



개발자 가이드

AWS Marketplace



AWS Marketplace: 개발자 가이드

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon의 상표 및 트레이드 드레스는 Amazon 외 제품 또는 서비스와 함께, Amazon 브랜드 이미지를 떨어뜨리거나 고객에게 혼동을 일으킬 수 있는 방식으로 사용할 수 없습니다. Amazon이 소유하지 않은 기타 모든 상표는 Amazon과 제휴 관계이거나 관련이 있거나 후원 관계와 관계없이 해당 소유자의 자산입니다.

Table of Contents

.....	viii
환영합니다	1
APIs 정보	2
API 사용	2
카탈로그 API 사용	2
계약 API 사용	18
측정 API 사용	25
권한 API 사용	27
배포 API 사용	27
Discovery API 사용	27
지원되는 리전:	30
카탈로그 API의 리전	30
계약 API의 리전	30
측정 API의 리전	30
권한 부여 API의 리전	32
배포 API의 리전	33
Discovery API의 리전	33
권한	33
카탈로그 API에 대한 액세스	34
계약 API에 대한 액세스	51
측정 API에 대한 액세스	56
권한 API에 대한 액세스	56
배포 API에 대한 액세스	56
규정 준수 API에 대한 액세스	57
Discovery API에 대한 액세스	71
AWS Marketplace 보고 API에 대한 액세스 제어	74
할당량	76
카탈로그 API 할당량	77
계약 API에 대한 할당량	79
권한 부여 API에 대한 할당량	81
배포 API 할당량	81
측정 API의 할당량	82
Discovery API 할당량	83
로깅	84

카탈로그 API 로깅	84
계약 API 로깅	86
측정 API 로깅	93
배포 API 로깅	99
Discovery API 로깅	102
알림	104
카탈로그 API에 대한 알림	104
AWS Marketplace MCP 서버	105
사용 가능한 도구	106
통합 가이드	106
Claude 커넥터(권장)	107
MCP 클라이언트 통합	107
도구 세부 정보	111
get_aws_marketplace_report_guidelines	111
search_aws_marketplace_solutions	113
get_aws_marketplace_solution	114
get_aws_marketplace_related_solutions	117
research_aws_marketplace_solution	119
submit_aws_marketplace_feedback	121
워크플로	122
보고서 생성(권장 사항, 비교, 평가)	122
직접 카탈로그 조회	123
피드백 수집	124
종합 예제	124
오류 처리	125
프로토콜 오류	126
도구 실행 오류	126
유효성 검사 오류	126
API를 판매자로 사용	127
판매자 제품 작업	127
제품 생성	129
제품 세부 정보 업데이트	132
요금 차원 추가	140
요금 차원 업데이트	147
요금 차원 제한	152
대상 지정 구성 업데이트	156

제품 표시 여부 업데이트	159
제품 게시	163
제품 ID 찾기	165
설정 상태 및 오류 변경	171
AMI 기반 제품 작업	178
EC2 Image Builder 구성 요소 제품 작업	211
컨테이너 기반 제품 작업	227
SaaS 제품 작업	265
기계 학습 제품 작업	289
제안 작업	311
비공개 제안 작업	312
재판매 권한 부여 작업	378
채널 파트너 비공개 제안 작업	425
갱신 작업	477
다중 제품 솔루션 작업	479
솔루션 작업	480
제안 세트 작업	514
카탈로그 데이터를 사용하여 인터페이스 구축	537
아키텍처 개요	538
찾아보기 경험 구축	538
빌드 제품 세부 정보 페이지	539
요금 정보 표시	539
모범 사례	540
API를 구매자로 사용	542
제품 및 요금 알아보기	542
카탈로그 찾아보기	542
목록 검색	542
목록 세부 정보 가져오기	543
제품 세부 정보 및 이행 옵션 가져오기	543
제품에 사용 가능한 제안 찾기	544
자세한 요금 조건 가져오기	544
제안 세트 세부 정보 가져오기	544
비공개 제안 요금에 액세스	545
조달 도구와 통합	546
계약 API 작업	546
사전 조건: 제품 및 제안 검색	546

다음표 생성	551
제안 수락	557
기존 계약 교체	557
기존 계약 수정	558
자동 갱신 활성화 또는 비활성화	558
계약 취소	559
계약 검색	559
계약 설명	559
계약 조건 가져오기	560
계약 권한 가져오기	560
등록 토큰 가져오기	560
계약 요금 나열	561
구매 주문 업데이트	561
취소 요청 나열	561
취소 요청 세부 정보 가져오기	562
취소 요청 수락	562
취소 요청 거부	562
결제 요청 나열	562
결제 요청 세부 정보 가져오기	563
결제 요청 수락	563
결제 요청 거부	563
프라이빗 마켓플레이스 작업	564
프라이빗 마켓플레이스 생성	565
프라이빗 마켓플레이스 경험의 브랜딩 변경	569
프라이빗 마켓플레이스 경험 활성화 또는 비활성화	571
사용자 요청 활성화 또는 비활성화	572
프라이빗 마켓플레이스 경험에서 제품 목록 가져오기	573
프라이빗 마켓플레이스에서 제품 추가 또는 제거	576
제품 찾기	577
용 프라이빗 마켓플레이스 작업AWS Organizations	578
보안 주체를 경험에 연결	579
프라이빗 마켓플레이스 경험 보관 및 재활성화	582
프라이빗 마켓플레이스 API의 오류	583
프라이빗 마켓플레이스에서 정의한 개체 유형	585
API를 사용하여 리소스 공유	587
AWS Marketplace 엔터티를 공유하기 위한 사전 조건	587

