수익 속성 ID가 생성되면 파트너는 `StartRevenueAttributionAllocationsTask` API 작업을 통해 월별 비용 할당 항목 배치를 제출하여 AWS Marketplace 제안 및/또는 ACE 기회를 여기에 연결합니다. 이는 작업당 최대 250개의 할당 변경(CREATE 및/또는 UPDATE)을 허용하는 비동기 작업입니다.

각 할당 항목은 다음을 지정합니다.

- 제안 ID 또는 기회 ID - 연결 중인 거래의 고유 식별자입니다. Marketplace 제안이 이미 ACE 기회에 연결되어 있는 경우 파트너는 제안 ID만 제공하면 됩니다. 그러면 연결된 기회에 AWS 자동으로 수익이 귀속됩니다.
- 결제 월 - 비용 할당이 적용되는 역월입니다(예: 2026-04).
- 비용 할당 % -이 결제 월의이 거래로 인한 제품 총 AWS 소비의 부분입니다. 고객이 인프라를 공유하는 하이브리드 배포를 위한 파트너 호스팅 구성 요소를 포함한 다중 테넌트 SaaS 제품에 필요합니다.
- 고객 AWS 계정 ID - 제품을 사용하는 고객의 AWS 계정입니다. 멀티 테넌트 SaaS 제품에 필요합니다.

지정된 수익 속성 ID 및 동일한 결제 월의 모든 할당 항목에 대한 총 비용 할당 %는 100%를 초과해서는 안 됩니다. 합계가 100%를 초과하면 레코드당 오류 코드로 비동기 비즈니스 검증 중에 문제가 되는 항목이 거부됩니다.

동기 검증: 기본 세이프 검증을 동기적으로 `StartRevenueAttributionAllocationsTask` 수행합니다(필드 유형, 패턴, 배치 크기). 기본 검증이 통과하면 API는 상태가 `TaskId` 인를 반환합니다 `IN_PROGRESS`. 비즈니스 검증(캡 검사, 불변성 규칙, Marketplace 및 ACE에 대한 종속성 조회)은 비동기적으로 실행되며 레코드별 결과를 통해 검색됩니다 `GetRevenueAttributionAllocationsTask`.

제출된 작업을 모니터링하기 위해 파트너는 수익 속성 리소스 ID 또는 `ARNGetRevenueAttributionAllocationsTask`으로 폴링합니다. 응답은에서 `IN_PROGRESS`로 진행되고 `COMPLETED`(개별 레코드의 성공 또는 실패 여부에 관계없이) 다음을 반환합니다.

- **TaskStatus** - `IN_PROGRESS`, `COMPLETED` 또는 `FAILED` (최상위 작업 실패).

- **RecordResults** - 각 입력 레코드에 대해 할당된 AllocationId (성공한 경우), 레코드당 상태 (SUCCEEDED 또는 FAILED), 레코드가 실패한 경우 구조화된 오류 코드 및 메시지.

파트너는 작업 결과를 즉시 검색해야 합니다. 작업이 완료되면 레코드별 결과를 조정하고 실패한 항목을 수정하여 새 작업에 다시 제출할 수 있습니다.

월별 비용 할당 항목 편집

비용 할당 항목은 다음 규칙을 사용하여 결제 월별로 관리됩니다.

- 파트너는 언제든지 현재 또는 향후 결제 월의 새 월별 항목을 추가할 수 있습니다.
- 파트너는 언제든지 현재 또는 향후 결제 월의 월별 항목을 업데이트할 수 있습니다.
- 파트너는 이번 달 7일까지 지난 달의 항목을 업데이트할 수 있습니다. 이 기간을 통해 파트너는 전월 할당을 완료하기 전에 AWS Cost Explorer에서 실제 사용량을 검토할 수 있습니다.

이번 달 7일 이후에는 이전 결제 월의 어트리뷰션을 더 이상 수정할 수 없습니다. 현재 또는 향후 달 항목에 대한 업데이트는 다음 월별 결제 주기부터 적용됩니다. 이전 달의 과거 속성은 소급하여 다시 계산되지 않습니다.

할당 항목에 UPDATE 대한 업데이트는 영향을 받는 각 항목에 대해 0 operation 설정된 동일한 StartRevenueAttributionAllocationsTask API 작업을 통해 제출됩니다. 비동기 작업 패턴은 업데이트 및 생성을 균일하게 처리합니다.

수익 속성 ID 업데이트

파트너는 UpdateRevenueAttribution API 작업을 사용하여 기존 수익 속성 IDDescription의를 업데이트할 수 있습니다. 어트리뷰션 레코드 자체는 변경할 수 없습니다. 생성 후에는 NameMarketplaceProduct, 및 tag-on-create 값을 변경할 수 없습니다. 리소스에 태그를 다시 지정하려면 어트리뷰션의 ARN에서 표준 AWS 태그 지정 작업을 사용합니다.

수익 속성 ID를 업데이트할 때 파트너는 다음을 제공해야 합니다.

- **Catalog** - AWS 또는 Sandbox.
- **Identifier** - 업데이트할 어트리뷰션의 ra- ID입니다.
- **Version** - 낙관적 잠금을 위한 속성의 현재 버전입니다.

파트너는 선택적으로 다음을 제공할 수 있습니다.

- **Description** - 업데이트된 자유 텍스트 설명입니다. 최대 1,024자입니다.
- **ClientToken** - 업데이트를 위한 멍등성 토큰입니다.

낙관적 잠금: Version 필드는 마지막으로 검색된 이후 속성이 변경되지 않은 경우에만 업데이트가 적용되도록 합니다. 제출된가 현재 리소스 버전과 일치하지 Version 않으면와 함께 업데이트가 거부ConflictException되고 제출된 및 현재 버전 번호가 포함된 메시지가 표시됩니다. 모범 사례는 각 업데이트 GetRevenueAttribution 전에를 사용하여 최신 버전을 검색하는 것입니다.

수익 속성 ID 세부 정보 보기

파트너는 GetRevenueAttribution API 작업을 사용하여 단일 수익 속성 ID에 대한 전체 정보를 검색할 수 있습니다. 반환:

리소스 메타데이터:

- 고유 식별자(Id) 및 Amazon 리소스 이름(Arn).
- 어트리뷰션이 존재하는 Catalog입니다.
- Name 및 Description.
- 검색된 Version 및 LatestVersion (후속 업데이트에는 최신 사용).

Marketplace 제품 속성(생성 시 MarketplaceProduct가 설정된 경우):

- ProductId - 파트너가 제공한 Marketplace 제품 ID입니다.
- ProductCode -AWS Marketplace 제품 코드에서 해결되었습니다ProductId.
- ProductType - SaaS, AMI, Container, MLData, 또는 Professional Services.
- TenancyModel - MULTI_TENANT 또는 SINGLE_TENANT.

감사 정보:

- CreatedDate 및 LastModifiedDate.

파트너는 필요에 따라 Version 요청에 특징를 제공하여 과거 버전의 어트리뷰션을 검색할 수 있습니다. 최신를 반환Version하려면 생략합니다.

파트너는 특정 월별 비용 할당 항목을 검색하기 위해 항목의와 함께 `GetRevenueAttributionAllocation` API 작업을 사용합니다 `AllocationId`. 그러면 제안 ID 또는 기회 ID, 결제 월, 비용 할당 %, 고객AWS계정 ID, 항목의 상태 및 감사 정보가 반환됩니다.

수익 속성 IDs 나열

파트너는 `ListRevenueAttributions` API 작업을 사용하여 계정의 모든 수익 속성 IDs를 볼 수 있습니다. 필터링 및 정렬 기능이 있는 속성 요약의 페이지 매김 목록을 반환합니다.

파트너는 다음을 기준으로 결과를 필터링할 수 있습니다.

- **Catalog** - AWS 또는 Sandbox (필수).
- **Identifiers** - 검색할 최대 100개의 특정 ra- ID 목록입니다. IDs
- **CreatedAfter / CreatedBefore** - 생성 타임스탬프 범위(포함)를 기준으로 필터링합니다.

파트너는 `Sort` 파라미터를 사용하여 결과 정렬을 구성할 수 있습니다.

- `SortBy` - `CreatedDate` 또는 `LastModifiedDate`.
- `SortOrder` - `ASCENDING` 또는 `DESCENDING`.

응답에는 `Arn`, `Id`, `Catalog`, `Name` 확인된 `MarketplaceProduct` 속성, `CreatedDate`, `Version`, `LastModifiedDate` 최신 및 `TotalRevenueAttributionAssociationCount` (속성에 연결된 활성 월별 비용 할당 항목 수)의 각 속성에 대한 요약이 포함됩니다.

`MaxResults` (기본값 25, 최대 100) 및 `NextToken`를 사용하여 큰 결과 집합을 페이지 매김합니다. 추가 페이지를 사용할 수 있는 `NextToken` 경우 응답에이 포함됩니다.

월별 비용 할당 항목 나열

파트너는 `ListRevenueAttributionAllocations` API 작업을 사용하여 특정 수익 속성 ID에 대한 월별 비용 할당 항목을 나열할 수 있습니다. 그러면 필터링 기능이 있는 할당 요약의 페이지가 매겨진 목록이 반환됩니다.

파트너는 다음을 기준으로 결과를 필터링할 수 있습니다.

- **MarketplaceOfferId** - 특정 Marketplace 제안과 연결된 항목만 나열합니다.
- **AwsPartnerCentralOpportunityId** - 특정 파트너 중앙 기회와 연결된 항목만 나열합니다.
- **BillingMonth** - 특정 월의 항목만 나열합니다.

응답에는 AllocationId, 관련 제안 ID 또는 기회 ID, 고객AWS계정 ID, 결제 월, 비용 할당 %, 항목의 이름과 상태, 감사 타임스탬프 등 각 항목에 대한 요약 정보가 포함됩니다.

대시보드 구축: ListRevenueAttributions 및의 조

합ListRevenueAttributionAllocations은 파트너 대시보드 생성을 지원하도록 설계되었습니다. 파트너는 다음과 같은 뷰를 빌드할 수 있습니다.

- 지난 30일 동안 생성된 모든 수익 속성 IDs 생성 날짜를 기준으로 정렬됩니다.
- 여러 수익 속성 IDs.
- 총계가 어트리뷰션당 100%를 초과하지 않는지 확인하기 위해 현재 결제 월의 모든 비용 할당 항목입니다.
- 하나 이상의 활성 할당 항목()이 있는 모든 수익 속성
IDsTotalRevenueAttributionAssociationCount > 0.

Marketplace 수익 공유 작업

Marketplace 수익 공유는AWS Marketplace를 통해 수집된 제품 총 수익의 일부를 선언하는AWS Marketplace 제품에 대한 입력입니다.이 입력을AWS사용하여 파트너의 Marketplace 수집 수익과 다른 수익 신호를 상호 연관시킵니다.

Marketplace Revenue Share Service는 APIs 노출하여 Marketplace Revenue Share 리소스를 생성, 검색 및 나열하고 할당(파트너가 선언한 수익 공유 비율 및 유효 날짜 기간)을 관리합니다. 표준AWS태그 지정 작업은 Marketplace Revenue Share 리소스에서 지원됩니다.

Marketplace 수익 공유란 무엇입니까?

Marketplace 수익 공유에는 두 개의 계층이 있습니다.

- 공유 - Marketplace 로 식별되는 단일AWS Marketplace 제품에 대한 입력입니다ProductId. 제품당 Marketplace 수익 공유는 정확히 1개입니다.
- 하나 이상의 할당 - 각 할당은 RevenueSharePercent 및 유효 날짜 기간(EffectiveDate, 선택 사항)을 선언합니다ExpirationDate. 공유는 시간이 지남에 따라 많은 할당을 가질 수 있지만 언제든지 하나의 ACTIVE 할당만 적용될 수 있습니다.

할당에는 다음과 같은 주요 속성이 있습니다.

- 상태 - ACTIVE 또는 INACTIVE. 생성 시 상태는 기본적으로 로 설정됩니다ACTIVE. ACTIVE 할당만 수익 계산 및 중복 검사에 참여합니다.

- 겹치는 불변성 - 단일 공유 내에서 두 ACTIVE 할당에 겹치는 [EffectiveDate, ExpirationDate] 범위가 있을 수 없습니다. 언제든지 공유당 최대 1개의 개방형 ACTIVE 할당 (없음ExpirationDate)이 허용됩니다.

Marketplace 수익 공유 작업

파트너는 Marketplace Revenue Share Service를 통해 Marketplace Revenue Shares 및 할당을 관리합니다. 수명 주기는 공유 흐름(생성, 검색, 나열, 태그)과 할당 흐름(생성, 업데이트, 소프트 삭제, 검색, 목록)의 두 흐름을 통해 진행됩니다.

Marketplace 수익 공유 생성

첫 번째 단계는 CreateMarketplaceRevenueShare API 작업을 사용하여 Marketplace 제품에 대한 AWS Marketplace 수익 공유를 생성하는 것입니다. 공유는 Marketplace에서 식별ProductId되며 모든 후속 할당을 위한 컨테이너 역할을 합니다.

Marketplace 수익 공유를 생성할 때 파트너는 다음을 제공해야 합니다.

- **Catalog** - AWS 프로덕션용 또는 테스트Sandbox용.
- **ProductId** - 25자 Marketplace 제품 ID입니다. 패턴: [a-z0-9]{25}.

파트너는 선택적으로 다음을 제공할 수 있습니다.

- **Tags** - 생성 시 리소스 조직에 대해 최대 50개의 키-값 페어. 생성 후 TagResource를 사용하여 태그를 추가합니다.
- **ClientToken** - 재시도 시 중복 생성을 방지하기 위한 멍등성 토큰입니다. 최대 64자입니다. SDKs 이 토큰을 자동으로 생성합니다.

제품 검증: 생성 시 서비스는AWS Marketplace를 통해 호출자의 계정 또는 PAC(파트너 계정 연결)를 통해 호출자에게 연결된 자회사 계정이 제품을 소유하고 있는지 확인합니다. 제품이 없거나 호출자에게 소유 계정이 승인되지 않은 경우 API는 이유와 ValidationException 함께를 반환합니다PRODUCT_NOT_FOUND_OR_NOT_OWNED.

제품당 공유 1개: 동일한 ProductId에서 동일한에 대한 두 번째 CreateMarketplaceRevenueShare 호출은를 Catalog 반환합니다ConflictException.

응답은 공유의 Arn (형식 arn:{partition}:partnercentral:{region}:{account-id}:catalog/{catalog}/marketplace-revenue-share/{productId}), Catalog,

ProductId, 초기 Version 숫자(각 할당 변형에 대해 1부터 시작하여 증분) 및 생성 시 적용된 모든 태그를 반환합니다.

할당 추가

Marketplace 수익 공유가 생성되면 파트너는 CreateMarketplaceRevenueShareAllocation API 작업을 통해 할당을 생성하여 수익 공유 비율과 유효 날짜 기간을 선언합니다.

할당을 생성할 때 파트너는 다음을 제공해야 합니다.

- **Catalog** - AWS 또는 Sandbox.
- **MarketplaceRevenueShareIdentifier** - 상위 공유의 식별자입니다.
- **RevenueSharePercent** - 10진수로 제공된 Marketplace를 통해 수집된 제품 수익의 백분율입니다 (예: 0.45 45%).
- **EffectiveDate** -이 할당이 적용되는 달력 날짜입니다. 형식: YYYY-MM-DD

응답은 할당Id(형식 mrsa-[A-Za-z0-9]{13}), 할당 상태(ACTIVE생성 시) 및 상위 공유의 범프Version(모든 할당 변형에서 공유당 단일 버전 정수가 범프되어 후속 할당 업데이트에서 낙관적 잠금에 사용됨)를 반환합니다.

할당 업데이트

파트너는 UpdateMarketplaceRevenueShareAllocation API 작업을 사용하여 할당을 업데이트 할 수 있습니다. 업데이트 가능한 필드는 할당이 미래 날짜인지, 현재 유효한지 또는 과거 날짜인지에 따라 달라집니다.

할당을 업데이트할 때 파트너는 다음을 제공해야 합니다.

- **Catalog** - AWS 또는 Sandbox.
- **MarketplaceRevenueShareIdentifier** - 상위 공유의 식별자입니다.
- **AllocationId** - 업데이트할 할당의 mrsa- ID입니다.
- **MarketplaceRevenueShareVersion** - 할당에 대한 낙관적 잠금을 위한 상위 공유의 현재 버전입니다. 범프 충돌을 반환합니다ConflictException.

파트너는 선택적으로 다음을 제공할 수 있습니다.

- **RevenueSharePercent** - 업데이트된 수익 공유 비율입니다. 미래 날짜의 할당에서만 편집할 수 있습니다.

- **EffectiveDate** - 업데이트된 유효 날짜입니다. 미래 날짜의 할당에서만 편집할 수 있습니다.
- **ExpirationDate** - 업데이트된 만료 날짜입니다. 미래 날짜 할당(모든 값 > EffectiveDate) 및 현재 유효하거나 개방형 할당(모든 값 > today)에서 편집할 수 있습니다.
- **Status** - ACTIVE 또는 INACTIVE. 를 로 설정하면 할당이 INACTIVE 소프트웨어 삭제됩니다. 미래 날짜의 할당에서만 편집할 수 있습니다.
- **ClientToken** - 맥동성 토큰입니다.

Marketplace 수익 공유 세부 정보 보기

파트너는 GetMarketplaceRevenueShare API 작업을 사용하여 단일 Marketplace 수익 공유에 대한 전체 정보를 검색할 수 있습니다. 그러면 공유의 Arn, Catalog, ProductId, Version (상위 공유의 현재 버전), CreatedDateLastModifiedDate, 및 공유에 적용된 모든 태그가 반환됩니다.

파트너는 할당 의와 함께 GetMarketplaceRevenueShareAllocation API 작업을 사용하여 단일 할당을 검색할 수 있습니다Id. 반환:

- 할당 Id 및 상위 공유의 입니다MarketplaceRevenueShareArn.
- RevenueSharePercent, EffectiveDate, ExpirationDate.
- Status (ACTIVE 또는 INACTIVE).
- 후속 업데이트 시 낙관적 잠금을 MarketplaceRevenueShareVersion 위한 상위 공유의 입니다.
- CreatedDate 및 LastModifiedDate.

Marketplace 수익 공유 나열

파트너는 ListMarketplaceRevenueShares API 작업을 사용하여 자신의 계정이 소유한 모든 Marketplace 수익 공유를 볼 수 있습니다. 페이지가 매겨진 공유 요약 목록이 반환됩니다.

파트너는 다음을 기준으로 결과를 필터링할 수 있습니다.

- **Catalog** - AWS 또는 Sandbox (필수).
- **ProductIds** - 검색할 특정 Marketplace 제품 IDs의 목록입니다.

응답에는 Arn, , Catalog, 최신 ProductId, CreatedDate 및 각 공유에 대한 요약Version이 포함됩니다LastModifiedDate.

MaxResults 및 NextToken를 사용하여 대규모 결과 집합을 페이지 매깁니다.

할당 나열

파트너는 ListMarketplaceRevenueShareAllocations API 작업을 사용하여 특정 Marketplace 수익 공유에 대한 할당을 나열할 수 있습니다. 그러면 필터링 기능이 있는 할당 요약의 페이지가 매겨진 목록이 반환됩니다.

파트너는 다음을 기준으로 결과를 필터링할 수 있습니다.

- **Status** - ACTIVE 또는 INACTIVE.
- 유효 날짜 범위 - 유효 날짜가 특정 기간 내에 속하는 할당을 검색EffectiveDateOnOrBefore하는 EffectiveDateOnOrAfter /입니다.

응답에는 Id, , RevenueSharePercent, EffectiveDate, 및 감사 타임스탬프ExpirationDateStatusMarketplaceRevenueShareVersion와 같은 각 할당에 대한 요약 정보가 포함됩니다.

Marketplace 수익 공유 태그 지정

표준AWS태그 지정 작업은 Marketplace Revenue Share 리소스에서 지원됩니다.

- **TagResource** - Marketplace 수익 공유에 태그를 추가하거나 덮어씁니다. 공유당 최대 50개의 태그.
- **UntagResource** - 키별로 특정 태그를 제거합니다.
- **ListTagsForResource** - Marketplace 수익 공유에 현재 적용된 모든 태그를 검색합니다.

할당은 태그를 지원하지 않습니다. 상위 공유에 태그를 지정하여 관련 할당을 구성합니다.

각 태그 지정 작업은 공유의 Arn (형식 `arn:{partition}:partnercentral:{region}:{account-id}:catalog/{catalog}/marketplace-revenue-share/{productId}`)를 리소스 식별자로 수락합니다. Standard AWS tag-on-create는의 Tags 필드를 통해서도 지원됩니다CreateMarketplaceRevenueShare.

지원되는 AWS 리전

다음 섹션에서는 지원되는 AWS 리전에 대한 정보를 제공합니다.

주제

- [AWS 파트너 중앙 계정 API에 지원되는 AWS 리전](#)
- [AWS Partner Central Selling API에 지원되는 AWS 리전](#)
- [AWS Partner Central Benefits API에 지원되는 AWS 리전](#)
- [AWS 파트너 중앙 채널 API에 지원되는 AWS 리전](#)
- [AWS 파트너 중앙 수익 측정 API에 지원되는 AWS 리전](#)

AWS 파트너 중앙 계정 API에 지원되는 AWS 리전

다음 엔드포인트를 사용하여 모든 AWS 미국 동부(버지니아 북부) 리전에서 AWS Partner Central 계정 서비스에 액세스할 수 있습니다.

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

AWS Partner Central Selling API에 지원되는 AWS 리전

다음 엔드포인트를 사용하여 모든 AWS 미국 동부(버지니아 북부) 리전에서 AWS Partner Central 판매 서비스에 액세스할 수 있습니다.

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

AWS Partner Central Benefits API에 지원되는 AWS 리전

다음 엔드포인트를 사용하여 모든 AWS 미국 동부(버지니아 북부) 리전에서 AWS Partner Central 혜택 서비스에 액세스할 수 있습니다.

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

AWS 파트너 중앙 채널 API에 지원되는 AWS 리전

다음 엔드포인트를 사용하여 모든 AWS 상용 리전에서 AWS Partner Central 채널 관리 서비스에 액세스할 수 있습니다.

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

SigV4a를 사용하여 요청에 서명

AWS Partner Central Channel API는 글로벌 엔드포인트를 사용하기 때문에 요청은 비대칭 암호화로 서명을 지원하는 SigV4의 확장인 서명 버전 4a(SigV4a)로 서명되어 여러 AWS 리전에서 확인할 수 있는 서명을 활성화합니다. SigV4

AWS SDK 또는 CLI 사용

AWS SDK 또는 AWS CLI를 사용하는 경우 글로벌 엔드포인트를 호출하면 SigV4a 서명이 자동으로 처리됩니다. 추가 구성은 필요하지 않습니다.

요청에 수동으로 서명

SDK 없이 요청에 수동으로 서명하는 경우 표준 SigV4와 다음과 같은 차이점에 유의하세요.

- SigV4a는 HMAC 기반 키 파생을 사용하는 대신 기존 AWS 보안 액세스 키에서 타원 곡선 디지털 서명 알고리즘(ECDSA) 키 페어를 파생합니다.
- 서명은 * 리전 간에 유효하므로 자격 증명 범위는 특정 리전 이름 대신 리전 구성 요소에 사용됩니다.

SigV4a의 작동 방식에 대한 자세한 내용은 [API 요청에 대한 AWS 서명 버전 4](#)를 참조하세요. SigV4a 서명의 코드 예제는 GitHub의 [sigv4a-signing-examples](#) 프로젝트를 참조하세요.

AWS 파트너 중앙 수익 측정 API에 지원되는 AWS 리전

다음 엔드포인트를 사용하여 모든 AWS 상용 리전에서 AWS 파트너 중앙 수익 측정 서비스에 액세스할 수 있습니다.

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

SigV4a를 사용하여 요청에 서명

수익 측정 서비스는 글로벌 엔드포인트를 사용하기 때문에 요청은 비대칭 암호화로 서명을 지원하는 SigV4의 확장인 서명 버전 4a(SigV4a)로 서명되어 여러 AWS 리전에서 확인할 수 있는 서명을 활성화합니다. SigV4

AWS SDK 또는 CLI 사용

AWS SDK 또는 AWS CLI를 사용하는 경우 글로벌 엔드포인트를 호출하면 SigV4a 서명이 자동으로 처리됩니다. 추가 구성은 필요하지 않습니다.

요청에 수동으로 서명

SDK 없이 요청에 수동으로 서명하는 경우 표준 SigV4와 다음과 같은 차이점에 유의하세요.