AWS Marketplace 개체 공유	587
AWS RAM 와를 통한 엔터티 공유의 차이점 AWS Marketplace Catalog API	589
리소스에 읽기 전용 정책 연결	590
리소스에 읽기 및 쓰기 리소스 정책 연결	591
리소스에 설정된 리소스 정책 보기	591
리소스에서 리소스 정책 삭제	592
사용자가 소유하고 공유한 모든 리소스 보기	593
AWSSDKs 작업	595
코드 예제	597
AWS Marketplace 카탈로그 API	602
AMI 제품	605
채널 파트너 제안	663
컨테이너 제품	703
개체	718
제안	728
Products	864
재판매 권한 부여	876
SaaS 제품	973
유틸리티	1030
AWS Marketplace 계약 API	1037
계약	1039
문서 기록	1296
AWS Marketplace Discovery API 릴리스 정보	1300
2026년 Discovery API 릴리스 정보	1300
2024년 Discovery API 릴리스 정보	1301
2022년 Discovery API 릴리스 정보	1301

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

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

AWS Marketplace 개발자 안내서 시작

이 개발자 안내서에서는 AWS Marketplace 판매자가 서비스 APIs 사용하여 제품 수명 주기 및 제안을 통합하고 관리하는 방법을 설명합니다. 또한 구매자가 서비스 APIs 사용하여 타사 소프트웨어, 데이터 및 서비스를 검색하는 방법과 프라이빗 마켓플레이스를 관리하는 방법을 설명합니다.

AWS Marketplace APIs 정보

다음 섹션에서는 APIs에 대한 일반적인 정보를 제공합니다.

주제

- [AWS Marketplace API 사용](#)
- [AWS Marketplace API AWS 리전 에 대해 지원됨](#)
- [AWS Marketplace API에 대한 권한](#)
- [AWS Marketplace API 할당량](#)
- [AWS Marketplace API에 대한 로깅](#)
- [AWS Marketplace API에 대한 알림](#)

AWS Marketplace API 사용

다음 섹션에서는 AWS Marketplace APIs 사용에 대한 정보를 제공합니다.

주제

- [AWS Marketplace 카탈로그 API 사용](#)
- [AWS Marketplace 계약 API 사용](#)
- [AWS Marketplace 측정 API 사용](#)
- [AWS Marketplace 권한 API 사용](#)
- [AWS Marketplace 배포 API 사용](#)
- [AWS Marketplace Discovery API 사용](#)

AWS Marketplace 카탈로그 API 사용

AWS Marketplace Catalog API 서비스는 조직에 AWS Marketplace 대해 AWS 또는를 관리할 수 있는 API 인터페이스를 제공합니다 AWS 계정. 승인된 판매자의 경우의 셀프 서비스 게시 기능을 포함하여 제품을 프로그래밍 방식으로 관리할 수 있습니다 [AWS Marketplace Management Portal](#). 프라이빗 마켓플레이스 관리자의 경우 프라이빗 마켓플레이스를 프로그래밍 방식으로 관리할 수 있습니다.

카탈로그 API 작업을 사용하면 기존 제품을 프로그래밍 방식으로 보고 업데이트할 수 있습니다. 를 제품 빌드 또는 배포 파이프라인 AWS Marketplace Catalog API 과 통합하여 AWS Marketplace 제품 업데이트 프로세스를 자동화할 수 있습니다. 카탈로그 API를 기반으로 자체 애플리케이션을 생성하여 제

품을 관리할 수도 있습니다 AWS Marketplace. AWS 계정 또는 AWS 조직의 사용자가 프라이빗 마켓플레이스를 통해 보고 구매할 수 있는 제품을 관리할 수 있습니다.

AWS Marketplace Catalog API 서비스는 표준 AWS API 기능을 제공합니다. 작업에 설명된 REST API 작업을 직접 사용하거나 AWS SDK를 사용하여 사용 중인 프로그래밍 언어 또는 플랫폼에 맞는 API에 액세스할 https://docs.aws.amazon.com/marketplace-catalog/latest/api-reference/API_Operations.html 수 있습니다. AWS 애플리케이션 개발에 대한 자세한 내용은 [시작하기를 참조하세요 AWS](#). AWS SDKs 사용에 대한 자세한 내용은 [AWS SDKs](#).

카탈로그 API 엔터티

AWS Marketplace 엔터티는 제품 또는 제안과 같이 다양한 비즈니스 목적에 맞는 데이터 컨테이너입니다. 개체는 유형별로 분류됩니다. 각 엔터티 유형은 특정 비즈니스 도메인(예: 제품 또는 판매자 계정)과 관련된 데이터를 캡슐화합니다.

이 패러다임을 단순화하기 위해 개체는 구조에서 일정 수준의 공통성을 갖도록 설계되었습니다. 따라서 새 비즈니스 도메인을 도입하더라도 완전히 새로운 구조를 배울 필요가 없습니다.

일반 구조

개체의 일반적인 구조는 다음과 같습니다.

- 버전이 있는 명명된 유형
- 유형의 특정 인스턴스에 대한 식별자입니다.
- 개체의 속성을 포함하는 하나 이상의 패킷

유형 버전 관리

이름이 지정된 모든 유형에는 유형과 버전이 연결되어 있습니다. 예: *Entity*Product@1.0. 유형(### 제품)은 콘텐츠의 분류를 나타냅니다. 버전(1.0)은 *Entity* Product의 구조를 나타냅니다.

버전은 개체의 구조에 대한 세부 정보를 제공합니다. 다음은 버전이 변경되는 시기를 설명합니다.

- 기존 엔터티는 버전을 변경하지 않고 재구성되지 않습니다. 선택 사항인 새 필드를 추가하면 마이너 버전이 업데이트됩니다.
- 유형의 구조를 근본적으로 변경하는 모든 기능은 메이저 버전 업데이트로 이어집니다. 그러한 예는 다음과 같습니다.
 - 필드 제거
 - 필드 이름 바꾸기(동일한 의미 체계의 다른 이름)

- 기존 필드의 의미 변경(예: 예상 유형 변경)
- 메이저 버전 업데이트는 이전 버전의 패치 하위 집합을 유지할 수 있습니다.
- 사용자에게 새 버전에 대한 알림 및 설명서가 제공됩니다.

식별자

각 개체는 비즈니스 도메인 내의 고유한 사물을 나타냅니다. 고유한 사물을 식별하기 위해 **prod-ad8EXAMPLE651@3**과 RevisionId같은 EntityId와를 연결하는 식별자를 사용합니다. 이 예제에서 EntityId는 prod-ad8EXAMPLE651 이고 RevisionId입니다3. 엔터티에 대한 모든 성공적인 변경 요청은 개정을 업데이트합니다.

다음은 식별자에 대한 중요한 세부 정보입니다.

- 각 엔터티는 한 엔터티와 다른 엔터티를 전역적으로 구분하는 키EntityId인 로 고유하게 식별됩니다.
- 개체의 게시된 각 개정에는 RevisionId. 는와 RevisionId함께 게시된 개정을 다른 개정과 EntityId구분합니다.
- AWS Marketplace 는 EntityId및 RevisionId를 생성합니다.

DescribeEntity 작업을 사용하여 세부 정보와 가장 최근의 식별자를 찾을 수 있습니다revisionId.

RevisionId는에 대한 요청의 선택적 부분입니다StartChangeSet(참조[변경 세트 작업](#)). 를 포함하면 RevisionId가 엔터티의 최신 개정RevisionId이 아닌 ValidationException 경우에 대한 요청이 StartChangeSet 실패합니다. 이를 통해 애플리케이션에서 낙관적 잠금을 구현할 수 있습니다.

Note

최신 개정RevisionId이 아닌를 포함하면 ValidationException 메시지에 최신이 포함됩니다RevisionId.

를 생략하면 엔터티의 최신 개정에서 RevisionId요청이 자동으로 수행됩니다.

Warning

두 개의 동일한 객체 변경 요청으로 인해 한 요청이 다른 요청의 변경 사항을 덮어쓸 수 있습니다. 두 번째 요청은 첫 번째 요청에 의해 변경된 데이터를 다시 쓰기 때문입니다. 요청에

RevisionId를 사용하면 이전 개정을 변경하여 현재 개정을 덮어쓰도록 허용하지 않아이 문제가 방지됩니다.

패킷

패킷은 속성의 논리적 그룹입니다. 엔터티에는 일반적으로 엔터티의 다양한 측면을 나타내는 여러 패킷이 포함됩니다. 패킷 내의 속성에는 다음과 같은 속성이 있습니다.

- 각 속성에는 해당 속성이 속한 컨테이너의 범위 내에 고유한 이름이 있습니다.
- 속성은 간단한 유형(문자열, 정수 또는 부동 소수)일 수 있습니다.
- 속성은 복잡한 유형(컨테이너/구조 또는 배열)일 수 있습니다.

엔터티 유형

개체 유형은 개체가 나타내는 내용을 정의합니다. 엔터티는 AWS Marketplace 또는 프라이빗 마켓플레이스의 판매자 제품일 수 있습니다. 자세한 내용은 [판매자 제품 작업](#) 및 [프라이빗 마켓플레이스 작업을 참조하세요](#).

변경 세트 작업

카탈로그 API를 사용하는 경우 엔터티를 통해 요청이 생성 및 업데이트되고 변경 요청을 사용하여 완료됩니다. 모든 변경에는 변경할 엔터티, 수행할 변경 유형 및 변경 세부 정보가 지정됩니다. 수행할 변경 유형을 라고 합니다ChangeType. 의 모음ChangeType을 라고 합니다ChangeSet.

변경 세트로 작업할 수 있는 네 가지 작업이 있습니다.

- StartChangeSet - 변경 세트를 요청합니다. 변경 사항이 대기열에 추가되고 처리됩니다. 자세한 내용은 [판매자 제품 작업](#) 및 [프라이빗 마켓플레이스 작업을 참조하세요](#).
- DescribeChangeSet - 요청 상태를 포함하여 변경 사항 세트의 세부 정보를 가져옵니다. 상태는 다음과 같습니다.
 - PREPARING - 변경 사항을 적용할 준비를 합니다.
 - APPLYING - 요청된 변경을 수행하는 중입니다.
 - SUCCEEDED - 요청이 성공적으로 완료되었습니다.
 - CANCELLED - 사용자가 요청을 취소했습니다.
 - FAILED - 요청이 성공적으로 완료되지 않았습니다. 자세한 내용은 응답에서 확인할 수 있습니다.

- ListChangeSets - 현재 진행 중인 변경 세트 목록을 가져옵니다.
- CancelChangeSet - 변경 세트 취소를 요청합니다. 변경 사항은 PREPARING 상태인 동안에만 취소할 수 있습니다.

일반적인 워크플로는 로 변경을 요청한 다음 StartChangeSet 반환된를 사용하여 변경이 완료될 때까지 DescribeChangeSet 작업을 ChangeSetId 폴링하는 것입니다.