- SigV4a는 HMAC 기반 키 파생을 사용하는 대신 기존 AWS 보안 액세스 키에서 타원 곡선 디지털 서명 알고리즘(ECDSA) 키 페어를 파생합니다.
- 서명은 * 리전 간에 유효하므로 자격 증명 범위는 특정 리전 이름 대신 리전 구성 요소에 사용됩니다.

SigV4a의 작동 방식에 대한 자세한 내용은 [API 요청에 대한 AWS 서명 버전 4](#)를 참조하세요. SigV4a 서명의 코드 예제는 GitHub의 [sigv4a-signing-examples](#) 프로젝트를 참조하세요.

권한

다음 섹션에서는 권한에 대한 정보를 제공합니다.

주제

- [계정 API에 대한 액세스](#)
- [Selling API에 대한 액세스](#)
- [Benefits API에 대한 액세스](#)
- [채널 API에 대한 액세스](#)
- [수익 측정 API에 대한 액세스](#)

계정 API에 대한 액세스

액세스 제어 및 권한은 AWS Identity and Access Management(IAM)에서 관리합니다. 이 섹션에서는 계정 API와 상호 작용하는 데 필요한 권한을 구성하는 지침을 제공합니다.

사전 조건

권한을 구성하기 전에 AWS 계정가에 연결되어 있고 필요한 IAM 역할 및 사용자를 생성했는지 확인합니다. 자세한 내용은 [설정 및 인증](#) 단원을 참조하십시오.

AWS관리형 정책 사용

AWS는 계정 API와 상호 작용하는 데 필요한 권한을 부여하는 관리형 정책을 제공합니다. 계정 리소스를 관리하는 데 필요한 액세스 권한을 제공하려면 `AWSPartnerCentralFullAccess` 정책을 IAM 자격 증명에 연결합니다. 자세한 내용은 [AWS사용자를 위한 관리형 정책](#)을 참조하세요.

IAM 역할 및 사용자에게 정책 할당

다음 단계에 따라 IAM 역할 및 사용자에게 정책을 할당합니다.

1. AWS Management Console에 로그인합니다.
2. IAM 서비스로 이동합니다.
3. 역할 또는 사용자를 선택하고 정책을 연결할 IAM 역할 또는 사용자를 선택합니다.
4. 선택한 IAM 역할 또는 사용자에게 `AWSPartnerCentralFullAccess` 정책을 연결합니다.

자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

조건 키를 사용하여 권한 관리

IAM 정책의 조건 키는 문 정책을 적용할 시기에 대한 리소스 수준 권한을 제공합니다. 조건 키를 사용하여 특정 권한이 허용되거나 거부되는 시기를 지정하는 조건을 지정할 수 있습니다.

자세한 내용은 [IAM JSON 정책 요소: 조건 연산자](#)를 참조하세요.

조건 키 개요

조건 키	설명	적용 가능한 작업	유효값
<code>partnercentral:Catalog</code>	연결된 카탈로그 엔터티의 유형을 기준으로 액세스를 필터링합니다.	모든 작업	AWS, 샌드박스

필수 권한 요약

필수 권한 요약

작업	설명
<code>partnercentral:AcceptConnectionInvitation</code>	연결 초대 수락 허용
<code>partnercentral:AssociateAwsTrainingCertificationEmailDomain</code>	AWS훈련 인증 이메일 도메인 연결 허용
<code>partnercentral:CancelConnection</code>	연결 취소 허용

작업	설명
partnercentral:CancelConnectionInvitation	연결 초대 취소 허용
partnercentral:CancelProfileUpdateTask	프로필 업데이트 작업 취소 허용
partnercentral:CreateConnectionInvitation	에서 연결 초대 생성 허용
partnercentral:CreatePartner	에서 파트너 생성 허용
partnercentral:DisassociateAwsTrainingCertificationEmailDomain	AWS훈련 인증 이메일 도메인의 연결 해제 허용
partnercentral:GetAllianceLeadContact	제휴 리드 연락처 세부 정보 검색 허용
partnercentral:GetConnection	연결 세부 정보 검색 허용
partnercentral:GetConnectionInvitation	연결 초대 세부 정보 검색 허용
partnercentral:GetConnectionPreferences	연결 기본 설정 검색 허용
partnercentral:GetPartner	파트너 세부 정보 검색 허용
partnercentral:GetProfileUpdateTask	프로필 업데이트 작업 세부 정보 검색 허용
partnercentral:GetProfileVisibility	프로필 가시성 설정 검색 허용
partnercentral:GetVerification	확인 세부 정보 검색 허용
partnercentral:ListConnectionInvitations	연결 초대 나열 허용
partnercentral:ListConnections	연결 나열 허용
partnercentral:ListPartners	파트너 나열 허용
partnercentral:PutAllianceLeadContact	제휴 책임자 연락처 세부 정보 업데이트 허용
partnercentral:PutProfileVisibility	프로필 가시성 설정 업데이트 허용
partnercentral:RejectConnectionInvitation	연결 초대 거부 허용
partnercentral:SendEmailVerificationCode	이메일 확인 코드 전송 허용

작업	설명
partnercentral:StartProfileUpdateTask	프로파일 업데이트 작업 시작 허용
partnercentral:StartVerification	확인 프로세스 시작 허용
partnercentral:UpdateConnectionPreferences	연결 기본 설정 업데이트 허용

Selling API에 대한 액세스

액세스 제어 및 권한은 AWS Identity and Access Management(IAM)에서 관리합니다. 이 섹션에서는 AWS Marketplace 엔터티를 나열하는 데 필요한 권한을 포함하여 API와 상호 작용하는 데 필요한 권한을 구성하는 지침을 제공합니다.

사전 조건

권한을 구성하기 전에 AWS 계정이 Partner Central에 연결되어 있고 필요한 IAM 역할 및 사용자를 생성했는지 확인합니다. 자세한 내용은 [설정 및 인증](#)을 참조하세요.

AWS관리형 정책 사용

AWS는 API와 상호 작용하는 데 필요한 권한을 부여하는 관리형 정책을 제공합니다. 기회를 관리하는 데 필요한 액세스 권한을 제공하려면 IAM 자격 증명에 AWSPartnerCentralOpportunityManagement 정책을 연결합니다. 자세한 내용은 [AWSAWS Partner Central 사용자를 위한 관리형 정책](#)을 참조하세요.

AWSPartnerCentralOpportunityManagement 정책

이 정책은 Partner Central 기회 관리 작업에 대한 전체 액세스 권한을 부여합니다.

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/*"
        ]
    }
]
}

```

사용자 지정 정책

관리형 정책이 요구 사항을 충족하지 않는 경우 사용 사례에 필요한 권한을 부여하는 사용자 지정 IAM 정책을 생성합니다. 다음 예제는 AWS Marketplace 엔터티를 나열할 수 있는 권한을 부여하는 사용자 지정 정책입니다.

Example 사용자 지정 정책의 예

JSON

```

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

```

사용자 지정 허용 정책

이 정책은 정책 업데이트 없이 향후 추가될 수 있는 기능을 포함하여 Partner Central 판매 작업에 대한 광범위한 액세스를 제공합니다. `partnercentral:*`이 정책은 와일드카드 작업을 사용하여 새 Partner Central 판매 기능이 출시되면 해당 기능에 대한 액세스 권한을 자동으로 부여하므로 수동 업

데이트가 필요하지 않습니다. 이 정책에는 AWS Marketplace 엔터티와 상호 작용할 수 있는 권한도 포함되어 있으므로 판매 및 Marketplace 작업 모두에 대한 액세스가 유지되도록 하는 데 도움이 됩니다.

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"
        }
      }
    }
  ]
}
```

}

IAM 역할 및 사용자에게 정책 할당

다음 단계에 따라 IAM 역할 및 사용자에게 정책을 할당합니다.

1. AWS Management Console에 로그인합니다.
2. IAM 서비스로 이동합니다.
3. 역할 또는 사용자를 선택하고 정책을 연결할 IAM 역할 또는 사용자를 선택합니다.
4. `AWSPartnerCentralOpportunityManagement` 정책 또는 사용자 지정 정책을 선택한 IAM 역할 또는 사용자에게 연결합니다.

자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

조건 키를 사용하여 권한 관리

IAM 정책의 조건 키는 문 정책을 적용할 시기에 대한 리소스 수준 권한을 제공합니다. 조건 키를 사용하여 특정 권한이 허용되거나 거부되는 시기를 지정하는 조건을 지정할 수 있습니다.

자세한 내용은 [IAM JSON 정책 요소: 조건 연산자](#)를 참조하세요.

조건 키 개요

조건 키	설명	적용 가능한 작업	유효값
partnercentral:Catalog	연결된 카탈로그 엔터티의 유형을 기준으로 액세스를 필터링합니다.	모든 작업	AWS, 샌드박스
aws-marketplace:PartyType	는 당사자 유형(예: 제안자)을 기준으로 액세스를 필터링합니다.	SearchAgreements, DescribeAgreement	제안자

필수 권한 요약

필수 권한 요약

작업	설명
partnercentral:CreateOpportunity	는 기회 생성을 허용합니다.
partnercentral:UpdateOpportunity	에서 기회 업데이트 허용
partnercentral:ListOpportunities	는 기회 나열을 허용합니다.
partnercentral:GetOpportunity	기회 세부 정보 검색 허용
partnercentral:ListSolutions	는 솔루션 나열을 허용합니다.
partnercentral:AssociateOpportunity	다른 엔터티와 기회를 연결할 수 있습니다.
partnercentral:DisassociateOpportunity	다른 엔터티와 기회를 연결 해제할 수 있습니다.
partnercentral:AcceptEngagementInvitation	참여 초대 수락 허용
partnercentral:RejectEngagementInvitation	참여 초대 거부 허용
partnercentral:GetEngagementInvitation	참여 초대 세부 정보 검색 허용
partnercentral:ListEngagementInvitations	참여 초대를 나열할 수 있습니다.
partnercentral:SubmitOpportunity	에서 기회 제출 허용
partnercentral:GetAwsOpportunitySummary	AWS기회 요약 검색 허용
partnercentral:StartProspectingFromEngagementTask	참여에서 잠재 고객 발굴 작업을 생성합니다.
partnercentral:GetProspectingFromEngagementTask	잠재 고객 발굴 작업 세부 정보를 반환합니다.
partnercentral:ListProspectingFromEngagementTasks	잠재 고객 발굴 작업 목록을 반환합니다.
aws-marketplace:ListEntities	AWS Marketplace엔터티 나열 허용

작업	설명
aws-marketplace:DescribeEntity	에서는AWS Marketplace엔터티를 설명할 수 있습니다.
aws-marketplace:SearchAgreements	에서 계약 검색 허용AWS Marketplace
aws-marketplace:DescribeAgreement	에서 계약 설명 허용AWS Marketplace

Benefits API에 대한 액세스

액세스 제어 및 권한은AWS Identity and Access Management(IAM)에서 관리합니다. 이 섹션에서는 혜택 API와 상호 작용하는 데 필요한 권한을 구성하는 지침을 제공합니다.

사전 조건

권한을 구성하기 전에이AWS 계정에 연결되어 있고 필요한 IAM 역할 및 사용자를 생성했는지 확인합니다. 자세한 내용은 [설정 및 인증](#) 단원을 참조하십시오.

AWS관리형 정책 사용

AWS는 Benefits API와 상호 작용하는 데 필요한 권한을 부여하는 관리형 정책을 제공합니다. 혜택 리소스를 관리하는 데 필요한 액세스 권한을 제공하려면 IAM 자격 증명에 `AWSPartnerCentralFullAccess` 정책을 연결합니다. 자세한 내용은 [AWS사용자를 위한 관리형 정책을](#) 참조하세요.

IAM 역할 및 사용자에게 정책 할당

다음 단계에 따라 IAM 역할 및 사용자에게 정책을 할당합니다.

1. AWS Management Console에 로그인합니다.
2. IAM 서비스로 이동합니다.
3. 역할 또는 사용자를 선택하고 정책을 연결할 IAM 역할 또는 사용자를 선택합니다.
4. 선택한 IAM 역할 또는 사용자에게 `AWSPartnerCentralFullAccess` 정책을 연결합니다.

자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

조건 키를 사용하여 권한 관리

IAM 정책의 조건 키는 문 정책을 적용할 시기에 대한 리소스 수준 권한을 제공합니다. 조건 키를 사용하여 특정 권한이 허용되거나 거부되는 시기를 지정하는 조건을 지정할 수 있습니다.

자세한 내용은 [IAM JSON 정책 요소: 조건 연산자](#)를 참조하세요.

조건 키 개요

조건 키	설명	적용 가능한 작업	유효값
partnercentral:Catalog	연결된 카탈로그 엔터티의 유형을 기준으로 액세스를 필터링합니다.	모든 작업	AWS, 샌드박스

필수 권한 요약

필수 권한 요약

작업	설명
partnercentral:AmendBenefitApplication	혜택 애플리케이션 수정 허용
partnercentral:AssociateBenefitApplicationResource	를 사용하면 리소스를 혜택 애플리케이션과 연결할 수 있습니다.
partnercentral:CancelBenefitApplication	혜택 애플리케이션 취소 허용
partnercentral:CreateBenefitApplication	에서 혜택 애플리케이션 생성 허용
partnercentral:DisassociateBenefitApplicationResource	혜택 애플리케이션에서 리소스 연결 해제 허용
partnercentral:GetBenefit	혜택 세부 정보 검색 허용
partnercentral:GetBenefitAllocation	혜택 할당 세부 정보를 검색할 수 있습니다.
partnercentral:GetBenefitApplication	혜택 애플리케이션 세부 정보를 검색할 수 있습니다.

작업	설명
partnercentral:ListBenefitAllocations	혜택 할당 나열 허용
partnercentral:ListBenefitApplications	혜택 애플리케이션 나열 허용
partnercentral:ListBenefits	에서 혜택 나열 허용
partnercentral:RecallBenefitApplication	혜택 애플리케이션 리콜 허용
partnercentral:SubmitBenefitApplication	혜택 애플리케이션 제출 허용
partnercentral:UpdateBenefitApplication	혜택 애플리케이션 업데이트 허용

채널 API에 대한 액세스

액세스 제어 및 권한은 AWS Identity and Access Management(IAM)에서 관리합니다. 이 섹션에서는 AWS Partner Central Channel API와 상호 작용하는 데 필요한 권한을 구성하기 위한 지침을 제공합니다.

채널 관리 계정 설정

AWS Partner Central의 채널 관리 기능을 사용하려면 다음 두 가지 유형의 AWS 계정에 IAM 역할을 배포해야 합니다.

1. [AWS Partner Central 연결 AWS 계정](#):

이는 AWS Partner Network 및 Partner Central 계정과 AWS 계정 연결된입니다. AWS 파트너 AWS 계정에 연결된 Partner Central은 하나뿐입니다. 일회성 설정 활동으로 Partner Central 채널 관리형 정책을 AWS 계정 사용하여 IAM 역할을 생성합니다.

2. [프로그램 관리 계정 AWS 계정](#):

최종 고객 소비에 대한 결제 및 결제를 관리하기 위해 Bill-Transfer 계정으로 AWS 계정 사용되는입니다. 이는 AWS Partner Central 채널 관리에서 프로그램 관리 계정으로 보고 AWS 계정으로됩니다. 프로그램 관리 계정이 여러 개 있을 수 있으며 프로그램 관리 계정으로 AWS 계정보고하는 각에서 IAM 역할을 생성해야 합니다.

각 AWS 계정 유형에 대해 서로 다른 IAM 역할을 특정 권한 및 신뢰 정책과 연결해야 합니다.

연결된AWS Partner Central에 대한 액세스AWS 계정

연결된AWS Partner Central 내에서 채널 리소스를 관리할 IAM 역할을AWS 계정생성합니다.이 역할은 Partner Central 사용자에게 부여됩니다. 이 IAM 역할을 통해 사용자는AWS Partner Central에서 채널 관리 리소스를 보고 관리할 수 있습니다. 역할 이름은 로 시작해야 하며 PartnerCentralRoleFor권장 이름은 입니다PartnerCentralRoleForChannel. 역할을 구성하려면 다음을 완료합니다.

- 다음 사용자 지정 신뢰 정책을 추가하여AWS Partner Central 클라우드 관리자 및 제휴가 [IAM 역할을 Partner Central 사용자에게 매핑](#)하도록 허용합니다.

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

- [AWSPartnerCentralChannelManagement](#) 관리형 정책을 PartnerCentralRoleForChannel 역할에 연결합니다.

프로그램 관리 계정AWS계정(들)에 대한 액세스

AWS Partner Central에서 프로그램 관리 계정으로AWS 계정보고된 각 내에서 아래 필수 이름을 사용하여 다음 IAM 역할을 생성합니다.

- PartnerCentralChannelHandshakeApprovalManagement:이 역할을 사용하여 Partner Central과 프로그램 관리 간의 핸드셰이크를 관리합니다AWS 계정. 이 IAM 역할을 활용하여 AWS Partner Central에서 프로그램 관리 계정을 활성화할 수 있습니다. 이 역할을 생성할 때 AWSPartnerCentralChannelHandshakeApprovalManagement 관리형 정책을 연결합니다.
- PartnerCentralChannelBillingTransferReadOnly:이 교차 계정 역할을 활용하여AWS Partner Central에서 결제 전송 상태를 확인할 수 있습니다. 이 역할은에서AWS Partner Central AWS 계정으로 결제 전송 상태를 공유하여 결제 전송 상태를 중앙에서 확인합니다. 이 역할이 생성되면AWS Partner Central의 사용자는AWS Partner Central에서 결제 전송 상태를 볼 수 있습니다. 그

러나이 역할은 AWS Partner Central 사용자에게 AWS Billing and Cost Management 콘솔에서 결제 전송에 액세스할 수 있는 권한을 부여하지 않습니다.

- 신뢰 정책:

- 아래 JSON 내에서 역할을 생성할 때를 실제 AWS Partner Central linked AWS 계정 12자리 ID{PartnerCentralLinkedAccountId}로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{{PartnerCentralLinkedAccountId}}:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- 권한 정책:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "BillingTransferReadOnly",
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeResponsibilityTransfer",
        "organizations:ListHandshakesForOrganization",
        "organizations:ListInboundResponsibilityTransfers",
        "organizations:ListOutboundResponsibilityTransfers"
      ],
      "Resource": "*"
    }
  ]
}
```

- PartnerCentralChannelBillingTransferManagement (선택 사항): 이 교차 계정 역할을 활용하여 AWS Partner Central에서 결제 전송을 생성하고 관리할 수 있습니다. 이 역할을 통해 AWS Partner Central 사용자는 AWSPartnerCentralChannelManagement 관리형 정책이 포함된 IAM

역할에 액세스하여 프로그램 관리 계정의 결제 전송을 관리할 수 있습니다AWS 계정. 이 교차 계정 역할을 사용하면AWS Partner Central 사용자는AWS Partner Central 채널 관리의 '청구에서 관리' 버튼을 사용하여AWS Billing and Cost Management 콘솔에서AWS결제 전송에 액세스하고 관리할 수 있습니다.

- 신뢰 정책:

- 아래 JSON 내에서 역할을 생성할 때를 실제AWS Partner Central linked AWS 계정 12자리 ID{PartnerCentralLinkedAccountId}로 바꿉니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{PartnerCentralLinkedAccountId}:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- 권한 정책:

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

```

AWS관리형 정책 사용

AWS는 채널 API와 상호 작용하는 데 필요한 권한을 부여하는 관리형 정책을 제공합니다. 모든 채널 리소스를 관리하는 데 필요한 액세스 권한을 제공하려면 `AWSPartnerCentralChannelManagement` 정책을 IAM 자격 증명에 연결합니다. 채널 핸드셰이크 요청을 보고 응답하는 데 필요한 액세스 권한을 제공하려면 `AWSPartnerCentralChannelHandshakeApprovalManagement` 정책을 IAM 자격 증명에 연결합니다. 자세한 내용은 [AWSAWS Partner Central 사용자를 위한 관리형 정책을 참조하세요](#).

AWSPartnerCentralChannelManagement 정책

이 정책은 Partner Central 채널 관리 작업에 대한 전체 액세스 권한을 부여합니다.

```

{
  "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 정책

이 정책은 채널 핸드셰이크 요청을 보고 응답할 수 있는 액세스 권한을 부여합니다. 이 정책은 채널 핸드셰이크 요청을 수신하는 AWS 계정의 역할에 적용해야 합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ChannelHandshakeManagement",
      "Effect": "Allow",
      "Action": [
        "partnercentral:ListChannelHandshakes",
        "partnercentral:AcceptChannelHandshake",
        "partnercentral:RejectChannelHandshake"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "partnercentral:Catalog": [
            "AWS",
            "Sandbox"
          ]
        }
      }
    }
  ]
}
```

IAM 역할 및 사용자에게 정책 할당

다음 단계에 따라 IAM 역할 및 사용자에게 정책을 할당합니다.

1. AWS Management Console에 로그인합니다.
2. IAM 서비스로 이동합니다.
3. 역할 또는 사용자를 선택하고 정책을 연결할 IAM 역할 또는 사용자를 선택합니다.
4. AWSPartnerCentralChannelManagement 정책 또는 사용자 지정 정책을 선택한 IAM 역할 또는 사용자에게 연결합니다.

자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

조건 키를 사용하여 권한 관리

IAM 정책의 조건 키는 문 정책을 적용할 시기에 대한 리소스 수준 권한을 제공합니다. 조건 키를 사용하여 특정 권한이 허용되거나 거부되는 시기를 지정하는 조건을 지정할 수 있습니다.

자세한 내용은 [IAM JSON 정책 요소: 조건 연산자](#)를 참조하세요.

조건 키 개요

조건 키	설명	적용 가능한 작업	유효값
partnercentral:Catalog	연결된 카탈로그 개체의 유형을 기준으로 액세스를 필터링합니다.	모든 채널 관리 작업	AWS, 샌드박스
partnercentral:ChannelHandshakeType	채널 핸드셰이크 유형에 따라 액세스를 필터링합니다.	CreateChannelHandshake, AcceptChannelHandshake, RejectChannelHandshake, CancelChannelHandshake, ListChannelHandshakes	PROGRAM_MANAGEMENT_ACCOUNT, START_SERVICE_PERIOD, REVOKE_SERVICE_PERIOD

필수 권한 요약

필수 권한 요약

작업	설명
partnercentral:CreateProgramManagementAccount	프로그램 관리 계정 생성 허용
partnercentral:UpdateProgramManagementAccount	프로그램 관리 계정 업데이트 허용
partnercentral>DeleteProgramManagementAccount	프로그램 관리 계정 삭제 허용

작업	설명
partnercentral:ListProgramManagementAccounts	프로그램 관리 계정 나열 허용
partnercentral:GetProgramManagementAccount	프로그램 관리 계정 세부 정보 검색 허용
partnercentral:CreateRelationship	관계 생성을 허용합니다.
partnercentral:UpdateRelationship	관계 업데이트 허용
partnercentral>DeleteRelationship	관계 삭제 허용
partnercentral:GetRelationship	관계 세부 정보 검색 허용
partnercentral:ListRelationships	관계 나열 허용
partnercentral:CreateChannelHandshake	채널 핸드셰이크 생성 허용
partnercentral:AcceptChannelHandshake	채널 핸드셰이크 수락 허용
partnercentral:RejectChannelHandshake	채널 핸드셰이크 거부 허용
partnercentral:CancelChannelHandshake	채널 핸드셰이크 취소 허용
partnercentral:ListChannelHandshakes	채널 핸드셰이크 나열 허용

수익 측정 API에 대한 액세스

액세스 제어 및 권한은 AWS Identity and Access Management(IAM)에서 관리합니다. 이 섹션에서는 AWS Marketplace 엔터티를 나열하는 데 필요한 권한을 포함하여 수익 측정 API와 상호 작용하는 데 필요한 권한을 구성하는 지침을 제공합니다.

사전 조건

권한을 구성하기 전에 AWS 계정이 Partner Central에 연결되어 있고 필요한 IAM 역할 및 사용자를 생성했는지 확인합니다. 자세한 내용은 [설정 및 인증](#)을 참조하세요.

AWS관리형 정책 사용

AWS는 수익 측정 API와 상호 작용하는 데 필요한 권한을 부여하는 관리형 정책을 제공합니다. 페르소나에 따라 두 가지 관리형 정책을 사용할 수 있습니다.

- 파트너 - `AWSPartnerCentralRevenueAttributionManagement` 정책을 IAM 자격 증명에 연결합니다.
- 고객 - `AWSRevenueAttributionManagement` 정책을 IAM 자격 증명에 연결합니다.

자세한 내용은 [AWSAWS Partner Central 사용자를 위한 관리형 정책을](#) 참조하세요.

AWSPartnerCentralRevenueAttributionManagement 정책

이 정책은AWS Partner Central에 등록된AWS계정의 수익 어트리뷰션 관리 활동에 필요한 액세스를 제공합니다.

```
{
  "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 정책

이 정책은 AWS Partner Central에 등록되지 않은 AWS 계정의 수익 어트리뷰션 관리 활동에 필요한 액세스를 제공합니다.

```

{
  "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"
            ]
        }
    }
}
]

```

}

IAM 역할 및 사용자에게 정책 할당

다음 단계에 따라 IAM 역할 및 사용자에게 정책을 할당합니다.

1. AWS Management Console에 로그인합니다.
2. IAM 서비스로 이동합니다.
3. 역할 또는 사용자를 선택하고 정책을 연결할 IAM 역할 또는 사용자를 선택합니다.
4. 선택한 IAM 역할 또는 사용자에게 `AWSPartnerCentralRevenueAttributionManagement` 정책 (파트너용) 또는 `AWSRevenueAttributionManagement` 정책(고객용)을 연결합니다.

자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

조건 키를 사용하여 권한 관리

IAM 정책의 조건 키는 문 정책을 적용할 시기에 대한 리소스 수준 권한을 제공합니다. 조건 키를 사용하여 특정 권한이 허용되거나 거부되는 시기를 지정하는 조건을 지정할 수 있습니다.

자세한 내용은 [IAM JSON 정책 요소: 조건 연산자](#)를 참조하세요.

조건 키 개요

조건 키	설명	적용 가능한 작업	유효값
<code>partnercentral:Catalog</code>	연결된 카탈로그 개체의 유형을 기준으로 액세스를 필터링합니다.	모든 작업	AWS, 샌드박스

필수 권한 요약

필수 권한 요약

작업	설명
<code>partnercentral:CreateRevenueAttribution</code>	새 수익 어트리뷰션 레코드 생성 허용
<code>partnercentral:UpdateRevenueAttribution</code>	기존 수익 어트리뷰션 레코드 업데이트 허용

작업	설명
partnercentral:GetRevenueAttribution	특정 수익 어트리뷰션의 세부 정보를 검색할 수 있습니다.
partnercentral:ListRevenueAttributions	선택적 필터를 사용하여 수익 어트리뷰션 나열 허용
partnercentral:GetRevenueAttributionAllocation	ID로 단일 할당 검색 허용
partnercentral:ListRevenueAttributionAllocations	필터링 지원을 통해 커밋된 할당 나열 허용
partnercentral:StartRevenueAttributionAllocationsTask	비동기 처리를 위해 최대 250개의 할당 변경 사항 배치 제출 허용
partnercentral:GetRevenueAttributionAllocationsTask	이전에 제출된 할당 작업의 상태를 검색할 수 있습니다.
partnercentral:CreateMarketplaceRevenueShare	새 마켓플레이스 수익 공유 리소스 생성 허용
partnercentral:GetMarketplaceRevenueShare	특정 마켓플레이스 수익 공유의 세부 정보 검색 허용
partnercentral:ListMarketplaceRevenueShares	선택적 필터를 사용하여 마켓플레이스 수익 공유 나열 허용
partnercentral:CreateMarketplaceRevenueShareAllocation	새 마켓플레이스 수익 공유 할당 생성 허용
partnercentral:GetMarketplaceRevenueShareAllocation	특정 마켓플레이스 수익 공유 할당의 세부 정보 검색 허용
partnercentral:UpdateMarketplaceRevenueShareAllocation	기존 마켓플레이스 수익 공유 할당 업데이트 허용
partnercentral:ListMarketplaceRevenueShareAllocations	마켓플레이스 수익 공유에 따른 할당 나열 허용

작업	설명
partnercentral:TagResource	리소스에 대한 태그 추가 또는 덮어쓰기 허용
partnercentral:UntagResource	리소스에서 태그 제거 허용
partnercentral:ListTagsForResource	리소스와 연결된 태그 나열 허용

AWS Partner Central API 할당량

다음 섹션에서는 서비스 할당량에 대한 정보를 제공합니다.

주제

- [AWS파트너 중앙 계정 API 할당량](#)
- [AWS Partner Central Selling API 할당량](#)
- [AWS Partner Central Benefits API 할당량](#)
- [AWS파트너 중앙 채널 API 할당량](#)
- [AWS파트너 중앙 수익 측정 API의 할당량](#)

AWS파트너 중앙 계정 API 할당량

AWS파트너 중앙 계정 API에는 다음과 같은 할당량이 있습니다.

추가 할당량

추가 할당량

표시 명칭	카탈로그	설명	기본값
계정당 연결 초대 열기	AWS	AWS카탈로그의 파트너 계정으로 유지할 수 있는 최대 미해결 연결 초대 수	1,000
계정당 활성 연결 수	AWS	AWS카탈로그의 파트너 계정과 유지할 수	1,000

표시 명칭	카탈로그	설명	기본값
		있는 최대 활성 연결 수	
파트너당 이메일 도메인	AWS	AWS카탈로그의AWS 훈련 인증을 위해 파트너 계정과 연결할 수 있는 최대 이메일 도메인 수	50
계정당 연결 초대 비율	AWS	AWS카탈로그에서 보낼 수 있는 일일 최대 연결 초대 수	50
계정당 프로필 업데이트 작업 속도	AWS	AWS카탈로그에서 시작할 수 있는 일일 최대 프로필 업데이트 작업 수	10
계정당 프로필 가시성 업데이트 속도	AWS	AWS카탈로그에서 시작할 수 있는 일일 최대 프로필 가시성 업데이트 수	5
이메일 주소당 이메일 확인 코드 비율	AWS	AWS카탈로그에서 요청할 수 있는 일일 최대 이메일 확인 코드 수	5
계정당 연결 초대 열기	샌드박스	샌드박스 카탈로그의 파트너 계정으로 유지할 수 있는 최대 미해결 연결 초대 수	1,000
계정당 활성 연결 수	샌드박스	샌드박스 카탈로그에서 파트너 계정으로 유지할 수 있는 최대 활성 연결 수	1,000

표시 명칭	카탈로그	설명	기본값
파트너당 이메일 도메인	샌드박스	샌드박스 카탈로그의 AWS훈련 인증을 위해 파트너 계정과 연결할 수 있는 최대 이메일 도메인 수	50
계정당 연결 초대 비율	샌드박스	샌드박스 카탈로그에서 보낼 수 있는 일일 최대 연결 초대 수	50
계정당 프로필 업데이트 작업 속도	샌드박스	샌드박스 카탈로그에서 시작할 수 있는 일일 최대 프로필 업데이트 작업 수	10
계정당 프로필 가시성 업데이트 속도	샌드박스	샌드박스 카탈로그에서 시작할 수 있는 일일 최대 프로필 가시성 업데이트 수	5
이메일 주소당 이메일 확인 코드 비율	샌드박스	샌드박스 카탈로그에서 요청할 수 있는 일일 최대 이메일 확인 코드 수입입니다. 샌드박스 카탈로그에서는 이메일이 전송되지 않습니다.	해당 사항 없음

할당량 이해 및 관리

속도 제한

API 속도 제한에 도달하면 서비스가 `ThrottlingException`으로 응답합니다. 속도 제한을 더 잘 처리하려면 애플리케이션에서 지수 백오프 및 재시도 전략을 구현하는 것이 좋습니다.

할당량 증가 요청

기본 할당량이 요구 사항을 충족하지 않는 경우 [Service Quotas 페이지](#)를 통해 할당량 증가를 요청할 수 있습니다. Service Quotas 콘솔은 서비스 할당량을 보고 관리하는 데 사용할 수 있는 브라우저 기반 인터페이스입니다. 상단 탐색 모음에서 Service Quotas를 선택하거나에서 Service Quotas를 검색하여 모든AWS Management Console페이지에서 Service Quotas에 액세스할 수 있습니다AWS Management Console.

AWS Partner Central Selling API 할당량

AWS Partner Central 판매 API는 할당량을 적용하여 공정한 사용을 보장하고 서비스가 오용되지 않도록 보호합니다. 다음은 기회당 다양한 API 작업 및 연결에 대한 자세한 할당량입니다.

API 작업 할당량

유형	API 작업	할당량(파트너 계정당)
읽기 작업	GetOpportunity	초당 10, 24시간당 100,000
	GetAwsOpportunitySummary	
	ListOpportunities	
	ListSolutions	
	GetEngagementInvitation	
	ListEngagementInvitations	
읽기 작업 예측	GetProspectingFromEngagementTask	100/초
읽기 작업 예측	ListProspectingFromEngagementTasks	50개/초
쓰기 작업	CreateOpportunity	초당 1, 24시간당 10,000
	UpdateOpportunity	
	AssociateOpportunity	

유형	API 작업	할당량(파트너 계정당)
	DisassociateOpportunity	
	RejectEngagementInvitation	
	AssignOpportunity	
	StartEngagementFromOpportunityTask	
	StartEngagementByAcceptingInvitationTask	
쓰기 작업 예측	StartProspectingFromEngagementTask	2/초
참여 쓰기 작업	CreateEngagement	초당 15개

기회당 연결 할당량

관련 엔터티	할당량
AWS제품	기회당 20개
파트너 솔루션	기회당 10개
AWS Marketplace솔루션	기회당 10개
AWS Marketplace제품	기회당 10개
AWS Marketplace비공개 제안	기회당 1개

할당량 이해 및 관리

속도 제한

API 속도 제한에 도달하면 서비스가 `ThrottlingException`으로 응답합니다. 속도 제한을 더 잘 처리하기 위해 AWS는 애플리케이션에서 지수 백오프 및 재시도 전략을 구현할 것을 권장합니다.

할당량 기간

일일 할당량은 24시간 동안 재설정됩니다. 예를 들어 지난 24시간 동안 10,000개의 쓰기 작업을 수행하고 10,001번째 요청을 수행하려고 하는 경우 요청이 제한됩니다. 의도하지 않은 제한을 방지하기 위해 애플리케이션의 사용 패턴을 고려해야 합니다.

할당량 증가 요청

기본 할당량이 요구 사항을 충족하지 않는 경우 [Service Quotas 페이지](#)를 통해 할당량 증가를 요청할 수 있습니다. Service Quotas 콘솔은 서비스 할당량을 보고 관리하는 데 사용할 수 있는 브라우저 기반 인터페이스입니다. 상단 탐색 모음에서 Service Quotas를 선택하거나에서 Service Quotas를 검색하여 모든AWS Management Console페이지에서 Service Quotas에 액세스할 수 있습니다AWS Management Console.

AWS Partner Central Benefits API 할당량

AWS Partner Central Benefits API에는 다음과 같은 할당량이 있습니다.

요청 할당량

요청 할당량

API 작업	할당량(AWS계정당)
ListBenefits	초당 20
GetBenefit	초당 20
AmendBenefitApplication	초당 10개
CancelBenefitApplication	초당 10개
CreateBenefitApplication	초당 10개
GetBenefitApplication	초당 20
ListBenefitApplications	초당 30
RecallBenefitApplication	초당 10개
SubmitBenefitApplication	초당 10개

API 작업	할당량(AWS계정당)
UpdateBenefitApplication	초당 10개
GetBenefitAllocation	초당 20
ListBenefitAllocations	초당 20
AssociateBenefitApplicationResource	초당 10개
DisassociateBenefitApplicationResource	초당 10개

추가 할당량

추가 할당량

표시 명칭	카탈로그	설명	기본값
혜택 애플리케이션	AWS	AWS카탈로그에서 생성할 수 있는 최대 혜택 애플리케이션 수	10,000
혜택 애플리케이션	샌드박스	샌드박스 카탈로그에서 생성할 수 있는 최대 혜택 애플리케이션 수	10,000

할당량 이해 및 관리

속도 제한

API 속도 제한에 도달하면 서비스가 `ThrottlingException`으로 응답합니다. 속도 제한을 더 잘 처리하기 위해 AWS는 애플리케이션에서 지수 백오프 및 재시도 전략을 구현할 것을 권장합니다.

할당량 증가 요청

기본 할당량이 요구 사항을 충족하지 않는 경우 [Service Quotas 페이지](#)를 통해 할당량 증가를 요청할 수 있습니다. Service Quotas 콘솔은 서비스 할당량을 보고 관리하는 데 사용할 수 있는 브라우저 기반 인터페이스입니다. 상단 탐색 모음에서 Service Quotas를 선택하거나에서 Service Quotas를

검색하여 모든 AWS Management Console 페이지에서 Service Quotas에 액세스할 수 있습니다 AWS Management Console.

AWS파트너 중앙 채널 API 할당량

AWS파트너 중앙 채널 API에는 다음과 같은 할당량이 있습니다.

요청 할당량

요청 할당량

API 작업	할당량(AWS계정당)
CreateProgramManagementAccount	초당 3개
UpdateProgramManagementAccount	초당 3개
ListProgramManagementAccounts	초당 5
DeleteProgramManagementAccount	초당 3개
CreateChannelHandshake	초당 3개
ListChannelHandshakes	초당 5
AcceptChannelHandshake	초당 3개
RejectChannelHandshake	초당 3개
CancelChannelHandshake	초당 3개
CreateRelationship	초당 3개
GetRelationship	초당 5
ListRelationships	초당 5
UpdateRelationship	초당 3개
DeleteRelationship	초당 3개

추가 할당량

추가 할당량

표시 명칭	카탈로그	설명	기본값
프로그램 관리 계정	AWS	AWS카탈로그에서 생성할 수 있는 최대 프로그램 관리 계정 수	50
프로그램 관리 계정당 관계	AWS	AWS카탈로그의 프로그램 관리 계정당 설정할 수 있는 최대 관계 수	1,000
계정당 프로그램 관리 계정 핸드셰이크	AWS	AWS카탈로그의 최대 활성 프로그램 관리 계정 핸드셰이크 수	100
계정당 서비스 기간 핸드셰이크 시작	AWS	AWS카탈로그에서 서비스 기간을 설정하기 위한 최대 활성 핸드셰이크 수	200
계정당 서비스 기간 핸드셰이크 취소	AWS	AWS카탈로그에서 서비스 기간을 취소하기 위한 최대 활성 핸드셰이크 수	200
일일 프로그램 관리 계정 핸드셰이크 비율	AWS	AWS카탈로그에서 하루에 동일한 계정으로 보낼 수 있는 프로그램 관리AWS계정 핸드셰이크의 최대 수	5
일일 시작 서비스 기간 핸드셰이크 비율	AWS	AWS카탈로그에서 매일 동일한AWS계정으로 보낼 수 있는 최대	5

표시 명칭	카탈로그	설명	기본값
		시작 서비스 기간 핸드셰이크 수	
일일 서비스 기간 핸드셰이크 취소 비율	AWS	AWS카탈로그에서 하루에 동일한AWS계정으로 보낼 수 있는 서비스 기간 핸드셰이크 취소의 최대 수	5
프로그램 관리 계정	샌드박스	샌드박스 카탈로그에서 생성할 수 있는 최대 프로그램 관리 계정 수	50
프로그램 관리 계정당 관계	샌드박스	샌드박스 카탈로그에서 프로그램 관리 계정당 설정할 수 있는 최대 관계 수	1,000
계정당 프로그램 관리 계정 핸드셰이크	샌드박스	샌드박스 카탈로그의 최대 활성 프로그램 관리 계정 핸드셰이크 수	100
계정당 서비스 기간 핸드셰이크 시작	샌드박스	샌드박스 카탈로그에서 서비스 기간을 설정하기 위한 최대 활성 핸드셰이크 수	200
계정당 서비스 기간 핸드셰이크 취소	샌드박스	샌드박스 카탈로그에서 서비스 기간 취소를 위한 최대 활성 핸드셰이크 수	200

표시 명칭	카탈로그	설명	기본값
일일 프로그램 관리 계정 핸드셰이크 비율	샌드박스	샌드박스 카탈로그에서 매일 동일한 계정으로 보낼 수 있는 프로그램 관리AWS계정 핸드셰이크의 최대 수	5
일일 시작 서비스 기간 핸드셰이크 비율	샌드박스	샌드박스 카탈로그에서 매일 동일한AWS계정으로 보낼 수 있는 최대 시작 서비스 기간 핸드셰이크 수	5
일일 서비스 기간 핸드셰이크 취소 비율	샌드박스	샌드박스 카탈로그에서 하루에 동일한AWS계정으로 보낼 수 있는 서비스 기간 핸드셰이크 취소의 최대 수	5

할당량 이해 및 관리

속도 제한

API 속도 제한에 도달하면 서비스가 ThrottlingException으로 응답합니다. 속도 제한을 더 잘 처리하기 위해 AWS는 애플리케이션에서 지수 백오프 및 재시도 전략을 구현할 것을 권장합니다.

할당량 증가 요청

기본 할당량이 요구 사항을 충족하지 않는 경우 [Service Quotas 페이지](#)를 통해 할당량 증가를 요청할 수 있습니다. Service Quotas 콘솔은 서비스 할당량을 보고 관리하는 데 사용할 수 있는 브라우저 기반 인터페이스입니다. 상단 탐색 모음에서 Service Quotas를 선택하거나에서 Service Quotas를 검색하여 모든AWS Management Console페이지에서 Service Quotas에 액세스할 수 있습니다AWS Management Console.

AWS파트너 중앙 수익 측정 API의 할당량

AWS파트너 중앙 수익 측정 API는 할당량을 적용하여 공정한 사용을 보장하고 서비스가 오용되지 않도록 보호합니다. 다음은 다양한 API 작업에 대한 자세한 할당량입니다.

API 작업 할당량

유형	API 작업	할당량(파트너 계정당)
읽기 작업	GetRevenueAttribution	50개/초
	GetRevenueAttributionAllocation	
	GetRevenueAttributionAllocationsTask	
	GetMarketplaceRevenueShare	
	GetMarketplaceRevenueShareAllocation	
작업 나열	ListRevenueAttributions	초당 10개
	ListRevenueAttributionAllocations	
	ListMarketplaceRevenueShares	
	ListMarketplaceRevenueShareAllocations	
	ListTagsForResource	
쓰기 작업	CreateRevenueAttribution	2/초
	UpdateRevenueAttribution	
	CreateMarketplaceRevenueShare	
	CreateMarketplaceRevenueShareAllocation	

유형	API 작업	할당량(파트너 계정당)
	UpdateMarketplaceRevenueShareAllocation	
	TagResource	
	UntagResource	
비동기 쓰기 작업	StartRevenueAttributionAllocationsTask	초당 1

할당량 이해 및 관리

속도 제한

API 속도 제한에 도달하면 서비스가 `ThrottlingException`으로 응답합니다. 속도 제한을 더 잘 처리하려면 애플리케이션에서 지수 백오프 및 재시도 전략을 구현하는 것이 좋습니다.

할당량 증가 요청

기본 할당량이 요구 사항을 충족하지 않는 경우 [Service Quotas 페이지](#)를 통해 할당량 증가를 요청할 수 있습니다. Service Quotas 콘솔은 서비스 할당량을 보고 관리하는 데 사용할 수 있는 브라우저 기반 인터페이스입니다. 상단 탐색 모음에서 Service Quotas를 선택하거나에서 Service Quotas를 검색하여 모든AWS Management Console페이지에서 Service Quotas에 액세스할 수 있습니다AWS Management Console.

AWS Partner Central API에 대한 로깅

다음 섹션에서는 로깅에 대한 정보를 제공합니다.

주제

- [AWS파트너 중앙 계정 API 로깅](#)
- [AWS Partner Central Selling API 로깅](#)
- [AWS Partner Central Benefits API 로깅](#)
- [AWS파트너 중앙 채널 API 로깅](#)
- [AWS파트너 중앙 수익 속성 API 로깅](#)

AWS파트너 중앙 계정 API 로깅

[AWS CloudTrail](#)는에 대한 거버넌스, 규정 준수, 운영 감사 및 위험 감사를 지원하는 서비스입니다AWS 계정.AWS CloudTrail를 사용하면AWS인프라 전반의 작업과 관련된 계정 활동을 로깅, 지속적인 모니터링 및 유지할 수 있습니다.AWS Partner Central Account API 활동은 CloudTrail에 이벤트로 기록됩니다. Amazon S3 버킷에 로그 파일로 이벤트를 전송할 수 있는 구성인 추적을 생성할 수 있습니다.

개요

AWS Partner Central Account API는AWS Partner Central에서 사용자AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는AWS서비스와 통합됩니다. CloudTrail은AWS파트너 중앙 계정 API에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는AWS Partner Central의 호출과AWS Partner Central Account API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면AWS파트너 중앙 계정 API에 대한 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여AWS Partner Central Account API에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

AWS파트너 중앙 계정 API 로그 파일 항목 이해

추적은 이벤트를 Amazon S3 버킷에 로그 파일로 전송할 수 있는 구성입니다. 추적이AWS파트너 중앙 계정 API 이벤트를 추적하면 CloudTrail은 모든 리전에서 이벤트를 로그 파일로 처리합니다. 각 로그 파일에는 하나 이상의 이벤트가 포함될 수 있습니다.

다음 예제는AWS파트너 중앙 계정 API에 대한 CreatePartner 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
  "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"
  }
}

```

이 예제에서는 CloudTrailTestUser라는 IAM 사용자가 CreatePartner 작업을 호출했습니다. 이 요청은 2025년 11월 7일 19:45:58 UTC에 이루어졌습니다. 요청이 법적 이름이 "ExampleLegalName"인 AWS 카탈로그EXAMPLE_PARTNER_ID의 ID가 인 새 파트너를 생성했습니다.

AWS파트너 중앙 계정 API 로그 파일 항목의 필드

CloudTrail 로그 파일의 각 항목에는 누가 요청했는지, 요청에서 조치를 취한 리소스, AWS Partner Central Account API에서 반환한 응답 요소에 대한 정보가 포함되어 있습니다. , eventVersion, userIdentity 및와 같은 로그 항목의 필드 목록은 작업에 대한 자세한 정보를 eventTime 제공합니다. 예를 들어 sourceIPAddress 필드에는 요청이 수행된 IP 주소가 표시됩니다.

AWS Partner Central Selling API 로깅

[AWS CloudTrail](#)는에 대한 거버넌스, 규정 준수, 운영 감사 및 위험 감사를 지원하는 서비스입니다AWS 계정.AWS CloudTrail를 사용하면AWS인프라 전반의 작업과 관련된 계정 활동을 로깅, 지속적인 모니터링 및 유지할 수 있습니다.AWS Partner Central API 활동은 CloudTrail에 이벤트로 기록됩니다. Amazon S3 버킷에 로그 파일로 이벤트를 전송할 수 있는 구성인 추적을 생성할 수 있습니다.

개요

AWS Partner Central API는AWS Partner Central에서 사용자AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는AWS서비스와 통합됩니다. CloudTrail은AWS Partner Central에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는AWS Partner Central의 호출과AWS Partner Central API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면AWS Partner Central 이벤트를 포함한 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여AWS Partner Central에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

AWS Partner Central Selling API 로그 파일 항목 이해

추적은 이벤트를 Amazon S3 버킷에 로그 파일로 전송할 수 있는 구성입니다. 추적이AWS Partner Central 이벤트를 추적하면 CloudTrail은 모든 리전에서 이벤트를 로그 파일로 처리합니다. 각 로그 파일에는 하나 이상의 이벤트가 포함될 수 있습니다.