다음은 DescribeChangeSet 응답의 예입니다.

```
{
  "ChangeSet": [
    {
      "ChangeName": "myChangeName",
      "ChangeType": "UpdateInformation",
      "Details": "{ \"ProductTitle\": \"My Product Title\", \"ShortDescription\": \"My product short description.\", \"LongDescription\": \"My product longer description.\", \"Sku\": \"123example456\", \"SupportDescription\": \"Need help? Contact our experts at support@example.com\\n\\nYour purchase includes 24x7 support.\", \"Categories\": [ \"Operating Systems\", \"Network Infrastructure\", \"Application Development\" ] }",
      "DetailsDocument": {
        "ProductTitle": "My Product Title",
        "ShortDescription": "My product short description.",
        "LongDescription": "My product longer description.",
        "Sku": "123example456",
        "SupportDescription": "Need help? Contact our experts at support@example.com\\n\\nYour purchase includes 24x7 support.",
        "Categories": [
          "Operating Systems",
          "Network Infrastructure",
          "Application Development"
        ]
      },
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "AmiProduct@1.0"
      },
      "ErrorDetailList":
```

```

    []
  }
],
"ChangeSetArn": "arn:aws:aws-marketplace:[exampleARN]",
"ChangeSetId": "example123456789012abcdef",
"ChangeSetName": "myChangeSetName",
"EndTime": "2023-03-03T00:00:00Z",
"FailureCode": null,
"FailureDescription": null,
"StartTime": "2023-03-02T00:00:00Z",
"Status": "SUCCEEDED"
}

```

Note

프로그래밍 방식으로 변경 세트를 폴링하거나 작업할 때는 서비스 제한을 준수해야 합니다. 자세한 내용은 [AWS Marketplace 카탈로그 API의 서비스 할당량](#) 단원을 참조하십시오.

변경이 완료되면 `ListEntities`를 사용하여 생성하거나 수정한 엔터티(및 연결된)를 찾을 수 있습니다. `EntityID`. 그런 다음 `DescribeEntity` 함께 `EntityID`를 사용하여 세부 정보를 가져올 수 있습니다.

콘솔에서 판매자를 위한 변경 요청 작업에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [변경 요청 생성](#)을 참조하세요.

여러 변경 요청을 동시에 수행

단일 변경 세트 내에서 모든 변경 유형을 번들링하고 함께 실행할 수 있습니다. 카탈로그 API는 최상의 성능을 제공하기 위해 동시에 여러 번 변경하도록 구축되었습니다. 판매자와 채널 파트너는 여러 번들로 `ChangeTypes` 번들링된 변경 사항을 호출할 수 있습니다 `ChangeSet`. 동일한 단일 또는 다른 엔터티에 대해 여러 변경 사항을 호출할 수 있습니다 `ChangeSet`. 카탈로그 API는 변경 사항을 적용해야 하는 순서를 평가하고 이러한 변경을 수행합니다.

그러나 요청이 별도의 변경 세트로 이루어진 경우는 동일한 제품에서 충돌하는 변경 요청을 시작할 수 없습니다. 이러한 경우는 `ResourceInUseException` 오류를 AWS Marketplace 반환합니다.

- AMI 및 컨테이너 제품을 수정하는 경우 대부분의 변경은 오류 없이 수행할 수 있습니다. 단, 다음과 같은 경우는 예외입니다.
- 동일한 제품 `ChangeType`에서 두 개의 요청이 동일한 경우 두 번째 요청은 오류를 반환합니다.

- 한 요청이 버전 정보를 업데이트하는 것이고 다른 요청은 버전을 제한하거나 추가하는 경우 두 번째 요청은 오류를 반환합니다.
- 요청이 인 경우 동일한 제품에 대해 PREPARING 다른 요청을 할 수 있습니다. 그러나 현재 로 변경하면 다른 요청이 차단되어 오류가 반환될 APPLYING 수 있습니다.
- 다른 제품 유형 및 프라이빗 마켓플레이스의 경우 한 번에 제품에 대한 단일 요청만 할 수 있습니다. 첫 번째 요청이 진행되는 동안 동일한 제품을 업데이트하기 위한 다른 요청이 이루어진 경우 두 번째 요청은 오류를 반환합니다.
- 판매자 운영 팀에 보류 AWS Marketplace 중인 제품에 대한 요청이 있는 경우 해당 제품에 대한 다른 요청은 오류를 반환합니다.

변경 요청에 대한 ResourceInUseException 오류가 발생하면 나중에 요청을 다시 시도할 수 있습니다. 진행 중인 요청의 상태에 따라 첫 번째 요청을 취소하여 다시 제출된 두 번째 요청을 더 빨리 완료할 수 있습니다.

하나의 변경 세트에서 여러 변경 유형 호출

카탈로그 API를 사용하여 하나 이상의 서로 다른 엔터티를 대상으로 하는 StartChangeSet 요청 하나에 최대 20개의 변경 사항을 결합하고 연결할 수 있습니다.

일반적인 사용 사례는 SaaSProduct@1.0 초안 제품, Offer@1.0 초안 제안을 생성하고 제품 및 제안의 메타데이터 정보를 작성하는 것입니다. 이 작업은 다음과 같은 네 가지 변경 유형을 하나의 변경 세트에 포함시켜 수행됩니다.

- SaaSProduct@1.0의 CreateProduct

ChangeName 파라미터를 지정합니다. 그런 다음이 변경 유형으로 생성된 제품을 후속 변경에 의해 설정된 동일한 변경 세트에서 참조할 수 있습니다.

예를 들어 CreateProductChange입니다.

- UpdateInformation 동일한 변경 세트에서 SaaSProduct@1.0 생성된의

Entity.Identifier 필드에서 다음 형식의 CreateProduct 변경 이름을 사용하여 변경 유형으로 생성된 제품을 참조할 수 있습니다.