다음 예제에서는AWS Partner Central에 대한 ListOpportunities 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
  "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"
}

```

이 예제에서는 CloudTrailTestUser라는 IAM 사용자가 ListOpportunities 작업을 호출했습니다. 작업은 us-east-1에서 호출되었으며AWS 리전요청은 2023년 10월 17일 21:49:23 UTC에 이루어졌습니다.

AWS Partner Central Selling API 로그 파일 항목의 필드

CloudTrail 로그 파일의 각 항목에는 누가 요청했는지, 요청에서 조치를 취한 리소스, AWS Partner Central에서 반환한 응답 요소에 대한 정보가 포함되어 있습니다. , eventVersion userIdentity 및와 같은 로그 항목의 필드 목록은 작업에 대한 자세한 정보를 eventTime 제공합니다. 예를 들어 sourceIPAddress 필드에는 요청이 수행된 IP 주소가 표시됩니다.

AWS Partner Central Benefits API 로깅

[AWS CloudTrail](#)는에 대한 거버넌스, 규정 준수, 운영 감사 및 위험 감사를 지원하는 서비스입니다AWS 계정.AWS CloudTrail를 사용하면AWS인프라 전반의 작업과 관련된 계정 활동을 로깅, 지속적인 모니터링 및 유지할 수 있습니다.AWS Partner Central Benefits API 활동은 CloudTrail에 이벤트로 기록됩니다. Amazon S3 버킷에 로그 파일로 이벤트를 전송할 수 있는 구성인 추적을 생성할 수 있습니다.

개요

AWS Partner Central Benefits API는AWS Partner Central에서 사용자AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는AWS서비스와 통합됩니다. CloudTrail은AWS Partner Central Benefits API에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는AWS Partner Central의 호출과AWS Partner Central Benefits API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면 AWS Partner Central Benefits API에 대한 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여 AWS Partner Central Benefits API에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

AWS Partner Central Benefits API 로그 파일 항목 이해

추적은 이벤트를 Amazon S3 버킷에 로그 파일로 전송할 수 있는 구성입니다. 추적이 AWS Partner Central Benefits API 이벤트를 추적하면 CloudTrail은 모든 리전에서 이벤트를 로그 파일로 처리합니다. 각 로그 파일에는 하나 이상의 이벤트가 포함될 수 있습니다.

다음 예제는 AWS Partner Central Benefits API에 대한 ListBenefitApplications 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
  "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"
}

```

이 예제에서는 Alice라는 IAM 사용자가 ListBenefitApplications 작업을 호출했습니다. 이 요청은 2025년 10월 21일 03:10:59 UTC에 이루어졌습니다. 나열된 요청은 AWS 카탈로그에 대한 혜택 애플리케이션을 제공합니다.

AWS Partner Central Benefits API 로그 파일 항목의 필드

CloudTrail 로그 파일의 각 항목에는 누가 요청했는지, 요청에서 조치를 취한 리소스, AWS Partner Central Benefits API에서 반환한 응답 요소에 대한 정보가 포함되어 있습니다. , eventVersion, userIdentity 및와 같은 로그 항목의 필드 목록은 작업에 대한 자세한 정보를 eventTime 제공합니다. 예를 들어 sourceIPAddress 필드에는 요청이 수행된 IP 주소가 표시됩니다.

AWS파트너 중앙 채널 API 로깅

[AWS CloudTrail](#)는에 대한 거버넌스, 규정 준수, 운영 감사 및 위험 감사를 지원하는 서비스입니다. AWS 계정. AWS CloudTrail를 사용하면 AWS 인프라 전반의 작업과 관련된 계정 활동을 로깅, 지속적인 모니터링 및 유지할 수 있습니다. AWS Partner Central Channel API 활동은 CloudTrail에 이벤트로 기록됩니다. Amazon S3 버킷에 로그 파일로 이벤트를 전송할 수 있는 구성인 추적을 생성할 수 있습니다.

개요

AWS Partner Central Channel API는 AWS Partner Central에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 AWS파트너 중앙 채널 API에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 AWS Partner Central의 호출과 AWS Partner Central Channel API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면 AWS Partner Central Channel API에 대한 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여 AWS Partner Central Channel API에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

AWS파트너 중앙 채널 API 로그 파일 항목 이해

추적은 이벤트를 Amazon S3 버킷에 로그 파일로 전송할 수 있는 구성입니다. 추적이 AWS파트너 중앙 채널 API 이벤트를 추적하면 CloudTrail은 모든 리전에서 이벤트를 로그 파일로 처리합니다. 각 로그 파일에는 하나 이상의 이벤트가 포함될 수 있습니다.

다음 예제에서는 AWS Partner Central Channel API에 대한 CreateProgramManagementAccount 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
  "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"
}
}

```

이 예제에서는 위임된 역할 세션을 통해 ExampleRole이라는 IAM 역할이 CreateProgramManagementAccount 작업을 호출했습니다. 이 요청은 2025년 10월 21일 17:07:26 UTC에 이루어졌습니다. 이 요청은 솔루션 공급자 프로그램의 ID가 인 새 프로그램 관리 계정을 생성 pma-example123456789했습니다.

AWS파트너 중앙 채널 API 로그 파일 항목의 필드

CloudTrail 로그 파일의 각 항목에는 누가 요청했는지, 요청에서 조치를 취한 리소스, AWS Partner Central Channel API에서 반환한 응답 요소에 대한 정보가 포함되어 있습니다. , eventVersion, userIdentity 및와 같은 로그 항목의 필드 목록은 작업에 대한 자세한 정보를 eventTime 제공합니다. 예를 들어 sourceIPAddress 필드에는 요청이 수행된 IP 주소가 표시됩니다.

AWS파트너 중앙 수익 속성 API 로깅

[AWS CloudTrail](#)는의 거버넌스, 규정 준수, 운영 감사 및 위험 감사를 지원하는 서비스입니다. AWS 계정. AWS CloudTrail를 사용하면 AWS 인프라 전반의 작업과 관련된 계정 활동을 로깅, 지속적인 모니터링 및 유지할 수 있습니다. AWS파트너 중앙 수익 측정 API 활동은 CloudTrail에 이벤트로 기록됩니다. Amazon S3 버킷에 로그 파일로 이벤트를 전송할 수 있는 구성인 추적을 생성할 수 있습니다.

개요

AWS Partner Central 수익 측정 API는 AWS Partner Central에서 사용자 AWS CloudTrail, 역할 또는 서비스가 수행한 작업에 대한 레코드를 제공하는 AWS 서비스와 통합됩니다. CloudTrail은 AWS파트너 중앙 수익 속성에 대한 모든 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 AWS Partner Central 콘솔의 호출과 AWS Partner Central 수익 측정 API 작업에 대한 코드 호출이 포함됩니다.

추적을 생성하면 AWS파트너 중앙 수익 측정 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다.

CloudTrail에서 수집한 정보를 사용하여 AWS파트너 중앙 수익 측정에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

다음 AWS파트너 중앙 수익 측정 API 작업은 CloudTrail에 로깅됩니다.

- CreateRevenueAttribution
- UpdateRevenueAttribution
- GetRevenueAttribution
- ListRevenueAttributions
- TagResource
- UntagResource
- ListTagsForResource

- CreateMarketplaceRevenueShare
- GetMarketplaceRevenueShare
- ListMarketplaceRevenueShares
- CreateMarketplaceRevenueShareAllocation
- GetMarketplaceRevenueShareAllocation
- UpdateMarketplaceRevenueShareAllocation
- ListMarketplaceRevenueShareAllocations
- StartRevenueAttributionAllocationsTask
- GetRevenueAttributionAllocationsTask
- ListRevenueAttributionAllocations
- GetRevenueAttributionAllocation

AWS파트너 중앙 수익 측정 API 로그 파일 항목 이해

추적은 이벤트를 Amazon S3 버킷에 로그 파일로 전송할 수 있는 구성입니다. 추적이AWS파트너 중앙 수익 측정 이벤트를 추적하면 CloudTrail은 모든 리전에서 이벤트를 로그 파일로 처리합니다. 각 로그 파일에는 하나 이상의 이벤트가 포함될 수 있습니다.

다음 예제는AWS파트너 중앙 수익 속성에 대한 CreateRevenueAttribution 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다.

```
{
  "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-ee2e-4578-8db0-01ee2e34ee56",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}

```

이 예제에서는 CloudTrailTestUser라는 IAM 사용자가 CreateRevenueAttribution 작업을 호출했습니다. us-east-1AWS 리전에서 작업이 호출되었으며 2026년 6월 2일 11:53:54 UTC에 요청이 이루어졌습니다. ID를 사용하여 새로운 수익 어트리뷰션 리소스가 생성되었습니다. ra-0123456789abcdef0.

AWS파트너 중앙 수익 측정 API 로그 파일 항목의 필드

CloudTrail 로그 파일의 각 항목에는 누가 요청했는지, 요청에서 조치를 취한 리소스, AWS Partner Central Revenue Measurement에서 반환한 응답 요소에 대한 정보가 포함되어 있습니다. , eventVersion userIdentity 및와 같은 로그 항목의 필드 목록은 작업에 대한 자세한 정보를 eventTime 제공합니다. 예를 들어 sourceIPAddress 필드에는 요청이 수행된 IP 주소가 표시됩니다.

AWS Partner Central API에 대한 알림

다음 섹션에서는 알림에 대한 정보를 제공합니다.

주제

- [AWS파트너 중앙 계정 API에 대한 알림](#)
- [AWS Partner Central Selling API에 대한 알림](#)

- [AWS Partner Central Benefits API에 대한 알림](#)
- [AWS파트너 중앙 채널 API에 대한 알림](#)

AWS파트너 중앙 계정 API에 대한 알림

파트너 계정 연결 알림을 통해 파트너는 비즈니스 관계 및 운영 워크플로에 직접적인 영향을 미치는 연결 수명 주기 이벤트에 대한 정보를 얻을 수 있습니다. 이러한 이벤트는 파트너가 다음을 수행하는 데 매우 중요합니다.

- 새로운 공동 작업에 즉시 대응
- 연결 포트폴리오 상태에 대한 인식 유지
- 연결이 설정되거나 종료될 때 적절한 조치를 취합니다.
- 관계가 변경되는 시점을 파악하여 비즈니스 연속성 보장

주제

- [이벤트를 모니터링하기 위한 사전 조건 완료](#)
- [이벤트를 모니터링하도록 Amazon EventBridge 구성](#)
- [계정 연결 API 이벤트에 대해 자세히 알아보기](#)

이벤트를 모니터링하기 위한 사전 조건 완료

계정 연결 API에서 게시한 이벤트에 액세스하고 관리하려면 사용자에게 적절한 IAM 권한이 필요합니다. EventBridge에 사용할 수 있는 작업, 리소스 및 조건 키에 대한 자세한 내용은 [Amazon EventBridge 사용 설명서의 Amazon EventBridge에서 IAM 정책 조건 사용](#)을 참조하세요. EventBridge 조건 키 중 하나는 특정 이벤트 유형에 대한 권한 범위를 지정하는 데 사용할 수 events:detail-type이 있는입니다.

다음 예제 정책은 제안된 이벤트에 대한 권한을 사용자 지정하고 범위를 지정하는 방법을 보여줍니다. AllowPutRuleForPartnercentralAccountEvents 문은 규칙 생성을 허용하지만 aws.partnercentral-account 소스의 이벤트에 대해서만 가능합니다.

자세한 IAM 정책 예제는 EventBridge 권한에 대한 AWS 설명서를 참조하세요.

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

```
{
  "Sid": "AllowPutRuleForPartnerCentralAccountEvents",
  "Effect": "Allow",
  "Action": "events:PutRule",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "events:source": "aws.partnercentral-account.connection"
    }
  }
}
```

이벤트를 모니터링하도록 Amazon EventBridge 구성

계정 연결 API 이벤트를 모니터링하려면 캡처하려는 이벤트와 일치하는 EventBridge 규칙을 생성합니다. AWS Management Console 또는 AWS SDKs를 사용하여 규칙을 생성하고 관리할 수 있습니다. 다음 섹션에서는 두 방법을 모두 사용하여 규칙을 생성하는 방법을 설명합니다. 사용하는 방법에 관계없이 미국 동부(버지니아 북부) us-east-1 리전에서 규칙을 생성해야 합니다.

AWS Management Console 설정

를 사용하여 EventBridge 규칙을 설정하려면 [Amazon EventBridge 사용 설명서의 Amazon EventBridge에서 이벤트에 반응하는 규칙 생성](#)의 단계를 AWS Management Console 따릅니다.

EventBridge 규칙을 생성할 때 이벤트 버스를 기본값으로 설정하고 미국 동부(버지니아 북부) us-east-1 리전에서 규칙을 생성해야 합니다.

다음은 이벤트 규칙의 예입니다.

```
{
  "source": ["aws.partnercentral-account"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS SDK 설정

AWS SDKs를 사용하여 EventBridge 규칙을 프로그래밍 방식으로 생성하고 관리할 수 있습니다. 자세한 내용은 Amazon EventBridge API 참조의 [PutRule](#)을 참조하세요.

다음 예제에서는 AWS SDK for Python(Boto3)을 사용합니다.

```
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'])
```

계정 연결 API 이벤트에 대해 자세히 알아보기

다음 섹션에서는 계정 연결 API 이벤트 유형, 이를 트리거하는 시나리오 및 이벤트 예제를 설명합니다.

이벤트 유형

다음은 이벤트 유형과 트리거입니다.

- [파트너 연결 초대 수신](#): 다른 파트너가 연결을 설정하려고 함을 수신자에게 알립니다.
- [파트너 연결 초대 수락](#): 발신자에게 초대가 수락되었으며 이제 연결이 활성 상태임을 알립니다.
- [파트너 연결 초대가 거부됨](#): 발신자에게 초대가 거부되었음을 알립니다.
- [파트너 연결 취소됨](#): 활성 연결이 종료되면 다른 연결된 참가자에게 알립니다.
- [파트너 연결 초대 취소됨](#): 발신자가 조치를 취하기 전에 초대를 취소했음을 수신자에게 알립니다.
- [파트너 연결 초대 만료](#): 발신자와 수신자 모두에게 초대가 만료되었음을 알립니다.

이벤트 예제

다음 섹션에서는 이전 섹션의 앞부분에 나열된 이벤트의 예를 제공합니다.

주제

- [파트너 연결 초대 수신됨](#)

- [파트너 연결 초대 수락됨](#)
- [파트너 연결 초대가 거부됨](#)
- [파트너 연결 취소됨](#)
- [파트너 연결 초대 취소됨](#)
- [파트너 연결 초대 만료됨](#)

파트너 연결 초대 수신됨

트리거 컨텍스트

이 이벤트는 연결 초대가 성공적으로 생성되어 CreateConnectionInvitation API를 통해 PAC의 데이터 스토어에 지속될 때 생성됩니다. 이벤트는 초대 생성이 성공적으로 완료된 후 즉시 전송되며 API 호출이 이루어진 시점뿐만 아니라 PAC의 데이터 스토어에도 유지됩니다.

새 연결 초대가 생성되면 이벤트가 자동으로 전송됩니다. 동일한 초대에 대한 중복 이벤트 없이 생성된 초대당 하나의 이벤트입니다.

수신자

연결 초대에 있는 수신자의AWS계정입니다.

예상 처리

일반적인 고객 작업:

- 즉시 알림: 새로운 파트너십 기회에 대해 비즈니스 팀에 알림 트리거
- 워크플로 자동화: 보류 중인 초대를 사용하여 파트너 관리 시스템 자동 업데이트
- 의사 결정 지원: 연결 유형에 따라 적절한 의사 결정자에게 초대 전달
- 감사 로깅: 규정 준수 및 파트너십 추적을 위한 초대장 수신 기록
- 응답 자동화: 신뢰할 수 있는 파트너의 경우 잠재적으로 특정 초대 유형을 자동 수락합니다.

일반적인 통합 패턴:

- Lambda 함수 → 파트너 포털 대시보드 업데이트
- SQS 대기열 → 검토를 위한 배치 프로세스 초대
- SNS 주제 → 비즈니스 팀에 이메일/슬랙 알림
- API 대상 → Webhook에서 외부 CRM/파트너 관리 시스템으로

예제

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

파트너 연결 초대 수락됨

트리거 컨텍스트

이 이벤트는 연결 초대가 성공적으로 수락되고 연결이 생성되어 AcceptConnectionInvitation API를 통해 PAC의 데이터 스토어에 유지될 때 생성됩니다. 초대 수락이 성공적으로 완료되고 연결이 설정되면 이벤트가 즉시 전송됩니다.

연결 초대가 수락되면 이벤트가 자동으로 전송됩니다. 동일한 수락에 대한 중복 이벤트 없이 초대당 하나의 이벤트가 수락됩니다.

수신자

연결 초대에 관련한 두AWS계정 모두 원래 초대를 보낸 발신자 계정과 초대를 수락한 수신자 계정입니다.

예상 처리

일반적인 고객 작업:

- 파트너십 활성화: 워크플로를 트리거하여 계층 2 비즈니스 활동 활성화
- 상태 업데이트: 파트너 관리 시스템을 활성 연결 상태로 업데이트
- 알림: 성공적인 파트너십 수립에 대해 비즈니스 팀에 알림
- 액세스 프로비저닝: 공동 작업에 적절한 권한 자동 부여
- 분석: 파트너십 전환율 및 성공 지표 추적

일반적인 통합 패턴:

- Lambda 함수 → 데이터 공유 권한 활성화 및 파트너 포털 업데이트
- SQS 대기열 → 비즈니스 설정을 위한 배치 프로세스 연결 활성화
- SNS 주제 → 비즈니스 및 기술 팀에 이메일/슬랙 알림
- API 대상 → 파트너십 상태를 업데이트하기 위해 CRM 시스템으로 Webhook

예제

```
{
  "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"
    }
}
}

```

파트너 연결 초대가 거부됨

트리거 컨텍스트

이 이벤트는 RejectConnectionInvitation API를 통해 연결 초대가 성공적으로 거부될 때 생성됩니다. 이벤트는 초대 거부가 처리되고 PAC의 데이터 스토어에 지속된 직후 전송됩니다.

연결 초대가 거부되면 이벤트가 자동으로 전송됩니다. 초대당 하나의 이벤트가 거부되고 동일한 거부 에 대한 중복 이벤트가 없습니다.

수신자

원래 연결 초대를 보낸 AWS 계정(발신자 계정).

예상 처리

일반적인 고객 작업:

- 상태 업데이트: 거부 상태로 파트너 관리 시스템 업데이트
- 후속 조치: 대체 파트너십 접근 방식을 위한 워크플로 트리거
- 알림: 비즈니스 팀에 파트너십 결정에 대해 알림
- 정리: 내부 시스템에서 보류 중인 초대 참조 제거

일반적인 통합 패턴:

- Lambda 함수 → 파트너 포털 업데이트 및 후속 워크플로 트리거
- SQS 대기열 → 비즈니스 분석을 위한 배치 프로세스 거부
- SNS 주제 → 비즈니스 개발 팀에 이메일/슬랙 알림
- API 대상 → 기회 상태를 업데이트하기 위해 Webhook에서 CRM 시스템으로

예제

```
{
```

```

"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-****"
  }
}
}

```

파트너 연결 취소됨

트리거 컨텍스트

이 이벤트는 활성 연결이 성공적으로 취소되고 CancelConnection API를 통해 PAC의 데이터 스토어에서 활성 상태가 종료되도록 업데이트될 때 생성됩니다. 연결 취소가 처리되고 연결 상태가 업데이트되면 이벤트가 즉시 전송됩니다.

연결이 종료되면 이벤트가 자동으로 전송됩니다. 동일한 취소에 대한 중복 이벤트 없이 연결당 하나의 이벤트가 취소되었습니다.

수신자

연결에 있는 다른 참가자의AWS계정(취소를 시작한 참가자가 아님).

예상 처리

일반적인 고객 작업:

- 액세스 취소: 권한을 즉시 취소하고 데이터 공유를 비활성화합니다.
- 상태 업데이트: 종료된 연결 상태로 파트너 관리 시스템 업데이트

- 알림: 비즈니스 및 기술 팀에 파트너십 종료 알림
- 정리: 연결 참조 제거 및 공유 리소스 정리
- 분석: 연결 기간 및 종료 패턴 추적

일반적인 통합 패턴:

- Lambda 함수 → 권한 취소 및 파트너 포털 상태 업데이트
- SQS 대기열 → 정리 워크플로를 위한 배치 프로세스 연결 종료
- SNS 주제 → 비즈니스 및 기술 팀에 이메일/슬랙 알림
- API 대상 → 파트너십 상태를 업데이트하기 위해 Webhook에서 CRM 시스템으로

예제

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

파트너 연결 초대 취소됨

트리거 컨텍스트

이 이벤트는 보류 중인 연결 초대가 CancelConnectionInvitation API를 통해 성공적으로 취소될 때 생성됩니다. 초대 취소가 처리되고 초대 상태가 취소됨으로 업데이트되면 이벤트가 즉시 전송됩니다.

연결 초대가 취소되면 이벤트가 자동으로 전송됩니다. 동일한 취소에 대한 중복 이벤트 없이 연결 초대 당 하나의 이벤트가 취소되었습니다.

수신자

연결 초대를 수신해야 하는 AWS 계정(수신자 계정).

예상 처리

일반적인 고객 작업:

- 상태 업데이트: 보류 중인 초대 참조를 제거하도록 파트너 관리 시스템 업데이트
- 알림: 비즈니스 팀에 잠재적 파트너십 기회가 취소되었음을 알립니다.
- 정리: 내부 추적 시스템에서 초대 참조 제거
- 분석: 파트너십 인사이트를 위한 초대 취소 패턴 추적

일반적인 통합 패턴:

- Lambda 함수 → 파트너 포털 업데이트 및 보류 중인 초대 알림 제거
- SQS 대기열 → 비즈니스 분석을 위한 배치 프로세스 취소
- SNS 주제 → 비즈니스 개발 팀에 이메일/슬랙 알림
- API 대상 → 기회 상태를 업데이트하기 위해 CRM 시스템으로 Webhook

예제

```
{
  "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-*****"
  }
}
}

```

파트너 연결 초대 만료됨

트리거 컨텍스트

이 이벤트는 보류 중인 연결 초대가 수신자가 수락하거나 거부하지 않고 ExpiresAt 타임스탬프에 도달한 후 자동으로 만료될 때 생성됩니다. 초대 만료 시간에 도달하면 시스템에서 이벤트가 자동으로 트리거됩니다.

연결 초대가 만료되면 이벤트가 자동으로 전송됩니다. 초대당 하나의 이벤트가 동일한 만료 기간 동안 중복 이벤트 없이 만료되었습니다.

수신자

연결 초대에 참여한 두 AWS 계정 모두 원래 초대를 보낸 발신자 계정과 초대를 받은 수신자 계정입니다.

예상 처리

일반적인 고객 작업:

- 상태 업데이트: 초대를 만료됨으로 표시하도록 파트너 관리 시스템 업데이트
- 후속 조치: 재참여 또는 대체 파트너십 접근 방식을 위한 워크플로 트리거
- 알림: 비즈니스 팀에 누락된 파트너십 기회에 대해 알림
- 정리: 내부 추적 시스템에서 만료된 초대 참조 제거

일반적인 통합 패턴:

- Lambda 함수 → 파트너 포털 업데이트 및 후속 워크플로 트리거
- SQS 대기열 → 비즈니스 분석을 위한 배치 프로세스 만료
- SNS 주제 → 비즈니스 개발 팀에 이메일/슬랙 알림
- API 대상 → 기회 상태를 업데이트하기 위해 CRM 시스템으로 Webhook

예제

```
{
  "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-****"
    }
  }
}
```

AWS Partner Central Selling API에 대한 알림

판매 API 이벤트는 상태 변경 또는 기회 세부 정보에 대한 실시간 알림을 제공합니다. 이러한 이벤트는 시스템을 AWS Partner Central과 동기화하고 적시에 응답 및 업데이트를 보장하는 데 도움이 됩니다.

주제

- [이벤트를 모니터링하기 위한 사전 조건 완료](#)
- [이벤트를 모니터링하도록 Amazon EventBridge 구성](#)
- [API 이벤트 판매에 대해 자세히 알아보기](#)

이벤트를 모니터링하기 위한 사전 조건 완료

사용자는 Partner Central 판매 API에서 게시한 이벤트에 액세스하고 관리하려면 적절한 IAM 권한이 필요합니다. EventBridge에 사용할 수 있는 작업, 리소스 및 조건 키에 대한 자세한 내용은 [Amazon EventBridge 사용 설명서의 Amazon EventBridge에서 IAM 정책 조건 사용을 참조하세요](#)

요. EventBridge 조건 키 중 하나는 특정 이벤트 유형에 대한 권한 범위를 지정하는 데 사용할 수 `events:detail-type` 있는 입니다.

다음 예제 정책은 제안된 이벤트에 대한 권한을 사용자 지정하고 범위를 지정하는 방법을 보여 줍니다. `AllowPutRuleForPartnercentralSellingEvents` 문은 규칙 생성을 허용하지만 `aws.partnercentral-selling` 소스의 이벤트에 대해서만 가능합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnercentralSellingEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "events:source": "aws.partnercentral-selling"
        }
      }
    }
  ]
}
```

이벤트를 모니터링하도록 Amazon EventBridge 구성

API 이벤트 판매를 모니터링하려면 캡처하려는 이벤트와 일치하는 EventBridge 규칙을 생성합니다. AWS Management Console 또는 AWS SDKs를 사용하여 규칙을 생성하고 관리할 수 있습니다. 다음 섹션에서는 두 방법을 모두 사용하여 규칙을 생성하는 방법을 설명합니다. 사용하는 방법에 관계없이 미국 동부(버지니아 북부) `us-east-1` 리전에서 규칙을 생성해야 합니다.

AWS Management Console 설정

를 사용하여 EventBridge 규칙을 설정하려면 [Amazon EventBridge 사용 설명서의 Amazon EventBridge에서 이벤트에 반응하는 규칙 생성](#)의 단계를 AWS Management Console 따릅니다. EventBridge 규칙을 생성할 때 이벤트 버스를 기본값으로 설정하고 미국 동부(버지니아 북부) `us-east-1` 리전에서 규칙을 생성해야 합니다.

다음은 이벤트 규칙의 예입니다.

```
{
  "source": ["aws.partnercentral-selling"],
  "detail": {
    "catalog": ["AWS"]
  }
}
```

AWS SDK 설정

AWS SDKs를 사용하여 EventBridge 규칙을 프로그래밍 방식으로 생성하고 관리할 수 있습니다. 자세한 내용은 Amazon EventBridge API 참조의 [PutRule](#)을 참조하세요.

다음 예제에서는 AWS SDK for Python(Boto3)을 사용합니다.

```
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'])
```

API 이벤트 판매에 대해 자세히 알아보기

다음 섹션에서는 판매 API 이벤트 유형, 이를 트리거하는 시나리오 및 이벤트 예제를 설명합니다.

이벤트 유형

다음은 이벤트 유형과 트리거입니다.

- [기회 생성됨](#): 새 기회가 생성될 때 트리거됩니다.
- [기회 업데이트](#)됨: 기회(기회 또는 해당AWS기회 요약)가 업데이트될 때 트리거됩니다.
- [참여 초대 생성됨](#): AWS추천 초대가 생성될 때 트리거됩니다.

- **참여 초대 수락**: 파트너가 AWS 참여 초대를 수락하여 기회에 대해와 협력하는 데 관심이 있음을 확인하면 트리거 AWS됩니다.
- **참여 초대 거부됨**: 파트너가 AWS 참여 초대를 거부할 때 트리거됩니다.
- **참여 초대 만료됨**: 파트너가 EngagementInvitation를 거부할 때 이 이벤트가 트리거됩니다. 발신 및 수신 파트너에게 초대가 만료되었음을 알립니다.
- **참여 구성원 추가됨**: 이 이벤트는 초대를 수락한 후 새 구성원이 참여에 참여할 때 트리거됩니다. 이 정책은 참여의 모든 현재 구성원에게 참여에 추가되는 새 구성원에 대해 알립니다.
- **참여 리소스 스냅샷 생성됨**: 이 이벤트는 참여와 연결된 리소스(예: 기회)의 스냅샷의 새 개정이 생성될 때 트리거됩니다. 연결된 리소스 스냅샷의 변경 사항에 대해 참여의 모든 현재 구성원에게 알립니다.
- **참여 생성됨**: 새 참여 생성 시 트리거됩니다.
- **참여 업데이트됨**: 참여 업데이트 시 트리거됩니다.

이벤트 시나리오

다음 표에서는 다양한 시나리오에서 이벤트가 작동하는 방식을 설명합니다. 이러한 시나리오에서는 다음과 같은 가정이 이루어집니다.

- AWS는 이러한 시나리오의 파트너이기도 합니다.
- ResourceSnapshotJob 파트너가를 올바르게 구성했습니다.
- 기회에 대해 업데이트된 필드는 리소스 스냅샷에 대한 기회 템플릿에도 있습니다.

파트너로서 사용자의 작업	수신된 이벤트	참가자 유형
기회를 생성합니다.	기회 생성됨	
StartEngagementFromOpportunityTask 를 사용하여 기회를 제출합니다.	기회 업데이트됨, 참여 생성됨, 참여 리소스 스냅샷 생성됨, 참여 구성원 추가됨, 참여 초대 생성됨	
AWS가 제출된 기회를 승인합니다.	참여 초대 수락됨, 참여 구성원 추가됨, 기회 업데이트됨, 참여 리소스 스냅샷 생성됨	

파트너로서 사용자의 작업	수신된 이벤트	참가자 유형
AWS는 제출된 기회를 거부합니다.	참여 초대 거부됨, 기회 업데이트됨, 참여 리소스 스냅샷 생성됨	
AWS Marketplace제안을 기회와 연결	기회 업데이트됨	
솔루션을 기회와 연결	기회 업데이트됨, 참여 리소스 스냅샷 생성됨	
기회를 업데이트합니다.	기회 업데이트됨, 참여 리소스 스냅샷 생성됨(ResourceSnapshotJob 이 설정되어 있고 필드 업데이트가 템플릿에 있는 경우 선택 사항)	
AWS가 기회를 업데이트합니다.	기회 업데이트됨, 참여 리소스 스냅샷 생성됨	
다른 파트너를 참여에 초대합니다.	참여 초대 생성됨	발신자
파트너가 참여 초대를 수락함	참여 초대 수락, 참여 구성원 추가	발신자
파트너가 참여 초대를 거부함	참여 초대 거부됨	발신자
파트너가 15일 동안 참여 초대를 처리하지 않습니다.	참여 초대 만료됨	발신자
다른 파트너로부터 참여 초대를 받습니다.	참여 초대 생성됨	수신기
를 통해 다른 파트너의 참여 초대를 수락합니다. StartEngagementByAcceptingInvitationTask	참여 초대 수락됨, 참여 구성원 추가됨, 기회 생성됨, 기회 업데이트됨, 참여 리소스 스냅샷 생성됨	수신기

파트너로서 사용자의 작업	수신된 이벤트	참가자 유형
다른 파트너의 참여 초대를 거부하는 경우	참여 초대 거부됨	수신기
15일 동안 다른 파트너의 참여 초대에 대해 조치를 취하지 않습니다.	참여 초대 만료됨	수신기
참여를 생성합니다.	참여 생성됨	발신자
참여를 업데이트합니다.	참여 업데이트됨	발신자, 수신자

이벤트 예제

다음 섹션에서는 이전 섹션의 앞부분에 나열된 이벤트의 예를 제공합니다.

주제

- [기회 생성됨](#)
- [기회 업데이트됨](#)
- [참여 초대 생성됨](#)
- [참여 초대 수락됨](#)
- [참여 초대 거부됨](#)
- [참여 초대 만료됨](#)
- [참여 구성원 추가됨](#)
- [참여 리소스 스냅샷 생성됨](#)
- [참여 생성됨](#)
- [참여 업데이트됨](#)

기회 생성됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 사용자에게 새 기회가 생성되었음을 알립니다.
- CRM 시스템 업데이트 또는 영업 팀에 새로운 기회 생성 알림과 같은 자동화된 워크플로를 트리거합니다.

다음 예제에서는 일반적인 기회 생성 이벤트를 보여줍니다.

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

기회 업데이트됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 기존 기회가 업데이트되었음을 사용자에게 알립니다.
- CRM 시스템 업데이트 또는 영업 팀에 알리는 등의 자동화된 워크플로를 트리거합니다.

다음 예제에서는 일반적인 기회 업데이트 이벤트를 보여줍니다.

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

```
}
}
```

참여 초대 생성됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 사용자에게 수신한 새 EngagementInvitation에 대해 알립니다.
- 자동 워크플로를 트리거하여 CRM 시스템 업데이트 또는 영업 팀에 알리는 등 초대를 수락하거나 거부합니다.

다음 예제에서는 일반적인 참여 초대 생성 이벤트를 보여줍니다.

```
{
  "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
    }
  }
}
```

참여 초대 수락됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 사용자에게 EngagementInvitation이 수락되었음을 알립니다.
- 참여의 새 파트너를 반영하도록 내부 레코드를 업데이트합니다.
- 자동화된 워크플로를 트리거하여 데이터를 동기화하거나 다른 팀원에게 새 참여 구성원에 대해 알립니다.

다음 예제에서는 일반적인 참여 초대 수락 이벤트를 보여줍니다.

```
{
  "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
    }
  }
}
```

참여 초대 거부됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 사용자에게 EngagementInvitation이 거부되었음을 알립니다.

- 거부된 초대를 반영하도록 내부 레코드를 업데이트합니다.
- 영업 팀에 거부 사실을 알리는 등 필요한 워크플로를 트리거합니다.

다음 예제에서는 일반적인 참여 초대 거부 이벤트를 보여줍니다.

```
{
  "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
    }
  }
}
```

참여 초대 만료됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 사용자에게 EngagementInvitation이 만료되어 더 이상 조치를 취할 수 없음을 알립니다.
- 만료된 초대를 반영하도록 내부 레코드를 업데이트합니다.
- 영업 팀에 만료된 초대에 대해 알리는 등 필요한 워크플로를 트리거합니다.

다음 예제에서는 일반적인 참여 초대 만료 이벤트를 보여줍니다.

```
{
  "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
    }
  }
}
```

참여 구성원 추가됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 참여 멤버십 변경 사항을 사용자에게 알립니다.
- 새 참여 구성원을 반영하도록 내부 레코드를 업데이트합니다.
- 팀원에게 변경 사항을 알리는 등 필요한 워크플로를 트리거합니다.

다음 예제에서는 일반적인 참여 구성원 추가 이벤트를 보여줍니다.

```
{
  "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"
  }
}
}

```

참여 리소스 스냅샷 생성됨

파트너는이 이벤트를 사용하여 참여와 관련된 리소스의 변경 사항을 추적하고 CRM 시스템 업데이트 또는 영업 팀에 알리는 등 자동화된 워크플로를 트리거합니다. 스냅샷 템플릿에 따라 업데이트 유형이 결정됩니다.

OpportunitySummaryView

- 참여의 모든 구성원에게 참여 업데이트 사실을 알립니다.

AwsOpportunitySummaryFullView

- AWS에서 특별히 수행한 업데이트를 참여 내의 기회로 추적합니다.
- 이 알림 및 스냅샷은 기회 소유자에게만 표시됩니다.
- OpportunitySummaryView 템플릿을 통해 사용할 수 없는 거래 크기 조정 업데이트가 포함되어 있습니다.

다음 예제는 일반적인 참여 리소스 스냅샷 생성 이벤트를 보여줍니다.

```

{
  "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"
  }
}
}

```

참여 생성됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 사용자에게 새 참여가 생성되었음을 알립니다.
- CRM 시스템 업데이트 또는 영업 팀에 새 참여 생성에 대한 알림과 같은 자동화된 워크플로를 트리거합니다.

다음 예제는 일반적인 참여 생성 이벤트를 보여줍니다.

```

{
  "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-567a1j5hxp351o"
  ],

```

```

    "detail": {
      "catalog": "AWS",
      "engagement": {
        "engagementArn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp351o",
        "engagementId": "eng-567a1j5hxp351o",
        "CreatedAt": "2023-04-18T18:20:15Z",
        "CreatedBy": "aws",
        "ContextTypes": ["Lead"]
      }
    }
  }
}

```

참여 업데이트됨

파트너는이 이벤트를 사용하여 다음을 수행합니다.

- 기존 계약이 업데이트되었음을 사용자에게 알립니다.
- CRM 시스템 업데이트 또는 영업 팀에 알리는 등의 자동화된 워크플로를 트리거합니다.

다음 예제에서는 일반적인 참여 업데이트 이벤트를 보여줍니다.

```

{
  "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-567a1j5hxp351o"
  ],
  "detail": {
    "catalog": "AWS",
    "engagement": {
      "engagementArn": "arn:aws:partnercentral:us-east-1::catalog/AWS/engagement/eng-567a1j5hxp351o",
      "engagementId": "eng-567a1j5hxp351o",
      "LastModifiedAt": "2023-04-18T18:20:15Z",
      "LastModifiedBy": "aws",

```

```

        "ContextTypes": ["Lead", "CustomerProject"]
    }
}
}

```

AWS Partner Central Benefits API에 대한 알림

Benefits Service는 BenefitApplications 및 BenefitAllocations 모두에 대한 EventBridge 이벤트를 제공합니다. 이러한 이벤트를 통해 고객은 초기 애플리케이션부터 할당 및 최종 이행까지 전체 리소스 수명 주기 동안 변경 사항에 대응하는 포괄적인 이벤트 기반 아키텍처를 구축할 수 있습니다.

주제

- [이벤트를 모니터링하기 위한 사전 조건 완료](#)
- [이벤트를 모니터링하도록 Amazon EventBridge 구성](#)
- [혜택 API 이벤트에 대해 자세히 알아보기](#)

이벤트를 모니터링하기 위한 사전 조건 완료

사용자는 혜택 API에서 게시한 이벤트에 액세스하고 관리하려면 적절한 IAM 권한이 필요합니다. EventBridge에 사용할 수 있는 작업, 리소스 및 조건 키에 대한 자세한 내용은 [Amazon EventBridge 사용 설명서의 Amazon EventBridge에서 IAM 정책 조건 사용을](#) 참조하세요. EventBridge 조건 키 중 하나는 특정 이벤트 유형에 대한 권한 범위를 지정하는 데 사용할 수 events:detail-type있는 입니다.

다음 예제 정책은 제안된 이벤트에 대한 권한을 사용자 지정하고 범위를 지정하는 방법을 보여줍니다. AllowPutRuleForPartnercentralBenefitsEvents 문은 규칙 생성을 허용하지만 aws.partnercentral-benefits 소스의 이벤트에 대해서만 가능합니다.

자세한 IAM 정책 예제는 EventBridge 권한에 대한 AWS 설명서를 참조하세요.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralBenefitsEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
      "Condition": {
        "StringEquals": {

```

```

        "events:source": "aws.partnercentral-benefits"
    }
}
]
}

```

이벤트를 모니터링하도록 Amazon EventBridge 구성

혜택 API 이벤트를 모니터링하려면 캡처하려는 이벤트와 일치하는 EventBridge 규칙을 생성합니다. AWS Management Console 또는 AWS SDKs를 사용하여 규칙을 생성하고 관리할 수 있습니다. 다음 섹션에서는 두 방법을 모두 사용하여 규칙을 생성하는 방법을 설명합니다. 사용하는 방법에 관계없이 미국 동부(버지니아 북부) us-east-1 리전에서 규칙을 생성해야 합니다.

AWS Management Console 설정

를 사용하여 EventBridge 규칙을 설정하려면 [Amazon EventBridge 사용 설명서의 Amazon EventBridge의 이벤트에 대응하는 규칙 생성](#)의 단계를 AWS Management Console 따릅니다. EventBridge 규칙을 생성할 때 이벤트 버스를 기본값으로 설정하고 미국 동부(버지니아 북부) us-east-1 리전에서 규칙을 생성해야 합니다.

다음은 이벤트 규칙의 예입니다.

```

{
  "source": ["aws.partnercentral-benefits"],
  "detail": {
    "catalog": ["AWS"]
  }
}

```

AWS SDK 설정

AWS SDKs를 사용하여 프로그래밍 방식으로 EventBridge 규칙을 생성하고 관리할 수 있습니다. 자세한 내용은 Amazon EventBridge API 참조의 [PutRule](#)을 참조하세요.

다음 예제에서는 AWS SDK for Python(Boto3)을 사용합니다.

```

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'])

```

혜택 API 이벤트에 대해 자세히 알아보기

다음 섹션에서는 이점 API 이벤트 유형, 이를 트리거하는 시나리오 및 이벤트 예제를 설명합니다.

이벤트 유형

다음은 이벤트 유형과 트리거입니다.

혜택 애플리케이션

- [혜택 애플리케이션이 생성된](#) 경우: 혜택 애플리케이션이 생성된 경우
- [혜택 애플리케이션 검토 중](#): 혜택 애플리케이션이 제출되거나 애플리케이션이 작업 필요에서 다시 검토 중으로 전환되면 이 이벤트가 트리거됩니다.
- [혜택 애플리케이션 작업 필요](#): 이 이벤트는 내부 검토자가 애플리케이션 상태를 업데이트하여 파트너가 검토를 계속하기 전에 추가 정보, 설명 또는 수정 사항을 제공해야 함을 나타낼 때 발생합니다.
- [혜택 애플리케이션 거부됨](#): 이 이벤트는 내부 검토자가 애플리케이션 상태를 업데이트하여 애플리케이션이 혜택 기준 또는 요구 사항을 충족하지 않고 거부되었음을 나타낼 때 발생합니다.
- [혜택 애플리케이션 승인됨](#): 이 이벤트는 내부 검토자가 애플리케이션 상태를 업데이트하여 애플리케이션이 혜택 할당 승인을 받았음을 나타낼 때 발생합니다.
- [혜택 애플리케이션 단계 변경됨](#): 이 이벤트는 내부 검토자가 비즈니스 검토, 검토, 재무AWS검토 등과 같은 검토 프로세스 중에 애플리케이션의 단계를 업데이트할 때 발생합니다.

혜택 할당

- [혜택 할당 활성화됨](#): 혜택 할당이 생성되거나 내부 서비스에 의해 다시 활성화됨으로 업데이트되는 경우.
- [혜택 할당 비활성화](#): 이 이벤트는 내부 검토자가 할당 상태를 업데이트하거나 혜택 할당이 자동으로 만료될 때 발생합니다.

- [혜택 할당 이행](#): 이 이벤트는 내부 검토자가 할당 상태를 업데이트하여 할당이 완전히 활용되었음을 나타낼 때 발생합니다.

이벤트 예제

다음 섹션에서는 이전 섹션의 앞부분에 나열된 이벤트의 예를 제공합니다.

주제

- [혜택 애플리케이션 생성됨](#)
- [검토 중인 혜택 애플리케이션](#)
- [혜택 애플리케이션 작업 필요](#)
- [혜택 애플리케이션 거부됨](#)
- [혜택 애플리케이션 승인됨](#)
- [혜택 애플리케이션 단계 변경됨](#)
- [혜택 할당 활성화됨](#)
- [혜택 할당 비활성화됨](#)
- [혜택 할당 이행됨](#)

혜택 애플리케이션 생성됨

트리거 컨텍스트

혜택 애플리케이션을 통해 새 혜택 애플리케이션이 성공적으로 생성되면 혜택 애플리케이션 생성 이벤트가 Amazon EventBridge에 게시됩니다. 이 이벤트는 CreateBenefitApplication API 호출이 성공적으로 완료될 때 발생하며, 이는 파트너가 특정 혜택에 대한 요청을 시작했음을 나타냅니다.

이 이벤트는 혜택 애플리케이션 수명 주기의 시작을 나타내며 새 애플리케이션이 계정에 제출되면 고객에게 즉시 알림을 제공합니다.

수신자

소유 계정은 혜택 할당에 대한 이벤트를 수신합니다.

예제

```
{
  "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"]
  }
}
}

```

검토 중인 혜택 애플리케이션

트리거 컨텍스트

혜택 애플리케이션이 혜택 서비스를 통해 IN_REVIEW 상태로 전환되면 혜택 애플리케이션 검토 중 이벤트가 Amazon EventBridge에 게시됩니다. 이 이벤트는 다음과 같은 경우에 발생합니다.

- 파트너가 SubmitBenefitApplication API 직접 호출을 통해 애플리케이션을 제출합니다.
- 파트너가 피드백을 해결한 후 애플리케이션이 ACTION_REQUIRED에서 IN_REVIEW로 다시 전환됩니다.
- 애플리케이션이 리콜된 다음 다시 제출됨

이 이벤트는 애플리케이션이 공식 검토 프로세스에 들어갔으며 AWS 내부 팀에서 평가 중임을 나타냅니다.

수신자

소유 계정은 혜택 할당에 대한 이벤트를 수신합니다.

예제

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

혜택 애플리케이션 작업 필요

트리거 컨텍스트

혜택 애플리케이션이 혜택 서비스를 통해 ACTION_REQUIRED 상태로 전환되면 혜택 애플리케이션 작업 필요 이벤트가 Amazon EventBridge에 게시됩니다. 이 이벤트는 내부 검토자가 UpdateBenefitApplicationInternal API를 통해 애플리케이션 상태를 업데이트하여 파트너가 추가 정보, 설명 또는 수정 사항을 제공해야 검토를 계속할 수 있음을 나타낼 때 발생합니다.

이 이벤트는 검토 프로세스에서 애플리케이션을 앞으로 이동하기 위해 파트너 개입이 필요한 애플리케이션 수명 주기의 중요 지점을 나타냅니다.

수신자

소유 계정은 혜택 할당에 대한 이벤트를 수신합니다.

예제

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

혜택 애플리케이션 거부됨

트리거 컨텍스트

혜택 애플리케이션이 혜택 서비스를 통해 거부됨 상태로 전환되면 혜택 애플리케이션 거부됨 이벤트가 Amazon EventBridge에 게시됩니다. 이 이벤트는 내부 검토자가 UpdateBenefitApplicationInternal API를 통해 애플리케이션 상태를 업데이트하여 애플리케이션이 혜택 기준 또는 요구 사항을 충족하지 않고 거부되었음을 나타낼 때 발생합니다.

이 이벤트는 애플리케이션이 공식적으로 거부되어 혜택 할당을 진행하지 않는 애플리케이션 수명 주기의 터미널 상태를 나타냅니다.

수신자

소유 계정은 혜택 할당에 대한 이벤트를 수신합니다.

예제

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

혜택 애플리케이션 승인됨

트리거 컨텍스트

혜택 애플리케이션이 혜택 서비스를 통해 APPROVED 상태로 전환되면 혜택 애플리케이션 승인 이벤트가 Amazon EventBridge에 게시됩니다. 이 이벤트는 내부 검토자가

UpdateBenefitApplicationInternal API를 통해 애플리케이션 상태를 업데이트하여 애플리케이션이 모든 혜택 기준 및 요구 사항을 충족하고 혜택 할당이 승인되었음을 나타낼 때 발생합니다.

이 이벤트는 애플리케이션 검토 프로세스를 성공적으로 완료했음을 나타내며 일반적으로 해당 혜택 할당 생성을 트리거합니다.

수신자

소유 계정은 혜택 할당에 대한 이벤트를 수신합니다.

예제

```
{
  "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": "l11i2844-i63f-85h8-e150-880099884456"
    },
    "benefit": {
      "id": "ben-xyz789uvw012",
      "name": "AWS Partner Credits",
      "program": ["AWS Partner Network"]
    }
  }
}
```

혜택 애플리케이션 단계 변경됨

트리거 컨텍스트

혜택 애플리케이션의 검토 단계가 혜택 서비스를 통해 업데이트되면 혜택 애플리케이션 단계 변경됨 이벤트가 Amazon EventBridge에 게시됩니다. 이 이벤트는 내부 검토자가 애플리케이션의 단계를 가능한 열거형 값(FINANCE_REVIEW, BUSINESS_REVIEW, TECH_REVIEW 및 AWS_REVIEW) 중 하나로 업데이트할 때 발생합니다.

상태 변경과 달리 단계 변경은 파트너 작업 없이 내부 검토 워크플로 진행 상황을 파악할 수 있는 읽기 전용 정보 업데이트입니다.

수신자

소유 계정은 혜택 할당에 대한 이벤트를 수신합니다.

예제

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

```
}
}
```

혜택 할당 활성화됨

트리거 컨텍스트

혜택 할당 활성화 이벤트는 혜택 서비스를 통해 혜택 할당이 ACTIVE 상태로 전환되면 Amazon EventBridge에 게시됩니다. 이 이벤트는 내부 검토자가 UpdateBenefitAllocationInternal API를 통해 할당 상태를 업데이트하여 이전에 비활성 상태였던 할당을 다시 활성화할 때 발생합니다.

이 이벤트는 일반적으로 일시적으로 비활성화된 할당(만료, 정책 변경 또는 관리상의 이유로)이 활성 상태로 복원되어 파트너 사용률이 다시 높아질 때 발생합니다.

수신자

파트너에 속한 AWS 계정이 메시지를 수신합니다. 즉, 리소스를 소유한 AWS 계정입니다.

샘플 이벤트

```
{
  "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"]
  }
}
}

```

혜택 할당 비활성화됨

트리거 컨텍스트

혜택 할당 비활성화 이벤트는 혜택 서비스를 통해 혜택 할당이 비활성 상태로 전환되면 Amazon EventBridge에 게시됩니다. 이 이벤트는 내부 검토자가 UpdateBenefitAllocationInternal API를 통해 할당 상태를 업데이트하거나 혜택 할당이 자동으로 만료될 때 발생합니다.

이 이벤트는 일반적으로 할당이 일시 중지될 때(만료, 정책 변경, 규정 준수 문제 또는 관리상의 이유로) 다시 활성화될 때까지 파트너 사용률이 혜택을 사용할 수 없게 됩니다.

수신자

파트너에 속한 AWS 계정이 메시지를 수신합니다. 즉, 리소스를 소유한 AWS 계정입니다.