`${ChangeName}.Entity.Identifier`

예를 들어 `$CreateProductChange.Entity.Identifier`입니다.

- CreateOffer 동일한 변경 세트에서 SaaSProduct@1.0 생성된에 Offer@1.0 연결된의

ChangeName 파라미터를 지정합니다. 그런 다음이 변경 유형으로 생성된 제품을 후속 변경에 의해 설정된 동일한 변경 세트에서 참조할 수 있습니다. 예를 들어 CreateOfferChange입니다.

CreateOffer 변경 유형의 페이로드에 있는 ProductId 파라미터의 경우 `${ChangeName}.Entity.Identifier` 구문을 사용하여 CreateProduct 변경 유형으로 생성된 SaaS 제품을 참조할 수도 있습니다.

예를 들어 `{"ProductId": "$CreateProductChange.Entity.Identifier"}`입니다.

- UpdateInformation 동일한 변경 세트에서 Offer@1.0 생성된의

Entity.Identifier 필드에서 다음 형식의 CreateOffer 변경 이름을 사용하여 변경 유형으로 생성된 제안을 참조할 수 있습니다.

`${ChangeName}.Entity.Identifier`

예를 들어 `$CreateOfferChange.Entity.Identifier`입니다.

다음은 결합된 변경 세트의 예입니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "ChangeName": "CreateProductChange",
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "ChangeName": "UpdateProductInformationChange",
      "DetailsDocument": {
```

```

    "ProductTitle": "My Product Title",
    "ShortDescription": "My product short description.",
    "LongDescription": "My product longer description.",
    "Sku": "123example456",
    "LogoUrl": "https://s3.amazonaws.com/presigned-or-public-url-to-logo-stored-in-
s3",
    "VideoUrls": [
        "https://example.com"
    ],
    "Highlights": [
        "123example45"
    ],
    "AdditionalResources": "123example456",
    "SupportDescription": "Need help? Contact our experts at support@example.com \n
\nYour purchase includes 24x7 support.",
    "Categories": [
        "Operating Systems",
        "Network Infrastructure",
        "Application Development"
    ],
    "SearchKeywords": [
        "123example45"
    ],
  },
  {
    "ChangeType": "CreateOffer",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "ChangeName": "CreateOfferChange",
    "DetailsDocument": {
      "ProductId": "$CreateProductChange.Entity.Identifier"
    }
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Name": "Offer created together with SaaSProduct",

```

```

    "Description": "Test offer created together with SaaSProduct in the same
    Catalog API change set"
  }
}
]
}

```

Details 속성 작업(레거시)

Note

이 섹션에서는 변경 요청의 레거시 Details 속성에 대해 설명합니다. 여기에는 변경 세부 정보에 대한 추가 형식 지정이 필요합니다. 대체 DetailsDocument 속성을 사용하는 것이 좋습니다. 추가 형식이 필요하지 않으며 변경 세부 정보를 변경할 필요가 없습니다. DetailsDocument 속성의 예는 [판매자 제품 작업](#) 및 [프라이빗 마켓플레이스 작업을 참조하세요](#).

StartChangeSet 작업의 Details 속성은 문자열 값입니다. 콘텐츠는 JSON 객체입니다. JSON 객체를 문자열 속성에 배치하려면 모든 JSON 제어 문자를 이스케이프하고 줄 바꿈을 제거하여 객체를 한 줄 문자열로 변환해야 합니다.

예를 들어에서 StartChangeSet 작업을 사용하여 프라이빗 마켓플레이스에 있는 사용자의 요청을 비활성화UpdateProcurementPolicy하는 경우 다음과 같이 요청합니다.

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateProcurementPolicy",
      "Details": "<string>",
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier" : "exp-1234example@5"
      }
    }
  ]
}

```

이 경우 Details 속성에 사용하는 JSON 객체는 다음과 같습니다(문자열로 변환하기 전).

```
{
  "Configuration": {
    "PolicyResourceRequests": "Deny"
  }
}
```

하지만 Details 속성에는 JSON이 아닌 문자열이 필요합니다. 이 JSON 객체를 한 줄 문자열로 변환한 후 다음과 같습니다.

```
"{\"Configuration\" : {\"PolicyResourceRequests\" : \"Deny\"}}"
```

이 문자열을 사용하면 다음과 같이 전체 변경 세트 요청을 생성할 수 있습니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateProcurementPolicy",
      "Details": "{\"Configuration\" : {\"PolicyResourceRequests\" : \"Deny\"}}",
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier" : "exp-1234example@5"
      }
    }
  ]
}
```

일반적으로 이 API 참조의 예제는 이미 문자열로 변환된 JSON 객체를 보여줍니다. 경우에 따라 이해를 높이기 위해 새 줄이 있는 더 복잡한 샘플이 포함됩니다.

JSON을 문자열로 자동 변환

경량 명령줄 JSON 프로세서인 [jq](#)와 같은 도구를 사용하여 JSON 객체를 문자열로 변환할 수 있습니다. 다음 예제에서는 jq를 사용하여 JSON 객체를 Details 속성에 사용할 수 있는 문자열로 변환하는 방법을 보여줍니다.