샘플 이벤트

```

{
  "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"]
    }
  }
}

```

혜택 할당 이행됨

트리거 컨텍스트

혜택 할당 이행 이벤트는 혜택 서비스를 통해 혜택 할당이 FULFILLED 상태로 전환되면 Amazon EventBridge에 게시됩니다. 이 이벤트는 내부 검토자가 UpdateBenefitAllocationInternal API를 통해 할당 상태를 업데이트하여 할당이 완전히 활용되었거나 혜택 조건이 충족되었음을 나타낼 때 발생합니다.

이 이벤트는 혜택 수명 주기의 완료를 나타내며, 파트너가 할당된 혜택을 성공적으로 활용했거나 할당이 자연스러운 결론에 도달했음을 나타냅니다.

수신자

파트너에 속한 AWS 계정이 메시지를 수신합니다. 즉, 리소스를 소유한 AWS 계정입니다.

샘플 이벤트

```

{
  "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"]
  }
}
}
}

```

AWS파트너 중앙 채널 API에 대한 알림

채널 API의 이벤트는 상태 변경 또는 채널 관련 활동의 세부 정보에 대한 실시간 알림을 제공합니다. 이러한 이벤트는 시스템을 AWS Partner Central과 동기화하고 적시에 응답 및 업데이트를 보장하는 데 도움이 됩니다.

사전 조건

AWS Partner Central Channel API에서 이벤트에 액세스하고 관리하려면 적절한 IAM 권한이 필요합니다. EventBridge에 사용할 수 있는 작업, 리소스 및 조건 키에 대한 자세한 내용은 [Amazon EventBridge 사용 설명서의 Amazon EventBridge에서 IAM 정책 조건 사용](#)을 참조하세요. EventBridge 조건 키 중 하나는 특정 이벤트 유형에 대한 권한 범위를 지정하는 데 사용할 수 있는 `events:detail-type`입니다.

다음 예제 정책은 이벤트에 대한 권한을 사용자 지정하고 범위를 지정하는 방법을 보여줍니다. `AllowPutRuleForPartnercentralChannelEvents` 문은 규칙 생성을 허용하지만 `aws.partnercentral-channel` 소스의 이벤트에 대해서만 가능합니다.

자세한 IAM 정책 예제는 EventBridge 권한에 대한 AWS 설명서를 참조하세요.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutRuleForPartnerCentralChannelEvents",
      "Effect": "Allow",
      "Action": "events:PutRule",
      "Resource": "*",
    }
  ]
}

```

```

    "Condition": {
      "StringEquals": {
        "events:source": "aws.partnercentral-channel"
      }
    }
  }
]
}

```

이벤트를 모니터링하도록 Amazon EventBridge 구성

채널 API 이벤트를 모니터링하려면 캡처하려는 이벤트와 일치하는 EventBridge 규칙을 생성합니다. AWS Management Console 또는 AWS SDKs를 사용하여 규칙을 생성하고 관리할 수 있습니다. 다음 섹션에서는 두 방법을 모두 사용하여 규칙을 생성하는 방법을 설명합니다. 사용하는 방법에 관계없이 미국 동부(버지니아 북부) us-east-1 리전에서 규칙을 생성해야 합니다.

AWS Management Console 설정

를 사용하여 EventBridge 규칙을 설정하려면 [Amazon EventBridge 사용 설명서의 Amazon EventBridge에서 이벤트에 반응하는 규칙 생성](#)의 단계를 AWS Management Console 따르세요. EventBridge 규칙을 생성할 때 이벤트 버스를 기본값으로 설정하고 미국 동부(버지니아 북부) us-east-1 리전에서 규칙을 생성해야 합니다.

다음은 이벤트 규칙의 예입니다.

```

{
  "source": ["aws.partnercentral-channel"],
  "detail": {
    "catalog": ["AWS"]
  }
}

```

AWS SDK 설정

AWS SDKs를 사용하여 프로그래밍 방식으로 EventBridge 규칙을 생성하고 관리할 수 있습니다. 자세한 내용은 Amazon EventBridge API 참조의 [PutRule](#)을 참조하세요.

다음 예제에서는 AWS SDK for Python(Boto3)을 사용합니다.

```
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'])

```

채널 API 이벤트에 대해 자세히 알아보기

다음 섹션에서는 채널 API 이벤트 유형, 이를 트리거하는 시나리오 및 이벤트 예제를 설명합니다.

이벤트 유형

다음은 이벤트 유형과 트리거입니다.

- [채널 핸드셰이크 생성됨](#): 새 채널 핸드셰이크가 생성될 때 트리거됩니다.
- [채널 핸드셰이크 수락됨](#): 새 채널 핸드셰이크가 수락되면 트리거됩니다.
- [채널 핸드셰이크 거부됨](#): 새 채널 핸드셰이크가 거부될 때 트리거됩니다.

채널 핸드셰이크 생성됨

다음 예제에서는 다양한 핸드셰이크 유형에 대한 일반적인 채널 핸드셰이크 생성 이벤트를 보여줍니다.

서비스 기간 핸드셰이크 시작:

```

{
    "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"
        }
      }
    }
  }
}

```

서비스 기간 핸드셰이크 취소:

```

{
  "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"
      }
    }
  }
}

```

프로그램 관리 계정 핸드셰이크:

```

{
  "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"
      }
    }
  }
}

```

채널 핸드셰이크 수락됨

다음 예제에서는 다양한 핸드셰이크 유형에 대해 허용되는 일반적인 채널 핸드셰이크 이벤트를 보여줍니다.

서비스 기간 핸드셰이크 시작:

```

{
  "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"
      }
    }
  }
}

```

서비스 기간 핸드셰이크 취소:

```

{
  "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"
      }
    }
  }
}

```

프로그램 관리 계정 핸드셰이크:

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

채널 핸드셰이크 거부됨

다음 예제에서는 다양한 핸드셰이크 유형에 대한 일반적인 채널 핸드셰이크 거부 이벤트를 보여줍니다.

서비스 기간 핸드셰이크 시작:

```
{
  "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"
            }
        }
    }
}

```

서비스 기간 핸드셰이크 취소:

```

{
    "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"
      }
    }
  }
}

```

프로그램 관리 계정 핸드셰이크:

```

{
  "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"
        }
    }
}
}
}

```

샌드박스에서 테스트

파트너는 샌드박스를 사용하여 안전하고 격리된 환경에서 AWS Partner Central API 상호 작용을 테스트하고 검증하여 솔루션을 프로덕션 환경으로 승격하기 전에 원활한 작동을 보장할 수 있습니다. AWS Partner Central API 사용자에게 프로덕션 환경과 유사한 응답을 반환하는 동적 샌드박스를 AWS 제공합니다. AWS는 샌드박스 환경에 대한 사용자 인터페이스를 제공하지 않습니다. 따라서 파트너는 프로그래밍 방식의 응답에 의존하여 솔루션을 테스트해야 합니다.

샌드박스 환경에 대한 액세스

파트너는 Partner Central 계정에 연결하는 즉시 테스트 환경에 액세스할 수 있습니다. 자세한 내용은 [AWS 계정과 AWS Partner Central 계정 연결을 참조하세요](#). 각 요청에는 데이터 환경을 결정하는 catalog 파라미터가 포함됩니다. catalog를 로 설정하면 프로덕션 데이터를 참조하고, AWS로 설정하면 샌드박스 데이터를 Sandbox 참조합니다.

샌드박스 환경에 대한 중요한 세부 정보

1. 데이터 새로 고침: 연 1회, 샌드박스 환경(일반적으로 연초)의 데이터를 AWS 새로 고칩니다. 이 새로 고침 후 테스트 환경에서 일부 데이터가 손실될 수 있습니다.
2. 테스트 범위: 샌드박스 환경은 일반적으로 확장성 또는 성능 테스트가 아닌 기능 테스트에 사용됩니다.

주제

- [AWS Partner Central Account API용 샌드박스에서 테스트](#)
- [AWS Partner Central Selling API용 샌드박스에서 테스트](#)
- [AWS Partner Central Benefits API용 샌드박스에서 테스트](#)
- [AWS Partner Central Channel API용 샌드박스에서 테스트](#)

- [AWS파트너 중앙 수익 측정 API의 샌드박스에서 테스트](#)

AWS Partner Central Account API용 샌드박스에서 테스트

샌드박스 사용 방법

샌드박스를 사용하려면 다음 단계를 완료합니다.

1. IAM 역할 생성:

AWS Partner Central 계정과AWS 계정연결된에서 IAM 역할을 생성합니다.

2. 정책 할당:

IAM 역할에 다음 정책을 할당합니다. 자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

```
{
  "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. IAM 역할 자격 증명 사용:

솔루션에서이 IAM 역할의 자격 증명(보안 키 및 액세스 키)을 사용하여 API 작업을 수행합니다.

AWS작업 테스트

테스트 단계에서 AWS는 작업을 시뮬레이션해야 하는 경우가 많습니다. 이 시뮬레이션을 통해 파트너는 AWS 서비스와의 통합에 대한 전체 end-to-end 흐름을 철저하게 테스트할 수 있습니다. 각 요청에는 데이터 환경을 결정하는 카탈로그 파라미터가 포함됩니다.

파트너 생성 시뮬레이션

파트너 생성을 시뮬레이션하려면 CreatePartner 작업의 페이로드에 `catalog`: "Sandbox"에 포함합니다.

예제:

```
{
  "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"
}
```

참고: 샌드박스에서 이메일 확인이 비활성화되었습니다. 6자리 EmailVerificationCode를 테스트에 사용할 수 있습니다. 또한 요청에 "태그"를 사용하지 마십시오.

추가 테스트 참고 사항

1. 다른 작업을 테스트하려면 페이로드 `catalog`: "Sandbox"에서 설정합니다.
2. 샌드박스에서 파트너 계정 연결을 테스트하려면 샌드박스 카탈로그에서 파트너를 생성한 후 StartProfileUpdateTask 작업을 실행해야 합니다.
3. 프로덕션으로 이동하려면 카탈로그 값에서 Sandbox로 변경합니다 AWS.

샌드박스 환경에서 이벤트 테스트

파트너는 샌드박스 환경의 이벤트를 사용하여 이벤트 기반 구현을 테스트할 수 있습니다. 이벤트 세부 정보에서 지정하여 샌드박스 이벤트를 수신하도록 규칙을 AWS 계정으로 사용하여 동일한 catalog:

Sandbox에 EventBridge를 설정합니다. 자세한 내용은 [AWS파트너 중앙 계정 API에 대한 알림](#) 단원을 참조하십시오.

이벤트 규칙의 예:

```
{
  "source": ["aws.partnercentral-account"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

를 catalog로 지정하는 이벤트 규칙Sandbox은 샌드박스 환경에서 수행하는 작업에 의해 생성된 샌드박스에서 오는 이벤트와만 일치합니다.

AWS Partner Central Selling API용 샌드박스에서 테스트

샌드박스 사용 방법

1. IAM 역할 생성:

AWS Partner Central 계정과AWS 계정연결된에서 IAM 역할을 생성합니다.

2. 정책 할당:

IAM 역할에 다음 정책을 할당합니다. 자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하십시오.

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. IAM 역할 자격 증명 사용:

솔루션에서 IAM 역할의 자격 증명(보안 키 및 액세스 키)을 사용하여 API 작업을 수행합니다.

AWS작업 테스트

테스트 단계에서 AWS는 작업을 시뮬레이션해야 하는 경우가 많습니다. 이 시뮬레이션을 통해 파트너는 AWS 서비스와의 통합에 대한 전체 end-to-end 흐름을 철저하게 테스트할 수 있습니다.

AWS시작된 기회 생성 시뮬레이션

AWS시작된 기회의 생성을 시뮬레이션하려면 CreateOpportunity 작업의 페이로드에 "Catalog": "Sandbox" 및 "Origin": "AWS Referral".

예제:

```
{
  "Catalog": "Sandbox",
  "Origin": "AWS Referral",
  "OpportunityIdentifier": "0123456",
  "Title": "Test Opportunity",
  ...
}
```

파트너 생성 기회에 대한 업데이트 시뮬레이션

파트너 생성 기회에 대한 업데이트 또는 기타 작업을 시뮬레이션하려면 페이로드 "Action Required"에서 "Catalog": "Sandbox" 및 Lifecycle.ReviewStatus: "Approved" 또는와 함께 UpdateOpportunity 작업을 사용합니다.

GetOpportunity 작업에서 페이로드를 가져와 업데이트를 수행하는 경우 ID를 로 변경Identifier하고 다음 필드를 제거해야 합니다.

- CreatedDate
- OpportunityTeam
- RelatedEntityIdentifiers

예제:

```
{
  "Catalog": "Sandbox",
  "Identifier": "0123456",
  "Title": "Updated Test Opportunity",
  "Lifecycle": {
    "ReviewStatus": "Approved"
  }
  ...
}
```

샌드박스 환경에서 이벤트 테스트

파트너는 샌드박스 환경의 기회 이벤트를 사용하여 이벤트 기반 구현을 테스트할 수 있습니다. 이벤트 세부 정보에를 지정하여 샌드박스 이벤트를 수신하도록 규칙을AWS 계정으로 사용하여 동일한 catalog:sandbox에 EventBridge를 설정합니다. 자세한 내용은 [AWS Partner Central Selling API에 대한 알림 단원을 참조하십시오](#).

이벤트 규칙의 예:

```
{
  "source": ["aws.partnercentral-selling"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

를 catalog로 지정하는 이벤트 규칙Sandbox은 샌드박스 환경에서 수행하는 작업에 의해 생성된 샌드박스에서 오는 이벤트와만 일치합니다.

추가 테스트 참고 사항

1. 테스트 AssociateOpportunity 작업:

- AssociateOpportunity 작업을 테스트"S-1234567"하려면 기본 솔루션을 사용합니다.
- Marketplace 제안을 테스트하려면 계정의 실제 제안 ID를 사용합니다.

2. 프로덕션으로 이동하려면 catalog 값에서 Sandbox로 변경합니다AWS.

도움말 가져오기

CRM을와 통합하는 데 문제가AWS발생하거나 여기에서 다루지 않은 특정 시나리오를 테스트해야 하는 경우 다음 단계를 통해 사례를 제기하여 지원팀에 문의하십시오.

1. [AWS파트너 네트워크 자격 증명으로 Partner Central](#)에 로그인합니다.AWS
2. [Partner Central 지원 센터에서](#) 새 사례를 로깅Open New Case하도록 선택합니다. 다음과 같이 필드를 작성합니다.
 - a. 지원 사례 유형: AWS Partner Central.
 - b. 관련 질문: Partner Central Tools or ACE leads and opportunities.
 - c. 자세히 알아보기: 가장 적합한 CRM 통합 사례 유형을 선택합니다.
 - d. 제목: 요청에 대한 간략한 설명을 포함합니다.
 - e. 설명: 문제, 질문, 오류 및 문제 해결 단계에 대한 자세한 설명을 제공합니다.
 - f. 첨부 파일: 해당하는 경우 로그와 스크린샷을 포함합니다.

AWS Partner Central Benefits API용 샌드박스에서 테스트

샌드박스 사용 방법

샌드박스를 사용하려면 다음 단계를 완료합니다.

1. IAM 역할 생성:

AWS Partner Central 계정과AWS 계정연결된에서 IAM 역할을 생성합니다.

2. 정책 할당:

IAM 역할에 다음 정책을 할당합니다. 자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

```
{
  "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. IAM 역할 자격 증명 사용:

솔루션에서이 IAM 역할의 자격 증명(보안 키 및 액세스 키)을 사용하여 API 작업을 수행합니다.

AWS작업 테스트

테스트 단계에서AWS는 작업을 시뮬레이션해야 하는 경우가 많습니다. 이 시뮬레이션을 통해 파트너는AWS서비스와의 통합에 대한 전체 end-to-end 흐름을 철저하게 테스트할 수 있습니다. 각 요청에는 데이터 환경을 결정하는 카탈로그 파라미터가 포함됩니다.

혜택 애플리케이션 생성 시뮬레이션

혜택 애플리케이션 생성을 시뮬레이션하려면 CreateBenefitApplication 작업의 페이로드에를 페이로드"catalog": "Sandbox"에 포함합니다.

예제:

```

{
  "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"
  }
]
}

```

추가 테스트 참고 사항

1. 다른 작업을 테스트하려면 페이로드 "catalog": "Sandbox"에서 설정합니다.
2. 샌드박스 카탈로그와 AWS 카탈로그IDs는 다릅니다. 를 사용하여 GetBenefit API를 호출 "catalog": "Sandbox"하여 샌드박스에서 혜택 식별자를 가져옵니다.
3. 샌드박스 카탈로그와 AWS 카탈로그의 혜택 할당은 다릅니다.
4. 프로덕션으로 이동하려면 카탈로그 값에서 Sandbox로 변경합니다AWS.

샌드박스 환경에서 이벤트 테스트

파트너는 샌드박스 환경의 이벤트를 사용하여 이벤트 기반 구현을 테스트할 수 있습니다. 이벤트 세부 정보에서 지정하여 샌드박스 이벤트를 수신하도록 규칙을AWS 계정으로 사용하여 동일한 catalog: Sandbox에 EventBridge를 설정합니다. 자세한 내용은 [AWS Partner Central Benefits API에 대한 알림 단원을 참조하십시오](#).

이벤트 규칙의 예:

```

{
  "source": ["aws.partnercentral-benefits"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}

```

```
}
```

를 catalog로 지정하는 이벤트 규칙Sandbox은 샌드박스 환경에서 수행하는 작업에 의해 생성된 샌드박스에서 오는 이벤트와만 일치합니다.

AWS Partner Central Channel API용 샌드박스에서 테스트

샌드박스의 계정은 채널 혜택을 받을 수 없으며 모든 프로그램 요구 사항에 대해 검증되지 않습니다.

샌드박스 사용 방법

샌드박스를 사용하려면 다음 단계를 완료합니다.

1. IAM 역할 생성:

AWS Partner Central 계정과AWS 계정연결된에서 IAM 역할을 생성합니다.

2. 정책 할당:

IAM 역할에 다음 정책을 할당합니다. 자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

```
{
  "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. IAM 역할 자격 증명 사용:

솔루션에서이 IAM 역할의 자격 증명(보안 키 및 액세스 키)을 사용하여 API 작업을 수행합니다.

AWS작업 테스트

테스트 단계에서AWS는 작업을 시뮬레이션해야 하는 경우가 많습니다. 이 시뮬레이션을 통해 파트너는AWS서비스와의 통합에 대한 전체 end-to-end 흐름을 철저하게 테스트할 수 있습니다. 각 요청에는 데이터 환경을 결정하는 카탈로그 파라미터가 포함됩니다.

프로그램 관리 계정 생성 시뮬레이션

프로그램 관리 계정 생성을 시뮬레이션하려면 `CreateProgramManagementAccount` 작업의 페이로드에 `"catalog": "Sandbox"`에 포함합니다.

예제:

```
{
  "catalog": "Sandbox",
  "accountId": "123456789012",
  "displayName": "Test PMA Name",
  "clientToken": "123abc456def789ghi",
  "program": "SOLUTION_PROVIDER",
  "tags": [
    {
      "key": "testkey",
      "value": "testvalue"
    }
  ]
}
```

관계에 대한 업데이트 시뮬레이션

관계에 대한 업데이트를 시뮬레이션하려면 페이로드에서 `"catalog": "Sandbox"`와 함께 `UpdateRelationship` 작업을 사용합니다.

예제:

```
{
  "catalog": "Sandbox",
  "displayName": "Updated Test PMA",
  "identifier": "rs-abc123def456g",
  "programManagementAccountIdentifier": "pma-123abc456def7"
}
```

추가 테스트 참고 사항

1. 다른 작업을 테스트하려면 페이로드 `"catalog": "Sandbox"`에서를 설정합니다.
2. 프로덕션으로 이동하려면 카탈로그 값에서 `Sandbox`로 변경합니다AWS.

샌드박스 환경에서 이벤트 테스트

파트너는 샌드박스 환경의 이벤트를 사용하여 이벤트 기반 구현을 테스트할 수 있습니다. 이벤트 세부 정보에서를 지정하여 샌드박스 이벤트를 수신하도록 규칙을AWS 계정으로 사용하여 동일한 catalog:Sandbox에 EventBridge를 설정합니다. 자세한 내용은 [AWS파트너 중앙 채널 API에 대한 알림](#) 단원을 참조하십시오.

이벤트 규칙의 예:

```
{
  "source": ["aws.partnercentral-channel"],
  "detail": {
    "catalog": ["Sandbox"]
  }
}
```

를 catalog로 지정하는 이벤트 규칙Sandbox은 샌드박스 환경에서 수행하는 작업에 의해 생성된 샌드박스에서 오는 이벤트와만 일치합니다.

도움말 가져오기

테스트에 문제가 발생하거나 여기에서 다루지 않은 특정 시나리오를 테스트해야 하는 경우 다음 단계를 통해 사례를 제기하여 지원팀에 문의하십시오.

1. [AWS파트너 네트워크 자격 증명으로 Partner Central](#)에 로그인합니다.AWS
2. [Partner Central 지원 센터에서](#) 새 사례를 로깅Open New Case하도록 선택합니다. 다음과 같이 필드를 작성합니다.
 - a. 지원 사례 유형: Channel Program
 - b. 다음에 대한 질문: General Program Inquiry
 - c. 구체적으로 알아보기: 가장 적합한 채널 사례 유형을 선택합니다.
 - d. 제목: 요청에 대한 간략한 설명을 포함합니다.
 - e. 설명: 문제, 질문, 오류 및 문제 해결 단계에 대한 자세한 설명을 제공합니다.
 - f. 첨부 파일: 해당하는 경우 로그와 스크린샷을 포함합니다.

AWS파트너 중앙 수익 측정 API의 샌드박스에서 테스트

샌드박스 사용 방법

샌드박스를 사용하려면 다음 단계를 완료합니다.

1. IAM 역할 생성:

AWS Partner Central 계정과AWS 계정연결된에서 IAM 역할을 생성합니다.