```
DETAILS_JSON='{
```

```
"ProductTitle": "My Product Title",
"ShortDescription": "My product short description.",
"LongDescription": "My product long description."
}';

DETAILS_JSON_STRING="$(echo "${DETAILS_JSON}" | jq 'tostring');";
```

를 에코하면 JSON "\${DETAILS_JSON_STRING}"이 제대로 이스케이프된 다음 문자열이 결과가 됩니다. {"ProductTitle": "My Product", "ShortDescription": "My product short description.", "LongDescription": "My product long description."}

DescribeEntity를 사용하여 엔터티에 대한 정보 가져오기

카탈로그 API를 통해 제품 및 프라이빗 마켓플레이스를 포함한 기존 엔터티에 대한 정보를 프로그래밍 방식으로 가져올 수 있습니다.

ListEntities 작업은 개체 목록을 반환합니다. 그런 다음 DescribeEntity 작업을 사용하여 개별 개체에 대한 세부 정보를 가져올 수 있습니다. 이는 예를 들어 판매하는 제품을 카탈로그화하는 데 직접 유용할 수 있습니다. 업데이트하려는 부분만 업데이트하기 전에 개체의 현재 상태를 가져올 수 있으므로 개체를 업데이트할 때도 유용할 수 있습니다.

다음 예제에서는 ListEntities를 사용하여 컨테이너 제품 목록을 가져온 다음 DescribeEntity를 사용하여 특정 제품 중 하나에 대한 정보를 가져오는 방법을 보여줍니다.

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "ContainerProduct"
}
```

엔터티 유형의 경우 버전 없이 엔터티 유형을 사용해야 합니다. 해당 유형의 모든 엔터티를 반환합니다 (버전을 기준으로 필터링하지 않음).

다음은 ListEntities 작업에 대한 응답의 샘플입니다.

```
{
  "EntitySummaryList": [
    {
      "Name": "Container Product 1",
      "EntityType": "ContainerProduct",
```

```

      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12",
      "EntityArn": "arn:aws:aws-marketplace:[exampleARN]",
      "LastModifiedDate": "2021-03-01T00:00:00Z",
      "Visibility": "Public"
    },
    {
      "Name": "Container Product 2",
      "EntityType": "ContainerProduct",
      "EntityId": "example2-abcd-1234-5ef6-7890abcdef12",
      "EntityArn": "arn:aws:aws-marketplace:[exampleARN]",
      "LastModifiedDate": "2021-03-02T00:00:00Z",
      "Visibility": "Public"
    }
  ],
  "NextToken": "exampleabcdef12345..."
}
```

이러한 제품 중 하나에 대한 세부 정보를 가져오려면 DescribeEntity 작업을 사용합니다. 다음 예제에서는 위에서 반환된 첫 번째 제품에 대한 세부 정보를 가져오는 방법을 보여줍니다.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=example1-abcd-1234-5ef6-7890abcdef12 HTTP/1.1
```

다음은에 대한 응답을 보여줍니다DescribeEntity.

```

{
  "EntityType": "ContainerProduct@1.0",
  "EntityIdentifier": "example1-abcd-1234-5ef6-7890abcdef12@9",
  "EntityArn": "arn:aws:aws-marketplace:[exampleARN]",
  "LastModifiedDate": "2021-03-02T20:19:14Z",
  "Details": "{\n  \"Versions\": [\n    {\n      \"Id\": \"example2-0000-aaaa-5ef6-7890abcdef12\",
      \"ReleaseNotes\": \"My release notes\",
      \"UpgradeInstructions\": \"N/A\",
      \"VersionTitle\": \"1.0\",
      \"CreationDate\": \"2021-03-02T00:00:00.000Z\",
      \"Sources\": [\n        {\n          \"Type\": \"DockerImages\",
          \"Id\": \"example3-1111-bbbb-5ef6-7890abcdef12\",
          \"Images\": [\n            \"111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-repo-1:some-tag\",
            \"Compatibility\": {\n              \"Platform\": \"Linux\"
            },
            \"DeliveryOptions\": [\n              {\n                \"Id\": \"example4-2222-cccc-2222-cccccccccccc\",
                \"Type\": \"ElasticContainerRegistry\",
                \"SourceId\": \"example3-1111-bbbb-5ef6-7890abcdef12\",
                \"Title\": \"New delivery option 1\",
                \"ShortDescription\": \"Delivery option 1\",
                \"isRecommended\": false,
                \"Compatibility\": {\n                  \"AWSservices\": [\n                    \"ECS\",
                    \"EKS\"
                  ],
                  \"Instructions\": {\n                    \"Usage\": \"test\"
                  },
                  \"Recommendations\": {\n                    \"AdditionalArtifacts\": []
                  },
                  \"Visibility\": \"Limited\"
                }
              ]
            },
            \"Description\": {\n              \"Highlights\": [\"Some highlight\"],
              \"LongDescription\": \"Description of my product\",
              \"ProductCode\": \"123456789012abcdef1234567\"
            }
          }
        ]
      }
    ]
  }
}
```

```

\ "Manufacturer\ ": null, \ "Visibility\ ": \ "Limited\ ", \ "AssociatedProducts\ ": null, \ "Sku
\ ": null, \ "SearchKeywords\ ": [ \ "some keyword\ " ], \ "ProductTitle\ ": \ "Container Product 1\ ",
\ "ShortDescription\ ": \ "Description of my product\ ", \ "Categories\ ": [ \ "Operating Systems
\ " ] }, \ "PromotionalResources\ ": { \ "LogoUrl\ ": \ "https://awsmp-logos.s3.amazonaws.com/
PLACEHOLDER_Logo_for_Containers_products.png\ ", \ "AdditionalResources\ ": [ ], \ "Videos
\ ": [ ] }, \ "SupportInformation\ ": { \ "Description\ ": \ "Description of support information.
\ ", \ "Resources\ ": [ ] }, \ "RegionAvailability\ ": { \ "Regions\ ": [ \ "ap-south-1\ ", \ "eu-
west-3\ ", \ "eu-north-1\ ", \ "eu-west-2\ ", \ "eu-west-1\ ", \ "ap-northeast-2\ ", \ "ap-
northeast-1\ ", \ "me-south-1\ ", \ "ca-central-1\ ", \ "sa-east-1\ ", \ "ap-east-1\ ", \ "ap-
southeast-1\ ", \ "ap-southeast-2\ ", \ "eu-central-1\ ", \ "us-east-1\ ", \ "us-east-2\ ", \ "us-
west-1\ ", \ "us-west-2\ " ] }, \ "FutureRegionSupport\ ": null }, \ "Repositories\ ": [ { \ "Url\ ":
\ "111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-repo-1\ ", \ "Type\ ":
\ "ECR\ " } ] },
  "DetailsDocument":
  {
    "Versions":
    [
      {
        "Id": "example2-0000-aaaa-5ef6-7890abcdef12",
        "ReleaseNotes": "My release notes",
        "UpgradeInstructions": "N/A",
        "VersionTitle": "1.0",
        "CreationDate": "2021-03-02T00:00:00.000Z",
        "Sources":
        [
          {
            "Type": "DockerImages",
            "Id": "example3-1111-bbbb-5ef6-7890abcdef12",
            "Images":
            [
              "111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-
repo-1:some-tag"
            ],
            "Compatibility":
            {
              "Platform": "Linux"
            }
          }
        ],
        "DeliveryOptions":
        [
          {
            "Id": "example4-2222-cccc-2222-cccccccccccc",
            "Type": "ElasticContainerRegistry",

```

```
    "SourceId": "example3-1111-bbbb-5ef6-7890abcdef12",
    "Title": "New delivery option 1",
    "ShortDescription": "Delivery option 1",
    "isRecommended": false,
    "Compatibility":
    {
        "AWServices":
        [
            "ECS",
            "EKS"
        ]
    },
    "Instructions":
    {
        "Usage": "test"
    },
    "Recommendations":
    {
        "AdditionalArtifacts":
        []
    },
    "Visibility": "Limited"
}
]
}
],
"Description":
{
    "Highlights":
    [
        "Some highlight"
    ],
    "LongDescription": "Description of my product",
    "ProductCode": "123456789012abcdef1234567",
    "Manufacturer": null,
    "Visibility": "Limited",
    "AssociatedProducts": null,
    "Sku": null,
    "SearchKeywords":
    [
        "some keyword"
    ],
    "ProductTitle": "Container Product 1",
    "ShortDescription": "Description of my product",
```



```
    "Categories":
    [
        "Operating Systems"
    ]
},
"PromotionalResources":
{
    "LogoUrl": "https://awsmp-logos.s3.amazonaws.com/
PLACEHOLDER_Logo_for_Containers_products.png",
    "AdditionalResources":
    [],
    "Videos":
    []
},
"SupportInformation":
{
    "Description": "Description of support information.",
    "Resources":
    []
},
"RegionAvailability":
{
    "Regions":
    [
        "ap-south-1",
        "eu-west-3",
        "eu-north-1",
        "eu-west-2",
        "eu-west-1",
        "ap-northeast-2",
        "ap-northeast-1",
        "me-south-1",
        "ca-central-1",
        "sa-east-1",
        "ap-east-1",
        "ap-southeast-1",
        "ap-southeast-2",
        "eu-central-1",
        "us-east-1",
        "us-east-2",
        "us-west-1",
        "us-west-2"
    ],
    "FutureRegionSupport": null
}
```