2. 정책 할당:

IAM 역할에 다음 정책을 할당합니다. 자세한 내용은 [IAM 자격 증명 권한 추가 및 제거](#)를 참조하세요.

```
{
  "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. IAM 역할 자격 증명 사용:

솔루션에서이 IAM 역할의 자격 증명(보안 키 및 액세스 키)을 사용하여 API 작업을 수행합니다.

AWS작업 테스트

테스트 단계에서AWS는 작업을 시뮬레이션해야 하는 경우가 많습니다. 이 시뮬레이션을 통해 파트너는AWS서비스와의 통합에 대한 전체 end-to-end 흐름을 철저하게 테스트할 수 있습니다. 각 요청에는 데이터 환경을 결정하는 카탈로그 파라미터가 포함됩니다.

수익 속성 ID 생성 시뮬레이션

수익 속성 ID 생성을 시뮬레이션하려면 CreateRevenueAttribution 작업의 페이로드에를 페이로드 "Catalog": "Sandbox"에 포함합니다.

예제:

```

{
  "Catalog": "Sandbox",
  "ClientToken": "String",
  "Name": "String",
  "Description": "String",
  "TenancyModel": "MULTI_TENANT | SINGLE_TENANT",
  "ProductIdentifier": "String",
  "Tags": [

```

```
{
  "Key": "String",
  "Value": "String"
}
]
```

추가 테스트 참고 사항

1. 다른 작업을 테스트하려면 페이로드 "Catalog": "Sandbox"에서 설정합니다.
2. 프로덕션으로 이동하려면 카탈로그 값에서 Sandbox로 변경합니다AWS.

AWS SDKs를 사용하는 Partner Central의 코드 예제

다음 코드 예제에서는 Partner Central을 AWS 소프트웨어 개발 키트(SDK)와 함께 사용하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접 호출하는 방법을 보여주며, 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#) [AWS SDK에서 AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

코드 예시

- [AWS SDKs 사용하는 Partner Central의 기본 예제](#)
 - [AWS SDKs를 사용한 Partner Central 작업](#)
 - [AWS SDK와 AssignOpportunity 함께 사용](#)
 - [AWS SDK와 AssociateOpportunity 함께 사용](#)
 - [AWS SDK와 CreateOpportunity 함께 사용](#)
 - [AWS SDK와 DisassociateOpportunity 함께 사용](#)
 - [AWS SDK와 GetAwsOpportunitySummary 함께 사용](#)
 - [AWS SDK와 GetEngagementInvitation 함께 사용](#)
 - [AWS SDK와 GetOpportunity 함께 사용](#)
 - [AWS SDK와 ListEngagementInvitations 함께 사용](#)
 - [AWS SDK와 ListOpportunities 함께 사용](#)
 - [AWS SDK와 ListSolutions 함께 사용](#)
 - [AWS SDK와 RejectEngagementInvitation 함께 사용](#)
 - [AWS SDK와 StartEngagementByAcceptingInvitationTask 함께 사용](#)
 - [AWS SDK와 StartEngagementFromOpportunityTask 함께 사용](#)
 - [AWS SDK와 UpdateOpportunity 함께 사용](#)
 - [AWS SDKs를 사용한 Partner Central 시나리오](#)
 - [기회에 연결된 엔터티 업데이트](#)

AWS SDKs 사용하는 Partner Central의 기본 예제

다음 코드 예제에서는 SDKs와 함께 AWS Partner Central의 기본 사항을 사용하는 방법을 보여줍니다.

예제

- [AWS SDKs를 사용한 Partner Central 작업](#)
 - [AWS SDK와 AssignOpportunity 함께 사용](#)
 - [AWS SDK와 AssociateOpportunity 함께 사용](#)
 - [AWS SDK와 CreateOpportunity 함께 사용](#)
 - [AWS SDK와 DisassociateOpportunity 함께 사용](#)
 - [AWS SDK와 GetAwsOpportunitySummary 함께 사용](#)
 - [AWS SDK와 GetEngagementInvitation 함께 사용](#)
 - [AWS SDK와 GetOpportunity 함께 사용](#)
 - [AWS SDK와 ListEngagementInvitations 함께 사용](#)
 - [AWS SDK와 ListOpportunities 함께 사용](#)
 - [AWS SDK와 ListSolutions 함께 사용](#)
 - [AWS SDK와 RejectEngagementInvitation 함께 사용](#)
 - [AWS SDK와 StartEngagementByAcceptingInvitationTask 함께 사용](#)
 - [AWS SDK와 StartEngagementFromOpportunityTask 함께 사용](#)
 - [AWS SDK와 UpdateOpportunity 함께 사용](#)

AWS SDKs를 사용한 Partner Central 작업

다음 코드 예제에서는 AWS SDKs를 사용하여 개별 Partner Central 작업을 수행하는 방법을 보여줍니다. 각 예제에는 GitHub에 대한 링크가 포함되어 있습니다. 여기에서 코드 설정 및 실행에 대한 지침을 찾을 수 있습니다.

Partner Central API를 직접적으로 호출하는 이들 발체문은 컨텍스트에서 실행되어야 하는 더 큰 프로그램에서 발체한 코드입니다. [AWS SDKs를 사용한 Partner Central 시나리오](#)에서 컨텍스트에 맞는 작업을 볼 수 있습니다.

다음 예제에는 가장 일반적으로 사용되는 작업만 포함되어 있습니다. 전체 목록은 [AWS Partner Central API 참조](#)에서 확인하세요.

예제

- [AWS SDK와 AssignOpportunity 함께 사용](#)
- [AWS SDK와 AssociateOpportunity 함께 사용](#)
- [AWS SDK와 CreateOpportunity 함께 사용](#)
- [AWS SDK와 DisassociateOpportunity 함께 사용](#)
- [AWS SDK와 GetAwsOpportunitySummary 함께 사용](#)
- [AWS SDK와 GetEngagementInvitation 함께 사용](#)
- [AWS SDK와 GetOpportunity 함께 사용](#)
- [AWS SDK와 ListEngagementInvitations 함께 사용](#)
- [AWS SDK와 ListOpportunities 함께 사용](#)
- [AWS SDK와 ListSolutions 함께 사용](#)
- [AWS SDK와 RejectEngagementInvitation 함께 사용](#)
- [AWS SDK와 StartEngagementByAcceptingInvitationTask 함께 사용](#)
- [AWS SDK와 StartEngagementFromOpportunityTask 함께 사용](#)
- [AWS SDK와 UpdateOpportunity 함께 사용](#)

AWS SDK와 **AssignOpportunity** 함께 사용

다음 코드 예시는 AssignOpportunity의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

기존 기회를 다른 사용자에게 재할당합니다.

```
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;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [AssignOpportunity](#)를 참조하세요.

Python

SDK for Python(Boto3)

기존 기회를 다른 사용자에게 재할당합니다.

```

#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [AssignOpportunity](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **AssociateOpportunity** 함께 사용

다음 코드 예시는 AssociateOpportunity의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서 컨텍스트 내 이 작업을 확인할 수 있습니다.

- [기회에 연결된 엔터티 업데이트](#)

Java

SDK for Java 2.x

기회와 다양한 관련 엔터티 간에 공식 연결을 생성합니다.

```
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_TO_USE)
                .opportunityIdentifier(opportunityId)
                .relatedEntityType(entityType)
                .relatedEntityIdentifier(entityIdIdentifier)
                .build();

        AssociateOpportunityResponse response =
            client.associateOpportunity(associateOpportunityRequest);

        return response;
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [AssociateOpportunity](#)를 참조하세요.

Python

SDK for Python(Boto3)

기회와 다양한 관련 엔터티 간에 공식 연결을 생성합니다.

```
#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def associate_opportunity(entity_type, entity_identifier, opportunityIdentifier):
    associate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : entity_type,
        "RelatedEntityIdentifier" : entity_identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.associate_opportunity(**associate_opportunity_request)
        return response

    except ClientError as err:
        # Catch all client exceptions
```

```

        print(err.response)

def usage_demo():
    #entity_type = Solutions | AWSProducts | AWSMarketplaceOffers
    entity_type = "Solutions"
    entity_identifier = "S-0059717"
    opportunityIdentifier = "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_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [AssociateOpportunity](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **CreateOpportunity** 함께 사용

다음 코드 예시는 CreateOpportunity의 사용 방법을 보여 줍니다.

.NET

SDK for .NET

기회를 생성합니다.

```

// 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.");
    }
}
}
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [CreateOpportunity](#)를 참조하세요.

Java

SDK for Java 2.x

기회를 생성합니다.

```

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(expectedCustomerSpends)
                .salesActivities(root.project.salesActivities)
                .competitorName(root.project.competitorName)
                .otherSolutionDescription(root.project.otherSolutionDescription)
                .build();
        }

        SoftwareRevenue softwareRevenue = null;
        if ( root.softwareRevenue != null) {
            MonetaryValue monetaryValue = null;
            if ( root.softwareRevenue.value != null) {
                monetaryValue = MonetaryValue.builder()
                    .amount(root.softwareRevenue.value.amount)
                    .currencyCode(root.softwareRevenue.value.currencyCode)
                    .build();
            }
            softwareRevenue = SoftwareRevenue.builder()
                .deliveryModel(root.softwareRevenue.deliveryModel)
                .effectiveDate(root.softwareRevenue.effectiveDate)
                .expirationDate(root.softwareRevenue.expirationDate)
                .value(monetaryValue)
                .build();
        }

        // Building the Actual CreateOpportunity Request
        CreateOpportunityRequest createOpportunityRequest =
        CreateOpportunityRequest.builder()
            .catalog(CATALOG_TO_USE)
            .clientToken(root.clientToken)
            .primaryNeedsFromAwsWithStrings(root.primaryNeedsFromAws)
            .opportunityType(root.opportunityType)
            .lifeCycle(lifeCycle)
            .marketing(marketing)
            .nationalSecurity(root.nationalSecurity)
            .origin(root.origin)
            .customer(customer)
            .project(project)
            .partnerOpportunityIdentifier(root.partnerOpportunityIdentifier)

```

```

        .opportunityTeam(opportunityTeamContact)
        .softwareRevenue(softwareRevenue)
        .build();

    CreateOpportunityResponse response =
    client.createOpportunity(createOpportunityRequest);
    System.out.println("Successfully created: " + response);

    return response;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [CreateOpportunity](#)를 참조하세요.

Python

SDK for Python(Boto3)

기회를 생성합니다.

```

#!/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_T0_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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [CreateOpportunity](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **DisassociateOpportunity** 함께 사용

다음 코드 예시는 DisassociateOpportunity의 사용 방법을 보여 줍니다.

작업 예제는 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 다음 코드 예제에서 컨텍스트 내 이 작업을 확인할 수 있습니다.

- [기회에 연결된 엔터티 업데이트](#)

Java

SDK for Java 2.x

기회 및 관련 엔터티 간의 기존 연결을 제거합니다.

```
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 entityIdentifier = "S-00000000";

        DisassociateOpportunityResponse response = getResponse(opportunityId,
            entityType, entityIdentifier );

        ReferenceCodesUtils.formatOutput(response);
    }

    static DisassociateOpportunityResponse getResponse(String opportunityId, String
        entityType, String entityIdentifier) {

        DisassociateOpportunityRequest disassociateOpportunityRequest =
            DisassociateOpportunityRequest.builder()
                .catalog(Constants.CATALOG_TO_USE)
                .opportunityIdentifier(opportunityId)
                .relatedEntityType(entityType)
                .relatedEntityIdentifier(entityIdentifier)
                .build();

        DisassociateOpportunityResponse response =
            client.disassociateOpportunity(disassociateOpportunityRequest);

        return response;
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조에서 [DisassociateOpportunity](#)를 참조하세요.

Python

SDK for Python(Boto3)

기회 및 관련 엔터티 간의 기존 연결을 제거합니다.


```
#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DisassociateOpportunity](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **GetAwsOpportunitySummary** 함께 사용

다음 코드 예시는 GetAwsOpportunitySummary의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

AWS기회의 요약을 검색합니다.

```

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;
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAwsOpportunitySummary](#)를 참조하세요.

Python

SDK for Python(Boto3)

AWS기회의 요약을 검색합니다.

```

#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAwsOpportunitySummary](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **GetEngagementInvitation** 함께 사용

다음 코드 예시는 GetEngagementInvitation의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

에서AWS파트너와 공유하는 참여 초대의 세부 정보를 검색합니다.

```
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;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetEngagementInvitation](#)을 참조하세요.

Python

SDK for Python(Boto3)

에서AWS파트너와 공유하는 참여 초대의 세부 정보를 검색합니다.

```

#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetEngagementInvitation](#)을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **GetOpportunity** 함께 사용

다음 코드 예시는 GetOpportunity의 사용 방법을 보여 줍니다.

.NET

SDK for .NET

기회를 가져옵니다.

```
// 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.");
    }
}
}
}
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [GetOpportunity](#)를 참조하세요.

Go

SDK for Go V2

기회를 가져옵니다.

```

package main

import (
    "context"
    "encoding/json"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go-v2/aws"

```

```

    "github.com/aws/aws-sdk-go-v2/config"
    "github.com/aws/aws-sdk-go-v2/service/partnercentralselling"
)

func main() {
    config, err := config.LoadDefaultConfig(context.TODO())

    if err != nil {
        log.Fatal(err)
    }

    config.Region = "us-east-1"

    client := partnercentralselling.NewFromConfig(config)

    output, err := client.GetOpportunity(context.TODO(),
        &partnercentralselling.GetOpportunityInput{
            Identifier: aws.String("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))
}

```

- API 세부 정보는 AWS SDK for Go API 참조의 [GetOpportunity](#)를 참조하세요.

Java

SDK for Java 2.x

기회를 가져옵니다.

```

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;
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetOpportunity](#)를 참조하세요.

Python

SDK for Python(Boto3)

기회를 가져옵니다.

```
#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetOpportunity](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **ListEngagementInvitations** 함께 사용

다음 코드 예시는 ListEngagementInvitations의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

파트너에게 보낸 참여 초대 목록을 검색합니다.

```
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;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ListEngagementInvitations](#)을 참조하세요.

Python

SDK for Python(Boto3)

파트너에게 보낸 참여 초대 목록을 검색합니다.

```

#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [ListEngagementInvitations](#)을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **ListOpportunities** 함께 사용

다음 코드 예시는 ListOpportunities의 사용 방법을 보여 줍니다.

.NET

SDK for .NET

기회를 나열합니다.

```
// 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.");
    }
}
}
}
}
}

```

- API 세부 정보는 AWS SDK for .NET API 참조의 [ListOpportunities](#)를 참조하세요.

Go

SDK for Go V2

기회를 나열합니다.

```

package main

import (
    "context"

```

```

"encoding/json"
"fmt"
"log"

"github.com/aws/aws-sdk-go-v2/aws"
"github.com/aws/aws-sdk-go-v2/config"
"github.com/aws/aws-sdk-go-v2/service/partnercentralselling"
)

func main() {
    config, err := config.LoadDefaultConfig(context.TODO())

    if err != nil {
        log.Fatal(err)
    }

    config.Region = "us-east-1"

    client := partnercentralselling.NewFromConfig(config)

    output, err := client.ListOpportunities(context.TODO(),
        &partnercentralselling.ListOpportunitiesInput{
            MaxResults: aws.Int32(2),
            Catalog:   aws.String("AWS"),
        })

    if err != nil {
        log.Fatal(err)
    }

    jsonOutput, err := json.MarshalIndent(output, "", "    ")
    fmt.Println(string(jsonOutput))
}

```

- API 세부 정보는 AWS SDK for Go API 참조의 [ListOpportunities](#)를 참조하세요.

Java

SDK for Java 2.x

기회를 나열합니다.

```
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_T0_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_T0_USE)
            .maxResults(5)
            .nextToken(response.nextToken())
            .build();
        response = client.listOpportunities(listOpportunityRequest);
        opportunitySummaries.addAll(response.opportunitySummaries());
    }

    client.close();

    return opportunitySummaries;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ListOpportunities](#)를 참조하세요.

Python

SDK for Python(Boto3)

기회를 나열합니다.

```

#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [ListOpportunities](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **ListSolutions** 함께 사용

다음 코드 예시는 ListSolutions의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

파트너가 Partner Central에 등록한 파트너 솔루션 목록을 검색합니다.

```

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;
    }
}
```

- API에 대한 세부 정보는 [AWS SDK for Java 2.x API 참조](#)의 ListSolutions를 참조하세요.

Python

SDK for Python(Boto3)

파트너가 Partner Central에 등록된 파트너 솔루션 목록을 검색합니다.

```
#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [ListSolutions](#)을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **RejectEngagementInvitation** 함께 사용

다음 코드 예시는 RejectEngagementInvitation의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

AWS공유된 EngagementInvitation을 거부합니다.

```

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;
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [RejectEngagementInvitation](#)을 참조하세요.

Python

SDK for Python(Boto3)

AWS공유된 EngagementInvitation을 거부합니다.

```
#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [RejectEngagementInvitation](#)을 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **StartEngagementByAcceptingInvitationTask** 함께 사용

다음 코드 예시는 StartEngagementByAcceptingInvitationTask의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

EngagementInvitation을 수락하여 참여를 시작합니다.

```

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;
    }
}
```


- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartEngagementByAcceptingInvitationTask](#)를 참조하세요.

Python

SDK for Python(Boto3)

EngagementInvitation을 수락하여 참여를 시작합니다.

```
#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartEngagementByAcceptingInvitationTask](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **StartEngagementFromOpportunityTask** 함께 사용

다음 코드 예시는 StartEngagementFromOpportunityTask의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

참여 초대를 수락하고 파트너의 시스템에서 해당하는 기회를 생성하여 기존 기회에서 참여 프로세스를 시작합니다.

```
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;
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartEngagementFromOpportunityTask](#)를 참조하세요.

Python

SDK for Python(Boto3)

참여 초대를 수락하고 파트너의 시스템에서 해당하는 기회를 생성하여 기존 기회에서 참여 프로세스를 시작합니다.

```
#!/usr/bin/env python

"""
Purpose
PC-API -11 Associating a product
PC-API -12 Associating a solution
PC-API -13 Associating an offer
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def start_engagement_from_opportunity_task(identifier):

    start_engagement_from_opportunity_task_request = {
        "AwsSubmission": {
            "InvolvementType": "Co-Sell",
            "Visibility": "Full"
        },
        "Catalog": CATALOG_TO_USE,
        "Identifier" : identifier,
        "ClientToken": "test-annjqwesdsd99"
```

```

    }
    try:
        # Perform an API call
        response =
partner_central_client.start_engagement_from_opportunity_task(**start_engagement_from_op
        return response

    except ClientError as err:
        # Catch all client exceptions
        print(err.response)
        return None

def usage_demo():
    identifier = "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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartEngagementFromOpportunityTask](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#) [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDK와 **UpdateOpportunity** 함께 사용

다음 코드 예시는 UpdateOpportunity의 사용 방법을 보여 줍니다.

Java

SDK for Java 2.x

기회를 업데이트합니다.

```
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_T0_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;
}
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [UpdateOpportunity](#)를 참조하세요.

Python

SDK for Python(Boto3)

기회를 업데이트합니다.

```

#!/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()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [UpdateOpportunity](#)를 참조하세요.

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS SDKs를 사용한 Partner Central 시나리오

다음 코드 예제에서는 Partner Central AWS SDKs에서 일반적인 시나리오를 구현하는 방법을 보여줍니다. 이러한 시나리오에서는 Partner Central 내에서 또는 다른AWS 서비스와 결합된 상태에서 여러 함수를 직접적으로 호출하여 특정 작업을 수행하는 방법을 보여줍니다. 각 시나리오에는 전체 소스 코드에 대한 링크가 포함되어 있습니다. 여기에서 코드를 설정 및 실행하는 방법에 대한 지침을 찾을 수 있습니다.

시나리오는 컨텍스트에 맞는 서비스 작업을 이해하는 데 도움이 되도록 중급 수준의 경험을 대상으로 합니다.

예시

- [기회에 연결된 엔터티 업데이트](#)

기회에 연결된 엔터티 업데이트

다음 코드 예제는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 이전 엔터티의 연결을 해제합니다.
- 새 엔터티를 연결합니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [시나리오](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

기회에 연결된 엔터티 업데이트

```
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-0011111";

        disassociateOpportunityResponse(opportunityId, entityType,
            originalEntityIdentifier );
        AssociateOpportunityResponse associateOpportunityResponse =
            associateOpportunityResponse(opportunityId, entityType, newEntityIdentifier );

        ReferenceCodesUtils.formatOutput(associateOpportunityResponse);
    }

    private static AssociateOpportunityResponse associateOpportunityResponse(String
        opportunityId, String entityType, String entityIdentifier) {

        AssociateOpportunityRequest associateOpportunityRequest =
            AssociateOpportunityRequest.builder()
                .catalog(Constants.CATALOG_TO_USE)
                .opportunityIdentifier(opportunityId)
                .relatedEntityType(entityType)
                .relatedEntityIdentifier(entityIdentifier)
                .build();

        AssociateOpportunityResponse response =
            client.associateOpportunity(associateOpportunityRequest);

        return response;
    }
}
```



```

private static DisassociateOpportunityResponse
disassociateOpportunityResponse(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;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 다음 주제를 참조하세요.
- [AssociateOpportunity](#)
- [DisassociateOpportunity](#)

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS코드 예 리포지토리](#)에서 전체 예를 찾고 설정 및 실행하는 방법을 배워보세요.

기회에 연결된 엔터티 업데이트

```
#!/usr/bin/env python

"""
Purpose
PC-API -17 Replacing a solution
"""

import logging
import boto3
import utils.helpers as helper
from botocore.client import ClientError

from utils.constants import CATALOG_TO_USE

serviceName = "partnercentral-selling"

partner_central_client = boto3.client(
    service_name=serviceName,
    region_name='us-east-1'
)

def replace_solution(original_entity_identifier, new_entity_identifier,
    opportunityIdentifier):
    disassociate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : "Solutions",
        "RelatedEntityIdentifier" : original_entity_identifier
    }

    associate_opportunity_request = {
        "Catalog": CATALOG_TO_USE,
        "OpportunityIdentifier" : opportunityIdentifier,
        "RelatedEntityType" : "Solutions",
        "RelatedEntityIdentifier" : new_entity_identifier
    }
    try:
        # Perform an API call
        response =
partner_central_client.disassociate_opportunity(**disassociate_opportunity_request)
        response =
partner_central_client.associate_opportunity(**associate_opportunity_request)
        return response
```

```
except ClientError as err:
    # Catch all client exceptions
    print(err.response)

def usage_demo():
    original_entity_identifier = "S-0049999"
    new_entity_identifier = "S-0050014"
    opportunityIdentifier = "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_identifier,
    new_entity_identifier, opportunityIdentifier))

if __name__ == "__main__":
    usage_demo()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 다음 주제를 참조하세요.
 - [AssociateOpportunity](#)
 - [DisassociateOpportunity](#)

AWS SDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWS SDK에서AWS Partner Central API 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

릴리스 노트

이 페이지에는 최신 업데이트부터 AWS Partner Central API 설명서의 주요 변경 사항에 대한 요약이 포함되어 있습니다.

수정 기록

변경	설명	Date
RC 6.0	AWS Partner Central에서 새로운 파트너 수익 측정 APIs 추가했습니다.	2026-06-30
RC 5.3	ACE Resonate 업데이트: GetAwsOpportunitySummary 응답이 업데이트되었습니다.	2026-06-16
RC 5.2	- EUR 통화 지원: CreateEngagementInvitation 이제 MRR 통화 코드에 USD CreateOpportunity , UpdateOpportunity 및 를 EUR 추가로 수락합니다. AWS 파티션이 aws-eusc (AWS European Sovereign Cloud)인 경우 통화 코드는 여야 합니다 EUR. - 읽기 작업의 EUR: 이제 GetOpportunity , 및 ListOpportunities 가 aws-eusc 파티션 EUR의 기획에 대해 CurrencyCode 필드에 GetAwsOpportunitySummary 반환 됩니다.	2026-03-30

변경	설명	Date
RC 5.1	• EUR 통화 지원: CreateEngagementInvitation 이제 MRR 통화 코드에 USD CreateOpportunity , UpdateOpportunity 및 를 EUR 추가로 수락합니다. AWS 파티션이 aws-eusc (AWS European Sovereign Cloud)인 경우 통화 코드는 여야 합니다EUR. • 읽기 작업의 EUR: 이제 GetOpportunity , 및 ListOpportunities 가 aws-eusc 파티션EUR의 기회에 대해 CurrencyCode 필드에 GetAwsOpportunitySummary 반환 됩니다.	2026-03-30

변경	설명	Date
RC 5	- 계정 API 도입: AWS Partner Central에 새 계정 API를 추가하여 파트너가 계정 정보, 등록, 프로필, 도메인 관리 및 파트너 연결을 관리할 수 있도록 했습니다. - 계정 API 참조 설명서: 파트너 등록, 프로필 관리, 도메인 확인 및 계정 연결 워크플로를 포함하여 계정 API에 대한 포괄적인 API 참조 설명서가 추가되었습니다. - 계정 API 로깅: API 사용량을 감사하고 추적하는 데 도움이 되도록 계정 API에 대한 로깅 지원 및 설명서가 추가되었습니다. - 계정 API 알림: 계정 API에 대한 이벤트 알림 지원이 추가되어 파트너가 파트너 계정 연결을 포함한 계정 관련 활동에 대한 실시간 업데이트를 받을 수 있습니다. - 계정 API 할당량: 계정 API에 대한 서비스 할당량 설명서가 추가되었습니다. - 계정 API에 대한 샌드박스 테스트: 계정 API에 대한 샌드박스 테스트 기능이 추가되어 파트너가 비프로덕션 환경에서 계정 관리 워크플로를 테스트할 수 있습니다.	2025-11-30

변경	설명	Date
	• 혜택 API 도입: AWS Partner Central에 새로운 혜택 API를 추가하여 파트너가 중앙 집중식 인터페이스를 통해 AWS파트너 프로그램 전체에서 혜택을 검색, 신청 및 관리할 수 있도록 했습니다. • 혜택 API 참조 설명서: 혜택 애플리케이션, 할당 및 이행 프로세스를 다루는 혜택 API에 대한 포괄적인 API 참조 설명서가 추가되었습니다. • 이점 API 로깅: API 사용량을 감사하고 추적하는 데 도움이 되도록 이점 API에 대한 로깅 지원 및 설명서가 추가되었습니다. • 혜택 API 알림: 혜택 API에 대한 이벤트 알림 지원이 추가되어 파트너가 혜택 애플리케이션 및 할당에 대한 실시간 업데이트를 받을 수 있습니다. • Benefits API 할당량: Benefits API에 대한 서비스 할당량 설명서가 추가되었습니다. • 혜택 API에 대한 샌드박스 테스트: 파트너가 비프로덕션 환경에서 혜택 관리 워크플로를 테스트할 수 있도록 혜택 API에 대한 샌드박스 테스트 기능이 추가되었습니다.	

변경	설명	Date
	• 판매 API 업데이트: 리드 관리 워크플로를 지원하도록 판매 API 참여 작업 개선. • API 설명서 업데이트 판매: 향상된 구조와 리드 관리를 위한 업데이트된 워크플로로 판매 API 설명서를 개선했습니다. • API 알림 업데이트 판매: 참여 및 참여 초대 이벤트에 대한 이벤트 알림 설명서가 개선되었습니다. • API 구조 확장: 각 APIs의 권한, 로깅, 알림, 할당량 및 샌드박스 테스트에 대한 전용 섹션이 있는 4개의 개별 API(계정, 혜택, 판매 및 채널)를 지원하도록 설명서 구조를 확장했습니다. • 향상된 API 사용 설명서: 자세한 워크플로 및 통합 패턴이 포함된 계정 및 혜택 APIs에 대한 포괄적인 사용 설명서가 추가되었습니다. • 지원되는 리전 설명서: 계정 및 혜택 APIs에 지원되는 AWS리전에 대한 설명서가 추가되었습니다. • 에 대한 새 필터ListOppor tunities : YYYY-MM- DD 형식의 AfterTarg etCloseDate 및 BeforeTargetCloseD ate 필드를 사용하여 예	

변경	설명	Date
	상 종료일을 기준으로 필터링 기회를 활성화하는 TargetCloseDate 필터가 추가되었습니다.	

변경	설명	Date
RC 4	- 채널 API 도입: AWS파트너 센트럴에 새 채널 API를 추가하여 파트너가 채널 관계 및 partner-to-partner 공동 작업을 관리할 수 있도록 했습니다. - 채널 API 참조 설명서: 사용 가능한 모든 작업 및 데이터 모델을 포함하여 채널 API에 대한 포괄적인 API 참조 설명서가 추가되었습니다. - 채널 API 권한: 채널 API와 관련된 IAM 권한 정책 및 액세스 제어 설명서가 추가되었습니다. - 채널 API 로깅: API 사용량을 감사하고 추적하는 데 도움이 되도록 채널 API에 대한 로깅 지원 및 설명서가 추가되었습니다. - 채널 API 알림: 채널 API에 대한 이벤트 알림 지원이 추가되어 파트너가 채널 관련 활동에 대한 실시간 업데이트를 받을 수 있습니다. - 채널 API 할당량: 채널 API에 대한 서비스 할당량 설명서가 추가되었습니다. - 채널 API에 대한 샌드박스 테스트: 채널 API에 대한 샌드박스 테스트 기능이 추가되어 파트너가 비프로덕션	2025-11-19

변경	설명	Date
	환경에서 채널 워크플로를 테스트할 수 있습니다. • API 구조 재구성: 각 API의 권한, 로깅, 알림 및 할당량에 대한 전용 섹션을 사용하여 판매 API와 채널 API를 구별하도록 설명서를 재구성했습니다. • 지원되는 리전 설명서: 판매 및 채널 APIs 모두에 대해 지원되는 AWS 리전에 대한 설명서가 추가되었습니다.	

변경	설명	Date
RC 3	- 업데이트된 API 권한 정책: 이 릴리스에 도입된 새 작업을 반영하도록 정책이 업데이트되었습니다. IAM 역할 및 권한이 적절히 조정되었는지 확인합니다. - 사용자 지정 헤더 사용 세부 정보: 사용자 지정 헤더를 사용하여 API 사용을 감사하고 측정하는 방법에 X-Amzn-User-Agent 대한 세부 정보가 추가되었습니다. - 작업을 비동기 작업으로 대체: 를 비동기 작업 <code>AcceptOpportunityEngagementInvitation</code> 으로 대체 <code>StartEngagementByAcceptingInvitationTask</code> 하고 RC2와 관련하여 페이로드 형식이 변경되었습니다. - 개체 및 작업 이름 변경: <code>OpportunityEngagementInvitation</code> 로 변경되었습니다 <code>EngagementInvitation</code> . 이러한 작업의 속성도 새 개체에 맞게 변경되었습니다. 다음 작업이 대체되었습니다. <code>GetOpportunityEngagementInvitation</code> 는 이제 <code>GetEngage</code>	2024-10-17

변경	설명	Date
	mentInvitation . 초대의 여러 속성이 변경, 추가 또는 제거되었습니다. • ListOpportunityEngagementInvitations 는 이제 ListEngagementInvitations . • RejectOpportunityEngagementInvitation 는 이제 RejectEngagementInvitation . • AssignOpportunity 파라미터 업데이트: AssignOpportunity 이 이제 Assignee 대신를 사용합니다AssigneeEmail . 그에 따라 구현을 업데이트합니다. • SubmitOpportunity 기능 변경: SubmitOpportunity 는 이제 비동기 작업에 의해 수행됩니다StartEngagementFromOpportunityTask . SubmitOpportunity 이 이제 InvolvementType 및 Visibility . • 속성 이름 변경 및 확장:가 로 EstimatedAwsMonthlyRevenue 변경되었습니다ExpectedCustomerSp	

변경	설명	Date
	end .에는 ExpectedCustomerSpend 이 제 Frequency 및와 같 은 새 속성이 포함됩니 다TargetCompany . • AWS기회 요약을 위한 새로 운 작업: 기회 가시성은 이 제 라는 별도의 작업을 통 해 제공됩니다GetAwsOpp portunitySummary . • 기회 작업을 위한 워크플로 업데이트: 기회 작업을 위한 워크플로가이 릴리스에 도입 된 새 작업을 사용하도록 업 데이트되었습니다. • 기회 작업을 위한 워크플 로 업데이트: 기회 작업을 위한 워크플로가를 사용하 도록 업데이트Engagemen tInvitations 되어 AWS에서 제공하는 처리 기 회가 개선되었습니다. • 업데이트 추적을 위한 워크 플로 업데이트: 업데이트 추 적 워크플로가 이제 기회 업 데이트로 작업하므로 기회 상태 모니터링의 명확성과 효율성이 향상됩니다. • 작업 나열을 위한 새 필 터: ListSolutions 및 ListOpportunities 작 업에 새 필터를 사용할 수 있 습니다.	

변경	설명	Date
	• ListSolutions 의 경우 새 필터는 FilterCategory , FilterIdentifier 및 입니다 FilterStatus . • ListOpportunities 의 경우 새 필터는 FilterCustomerCompany , FilterIdentifier , FilterLastModifiedDate , FilterLifeCycleApprovalStatus , 및 FilterLifeCycleStage 입니다 FilterOrigin . • AccessControl 하위 객체 제거: 기회 모델의 AccessControl 하위 객체가 제거되었습니다. NationalSecurity 는 이제 최상위 속성입니다. • 로 재배치된 속성 GetAwsOpportunitySummary : InvolvementType 이제 AWSOpportunityTeam , 인사이트 (EngagementScore , NextBestAction), 오리진 및와 같은 속성을 GetAwsOpportunitySummary 대신를 통해 사용	

변경	설명	Date
	할 수 있습니다GetOpportunity . • InvolvementType 필드 소개: 코셀 기회와 가시성 전용 기회를 구분하는 InvolvementType 필드를 도입했습니다. • 이벤트 이름 업데이트: 변경 사항을 반영하도록 이벤트가 로 업데이트되었습니다. 예를 들어, 생성된 기회 참여 초대는 이제 참여 초대가 생성되었습니다. • 응답 페이로드 업데이트: 이제 모든 응답 페이로드에 ID 및/또는 ARN이 포함됩니다. • 페이로드 업데이트 요청: 이제 모든 요청 페이로드가 ID 또는 ARN을 수락하는 식별자를 입력으로 사용합니다. • 참여 엔터티 업데이트: 참여 엔터티의 제목 속성 이름이 변경되었습니다EngagementTitle . • 필터 접두사 제거: 모든 필터에 필터 접두사가 더 이상 포함되지 않으므로 필터링 메커니즘이 간소화됩니다. • StartEngagementFromOpportunity 작업 업데이트: 속성이 로 AwsInvolvement 변경되었습니다AwsSubmission .	

변경	설명	Date
	- 초대 프로젝트 세부 정보 업데이트: <code>Invitation.ProjectDetails.TargetCompletionDate</code> 이름이 변경되었습니다. <code>Invitation.ProjectDetails.EstimatedCompletionDate</code>. - 수익 추정 업데이트:는 이제 <code>Invitation.PartnerRevenueEstimate</code> 입니다. <code>ExpectedCustomerSpend</code> . 데이터 형식이 객체 하나에서 객체 하나가 포함된 배열로 변경되었습니다. - 수신자 계정 업데이트: 필드 <code>ReceiverAccount</code> 이름이 변경되었습니다. <code>AccountReceiver</code> . - 참여 초대 새 속성: 참여 초대 에 <code>SenderContact</code> 속성이 도입되었습니다. <code>Customer.Contact</code> 는 이제 객체 대신 배열입니다. <code>OpportunityOwner</code> , <code>PartnerAccountManager</code> 는 로 지원되는 값입니다. <code>BusinessTitle</code> . <code>PartnerOpportunityTeam</code> 가 <code>OpportunityTeam</code> 로 바뀝니다. <code>APNPrograms</code> 가 이제 <code>ApnPrograms</code> 입니다.	

변경	설명	Date
	• 참여 초대 상태가 업데이트되었습니다. • ListEngagementInvitations 이제가 ParticipantType 로 필요합니다RECEIVER.	

변경	설명	Date
RC 2	• OpportunityEngagementInvitation이라는 새 엔티티를 도입했습니다. • ListOpportunityEngagementInvitations라는 새 작업을 도입했습니다. • GetOpportunityEngagementInvitation이라는 새 작업을 도입했습니다. • “기회 참여 초대 생성됨”이라는 새 이벤트를 도입했습니다. • AcceptOpportunity를 AcceptOpportunityEngagementInvitation으로 대체했습니다. • RejectOpportunity를 RejectOpportunityEngagementInvitation으로 대체했습니다. • “기회 수락됨”을 “기회 참여 초대 수락됨”으로 대체했습니다. • "Opportunity Rejected"를 "Opportunity Engagement Invitation Rejected"로 대체했습니다. • PrimaryNeedsFromAWS 가 PrimaryNeedsFromAws로 변경되었습니다. • AWSOpportunityTeam 가 로 변경되었습니다.	2024-08-07

변경	설명	Date
	다AwsOpportunityTeam . • AWSSalesLifeCycle 가 로 변경되었습니다다AwsSalesLifeCycle . • PrimaryNeedsFromAWSChangeReason 가 로 변경되었습니다다PrimaryNeedsFromAwsChangeReason . • AWSAccountId 가 로 변경되었습니다다AwsAccountId . • ExpectedMonthlyAWSRevenue 가 로 변경되었습니다다ExpectedMonthlyAwsRevenue . • AWSFundingUsed 가 로 변경되었습니다다AwsFundingUsed . • AWSMarketplaceOffers 가 로 변경되었습니다다AwsMarketplaceOffers . • Country가 로 변경되었습니다다CountryCode . • PartnerOpportunityContact 가 로 변경되었습니다다PartnerOpportunityTeam . • 필드를 Contact.Title 로 업데이트했습니다	

변경	설명	Date
	다Contact.BusinessTitle . • 필드를 ContactInfo 로 업데이트했습니다. - 다OwnerInfo . • AWS팀이 맵에서 목록으로 변경되었습니다. • 수락된 RejectionReasons 목록에 대한 세부 정보가 추가되었습니다. • 클라이언트 토큰 사용 방법에 대한 정보를 제공했습니다. • UpdateOpportunity 사용 방법에 대한 세부 정보가 포함되어 있습니다. • AWS제품 및 파트너 솔루션 사용에 대한 지침이 추가되었습니다. • OpportunityEngagementInvitation 엔터티에 대한 세부 정보를 제공했습니다. • 수락된 값의 수정된 목록을 포함하도록 StateOrRegion 을 업데이트했습니다. • 성능 및 오류 메시징을 개선하기 위해 여러 버그 수정을 구현했습니다.	

변경	설명	Date
RC 1	• [분리] ApprovalStatus를 ReviewStatus로 이름 변경: 용도를 더 잘 반영하기 위해 ApprovalStatus 필드의 이름을 ReviewStatus로 변경했습니다. 또한 초안 기회를 나타내기 위해 새로운 상태인 제출 보류를 도입합니다. ReviewStatus는 이제 제출 대기 중, 제출됨, 검토 중, 승인됨, 거부됨 및 작업 필요 값을 수락합니다. 이전 초안 상태는 제출 보류 중으로 바뀝니다. • [중요] SubmitOpportunity 작업 소개: 새 작업인 SubmitOpportunity가 도입되었습니다. CreateOpportunity도 기회를 제출하는 현재 동작과 달리 이제 솔루션 또는 AWS 제품을 즉시 연결하지 않고도 제출 보류 중 상태에서 기회를 생성할 수 있습니다. 제출을 완료하려면 AssociateOpportunity 작업을 사용하여 솔루션 또는 제품을 연결한 다음 SubmitOpportunity를 연결합니다. 이 분리는 제출 전에 더 나은 API 응답 성능과 유연한 준비 단계를 제공합니다. • [분리] 카탈로그 파라미터 요구 사항 및 샌드박스 엔드포인트 사용 중	2024-06-21

변경	설명	Date
	단: '감마' 엔드포인트(<code>partnercentral-selling-gamma.us-east-1.amazonaws.com</code>)는 7/30/2024 이후에는 사용할 수 없게 됩니다. 이제 모든 API 작업에는 샌드박스 또는 카탈로그에서 작업을 수행할지 여부를 지정하는 AWS 카탈로그 파라미터가 필요합니다. 이제 알림 규칙이 <code>environmentName</code> 대신 카탈로그를 기준으로 필터링 됩니다. • [분리] <code>OpportunityIdentifier</code>를 식별자로 이름 바꾸기: 작업 <code>AssignOpportunity</code>, <code>AcceptOpportunity</code> 및 <code>RejectOpportunity</code>는 이제 <code>UpdateOpportunity</code>와 일관성을 유지하기 위해 <code>OpportunityIdentifier</code> 대신 식별자를 사용합니다. • <code>APNProgram CustomerUseCase</code> 및 <code>UseCase</code>의 유형 Enum에서 String으로 변경: 값 변경과 관련된 위험을 줄이기 위해 <code>Project.APNPrograms, Marketing.ActivityUseCases</code> 및 <code>Project.CustomerUseCase</code>를 열거 유형에서 문자열 유형으로 변환합니다. 이러한 필드는 사전 정의된 목록의 값만 허용하지만 SDKs에서	

변경	설명	Date
	는 기본적으로 사용할 수 없습니다. - 향상된 오류 처리: 더 쉽게 디버깅하고 문제를 더 빠르게 해결할 수 있도록 APIs 내의 오류 처리가 크게 개선되었습니다. 새로운 오류 처리는 오류를 1의 두 단계로 구성합니다. 요청 검증 실패: 데이터 유형 및 형식 또는 2를 확인합니다. 비즈니스 검증 실패: 페이로드의 모든 문제는 오류가 있는 필드를 지정하여 하나의 응답으로 나열됩니다. 이 통합 피드백을 통해 오류를 보다 효율적으로 해결하고 수정할 수 있습니다. 오류 코드와 형식이 표준화되었습니다. - 클라이언트 소스 식별자: 파트너 기능에는 트래픽의 소스를 식별하는 데 도움이 되는 X-Amzn-User-Agent 헤더가 요청에 포함됩니다. [솔루션 공급자를 포함한 솔루션 이름] [버전] 형식을 사용합니다. 예:AWS파트너 CRM 커넥터 v2.0.1 - 버그 수정, 이미 배포됨, SDK 변경 없음] 기회 생성 및AWS계정 ID: 이제 파트너는 오류 없이AWS계정 ID를 null에서 유효한 값으로 변경하면서 기회를 시작할 수 있	

변경	설명	Date
	습니다. 이전에는 버그로 인해 파트너가 이러한 작업을 별도로 수행하지 못했습니다. • 버그 수정, 이미 배포됨, SDK 변경 없음] 솔루션과 새로운 기회 연결: 파트너는 솔루션을 새로 생성된 기회와 연결할 수 있습니다. 이전에는 파트너가 새로운 기회를 만들고 동시에 솔루션을 연결했을 때 솔루션 정보가 손실되었습니다. 이 문제가 해결되었습니다. • 버그 수정, 이미 배포됨, SDK 변경 없음] 기회 소유자의 세부 정보 표시: 일부 기회가 예상 "이메일", "FirstName" 및 "LastName" 필드 대신 결합된 "LastName" 필드에 소유자의 세부 정보를 표시하는 문제가 수정되었습니다. FirstName LastName • 버그 수정, 이미 배포됨, SDK 변경 없음] Customer.Account.WebsiteUri 필드 선택 사항: AccessControls.NationalSecurity 필드가 "Yes"인 경우 Customer.Account.WebsiteUri 필드가 선택 사항으로 설정되었으므로 API 동작을 기존 UI 동작과 일치시킵니다. AccessControls.NationalSecurity 필드가 "No" 또	

변경	설명	Date
	는 null인 경우 비즈니스 검증으로 WebsiteUrl 필드가 필요합니다. • 버그 수정, 이미 배포됨, SDK]AWS컨설팅 파트너의 계정 ID 요구 사항 변경 없음:AWS계정 ID 필드를 컨설팅 파트너의 선택 사항으로 표시한 UI의 불일치가 해결되었습니다. 이제 설명서에 따라 조건부 필수 필드입니다. • 버그 수정, 6월 19일에 배포 예정, SDK 변경 사항 없음] 공동 판매 기회의 스테이지에서 종료된 손실: 이제 파트너는 종료된 손실 이유가 제공되는 한 공동 판매 기회의 스테이지를 종료된 손실로 성공적으로 업데이트할 수 있습니다. 이렇게 하면 “기회를 종료할 때 필수 필드 누락: 누락된 사유 종료가 필요합니다.”라는 이전 오류가 해결됩니다. • 설명서] 설명서 개선 사항: 변경 사항을 반영하기 위해 설명서를 몇 가지 개선했습니다. • 보류 중인 수정 사항 및 예정된 기능 목록에 대해 알려진 문제: • 솔루션 또는 제품으로 AssociateOpportunity 작	

변경	설명	Date
	업을 수행하는 동안 제안의 오류 메시지가 잘못 표시됩니다. - 샌드박스에서 원격으로 기회를 생성하고AWS검증 프로세스를 시뮬레이션할 수 있는 기능 - 우편 번호가 미국의 경우 XXXXX-XXXX 형식일 때 null을 반환합니다. "지원 필요 없음"으로 표시된 기회에는 다음 단계 및 다음 단계 기록을 사용할 수 없습니다AWS. - 고객 비즈니스 문제 설명이 255자를 초과하는 기록 기회를 가져올 수 없습니다. - 제출 보류 중인 기회를 삭제할 수 있습니다. - 데이터 모델에서는 LeadSource를 사용할 수 없습니다.	

변경	설명	Date
베타 5	• LifeCycle.ApprovalStatus를 LifeCycle.ReviewStatus로 변경 • Insights.ReviewComments를 LifeCycle.ReviewComments로 변경 • LifeCycle.ReviewStatusReason 추가(새 속성) • PartnerAcceptance.Status에 새 값 "Pending" 추가 • dotnet, java, javascript, python, ruby용 SDKs를 릴리스했습니다. • 위의 변경 사항을 반영하도록 API 참조 안내서를 업데이트했습니다.	2024-04-19
베타 4	• 샌드박스 테스트: 샌드박스에서 기회 이벤트 알림을 테스트할 수 있는 기능이 추가되었습니다. • 설명서 추가: 기회 이벤트 알림 샘플 규칙 도입. • SDK 릴리스: Java(V2), Javascript(V2), DotNet(V3) 및 Python(V3)용 SDKs 릴리스가 추가되었습니다. • 업데이트된 날짜 형식: ISO 표준을 따르도록 업데이트되었습니다.	2024-03-04

변경	설명	Date
베타 3	• 변경 로그 파일 추가: 업데이트 및 변경 사항을 더 잘 추적할 수 있도록 변경 로그 파일을 도입했습니다. • 업데이트된 API 참조 가이드 (PDF): PDF 형식의 API 참조 가이드의 새 버전을 릴리스했습니다. • 업데이트된 API 참조 가이드 (웹 버전): API 참조 가이드의 웹 버전을 업데이트했습니다. • DotNet SDK 추가: DotNet SDK를 릴리스하여 다양한 프로그래밍 환경에 대한 지원을 확장했습니다. • Python Boto 3 SDK(V2): Python Boto 3 SDK 버전 2를 릴리스하여 새로운 기능과 개선 사항을 도입했습니다.	2024-01-24
베타 2	• API 참조 가이드(웹 버전): API 참조 가이드(웹 버전)의 새 버전을 릴리스했습니다. 참고:이 버전을 사용하려면 적절한 기능을 위해 추출 후 로컬 웹 서버를 실행해야 합니다.	2024-01-07

변경	설명	Date
베타 1	• Boto3(초기 릴리스): 서비스용 Python SDK인 Boto3의 초기 릴리스를 시작했습니다. • 베타 프로그램 가이드 추가: 베타 테스트 프로그램 참가자를 위한 가이드가 도입되었습니다. • API 참조 가이드(PDF): API 참조 가이드의 첫 번째 버전을 PDF 형식으로 업로드하여 API에 대한 포괄적인 설명서를 제공했습니다.	2023-12-15
베타 0	이 최초 릴리스는 AWS Partner Central에서 파트너 참여 및 기회를 관리하고 최적화하는 기능 모음을 제공합니다.	2023-11-15

문서 기록

다음은이 API 참조에 대한 설명서 릴리스 목록입니다.

변경 사항	설명	날짜
파트너 수익 측정 APIs에 대한 설명서 추가	AWS파트너 중앙에서 파트너 수익 측정(PRM)을 위한 새로운 APIs를 추가했습니다.	2026년 6월 30일
ACE Resonate에 대한 설명서 업데이트	AWS Cosell 모션을 지원하는 GetAwsOpportunitySummary 작업에 대한 파트너 중앙 업데이트 API 응답입니다.	2026년 6월 16일
파트너 중앙 MCP 서버에 대한 설명서 추가	AWS파트너 중앙 에이전트 MCP 서버는 모델 컨텍스트 프로토콜(MCP)을 통해 파트너 중앙 도구를 제공합니다.	2026년 3월 16일
RC 5 릴리스	AWS Partner Central API의 후보 5 버전이 릴리스되었습니다. AWS Partner Central에 새 계정 API를 도입하여 파트너가 계정 정보, 등록, 프로필, 도메인 관리 및 파트너 연결을 관리할 수 있도록 했습니다. AWS Partner Central에 새로운 Benefits API를 도입하여 파트너가 중앙 집중식 인터페이스를 통해 AWS파트너 프로그램 전체에서 혜택을 검색, 신청 및 관리할 수 있도록 했습니다. 리드 관리 워크플로를 지원하도록 API 참여 작업 판매를 개선했습니다. 자세한 워크플로 및 통합 패턴이 포함된 계정,	2025년 11월 30일

혜택 API에 대한 포괄적인 사용 설명서 및 APIs 추가되었습니다. 리드 컨텍스트를 사용한 참여 관리를 위한 향상된 판매 API 설명서.

RC 4 릴리스

AWS Partner Central API의 후보 4 버전이 릴리스되었습니다. 채널 관계 및 partner-to-partner 공동 작업을 관리하기 위한 채널 API를 도입하여 파트너가 Billing Transfer를 사용하여 최종 고객과 거래할 때 프로그램 혜택을 받을 수 있도록 했습니다. 여러 AWS Partner Central APIs를 지원하도록 설명서가 재구성되었습니다.

2025년 11월 19일

멀티 파트너 기회에 대한 설명서 추가

다중 파트너 기회(평가판)는 AWS Partner Central의 통합 환경으로, AWS 파트너가 다른 파트너를 찾아 연결하고 고객 거래에서 함께 판매할 수 있습니다.

2024년 12월 4일

RC 3 릴리스

AWS Partner Central API의 후보 3 버전 릴리스

2024년 10월 17일

RC 2 릴리스

AWS Partner Central API의 후보 2 버전 릴리스

2024년 8월 7일

RC 1 릴리스

AWS Partner Central API의 후보 1 버전 릴리스

2024년 6월 21일

베타 5 릴리스

AWS Partner Central API의 베타 5 출시

2024년 4월 19일

베타 4 릴리스	AWS Partner Central API의 베타 4 출시	2024년 3월 4일
베타 3 릴리스	AWS Partner Central API의 베타 3 출시	2024년 1월 24일
베타 2 릴리스	AWS Partner Central API의 베타 2 출시	2024년 1월 7일
베타 1 릴리스	AWS Partner Central API의 베타 1 출시	2023년 12월 15일
베타 0 릴리스	AWS Partner Central API 최초 릴리스	2023년 11월 15일