```

    },
    "Repositories":
    [
      {
        "Url": "111122223333.dkr.ecr.us-east-1.amazonaws.com/some-seller-prefix/my-repo-1",
        "Type": "ECR"
      }
    ]
  }
}

```

Note

DetailsDocument 속성에는 개체 세부 정보가 JSON 객체로 포함됩니다. 레거시 Details 속성에는 문자열과 동일한 JSON 객체가 포함됩니다.

AWS Marketplace 계약 API 사용

AWS Marketplace 는 고객이 솔루션을 구축하고 비즈니스를 운영하기 위해 타사 소프트웨어, 데이터 및 서비스를 찾고, 구매하고, 배포하고, 관리하는 데 사용할 수 있는 엄선된 디지털 카탈로그입니다. AWS Marketplace 계약 서비스는 구매자와 판매자가 계약 나열, 검색 및 필터링을 포함하여 계약을 관리하는 데 도움이 AWS Marketplace 되는 API 인터페이스를 제공합니다.

주요 개념

다음 표에서는 AWS Marketplace 계약 API에 사용되는 주요 개념을 설명합니다.

개념	설명
수락자	계약에서 제안자가 연장한 약관의 전부 또는 일부를 수락하는 당사자입니다. 대부분의 일반적인 사용 사례에서 수락자는 제품의 구매자입니다.
계약	제안자(일반적으로 판매자/ISV 또는 채널 파트너)와 수락자(일반적으로 구매자)를 포함하여 두 당사자를 바인딩하고 이들 간에 적용되는 이용 약관을 정의하는 문서입니다.
계약 취소 요청	판매자(제안자)가 활성 계약을 취소하기 위해 시작한 요청입니다. 구매자(수락자)는 취소 요청을 수락하거나 거부할 수 있습니다. 수락하면 계약 취

개념	설명
	소 워크플로가 비동기적으로 실행되고 계약 상태가 취소됨으로 변경됩니다. 거부된 경우 계약은 활성 상태로 유지되며 판매자는 새 취소 요청을 제출할 수 있습니다.
계약 제안	판매자/제안자가 제안한 용어/제안 세트를 참조하는 엔터티입니다.
계약 요청	고객 작업의 결과로 생성된 리소스를 참조하는 엔터티로, 요청 수락 시 계약 생성/수정 파라미터, 요청 수명 주기 및 예상 결과를 간략하게 설명합니다.
청구 가능한 차원	제품 가격 책정 방식과 관련된 요소. 예를 들어 Amazon Machine Image(AMI) 제품의 경우 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스, 서비스형 소프트웨어(SaaS) 제품의 경우 읽기 전용 사용자와 비교한 프리미엄 인스턴스입니다.
채널 파트너(CP)	제품 소유자(ISV)가 대신 제품을 재판매할 권한을 부여한 조직입니다 AWS Marketplace.
독립 소프트웨어 공급 업체(ISVs)	자체 제품을 나열하고 관리하는 판매자 AWS Marketplace.
의도	의도는 계약을 생성하는 동안 구매자가 갖는 동기를 나타냅니다. 현재 지원되는 의도는 계약 생성 흐름 중에 신규, 수정 및 교체입니다.
결제 요청	계약과 관련된 결제에 대해 판매자(제안자)가 시작한 요청입니다. 구매자(수락자)는 결제 요청을 수락하거나 거부할 수 있습니다.
제안자	가장 일반적으로 제안을 사용하여 계약의 초기 조건 집합을 연장하는 당사자입니다.
구매 주문	조직은 구매자에게 구매 주문을 사용하여 승인된 공급업체로부터 구매하고, 구매 대상을 추적하고, 지출이 예측되고, 예산이 책정되고, 승인되고, 책임 있는 비용 센터(들)에 할당되도록 해야 합니다.

개념	설명
Resource	판매자가 판매하려는 단위 또는 리소스로 AWS Marketplace, 종종 기본 제품이라고 합니다. 기본 제품은 제품 정보, 배포 속성 및 결제 정보가 추가될 때까지 구매자 소비를 위해 완료되지 않습니다. 제품은 판매자가 판매하려는 목록의 제품 정보, 소프트웨어 배포 속성 및 결제 메커니즘을 설명합니다. 구매자가 판매하고 사용할 수 있는 거래 가능한 단위가 되려면 제품을 제안과 페어링해야 합니다 AWS Marketplace.
Term	계약 적용 방식(예: 요금)을 관리하는 계약 내 단위입니다.

엔드포인트

다음 엔드포인트를 사용하여 AWS Marketplace 계약 서비스에 액세스할 수 있습니다.

엔드포인트	리전	지원되는 IP 프로토콜
agreement-marketplace.us-east-1.api.aws	미국 동부(버지니아 북부)	IPv4, IPv6

권한

계약 API는 표준 AWS 서명 버전 4(SigV4) 인증을 사용합니다. API를 호출하려면 유효한 AWS 자격 증명과 적절한 IAM 권한이 있어야 합니다. 자세한 내용은 [AWS Marketplace 계약 API에 대한 액세스 제한](#)을 참조하세요.

오류 코드

다음 오류 코드는 [AWS Marketplace 계약 서비스 작업에](#) 적용됩니다. 여러 사용 사례에 대해 단일 오류 코드가 반환될 수 있습니다. 각 오류 코드가 발생하는 특정 시나리오를 식별하려면 아래 설명을 참조하세요. 이 목록은 전체 목록이 아니며 새 기능이 릴리스될 때 업데이트될 수 있습니다.

ValidationException

오류 코드	설명
INVALID_SOURCE_AGREEMENT_IDENTIFIER	소스 계약 식별자가 예상 형식과 일치하지 않거나 소스 계약 식별자가 필요하지 않은 의도에 대해 제공된 경우(예: 신규) 반환됩니다.
MISSING_SOURCE_AGREEMENT_IDENTIFIER	소스 계약 식별자가 필요한 의도(예: REPLACE 또는 AMEND)에 대해 제공되지 않은 경우 반환됩니다.
INVALID_REQUESTED_TERM_CONFIGURATION	요청된 용어 구성이 유효하지 않을 때 반환됩니다. 가능한 시나리오: • 구성은 수락하지 않는 용어에 대해 제공됩니다. • 필수 구성이 누락되었습니다. • 선택기 값은 계약 제안에 속하지 않습니다. • 여러 차원 선택이 허용되지 않는 경우 차원에는 하나의 차원만 포함될 수 있습니다. • 차원 키는 계약 제안 또는 수락된 조건에서 인식되지 않습니다. • 차원에 수량 구성이 활성화되지 않은 경우 차원 값은 1일 수 있습니다. • 구성에는 중복 차원 키가 포함되어 있습니다. • 구성은 소스 계약과 다르지 않습니다.
INVALID_AGREEMENT_PROPOSAL_IDENTIFIER	계약 제안 식별자가 예상 형식과 일치하지 않거나 계약 제안 식별자가 필요하지 않은 의도(예: AMEND)에 대해 제공된 경우 반환됩니다.
MISSING_AGREEMENT_PROPOSAL_IDENTIFIER	계약 제안 식별자가 필요한 의도(예: NEW 또는 REPLACE)에 대해 제공되지 않은 경우 반환됩니다.
INVALID_FILTER_NAME	제공된 필터 이름이 인식된 필터가 아니거나 요청에 중복 필터 이름이 포함된 경우에 반환됩니다.

오류 코드	설명
INVALID_FILTER_VALUES	필터 값이 유효하지 않거나, 중복되거나, 비어 있거나, 허용되는 최대 개수를 초과하거나, 날짜 범위 필터가 BeforeEndTime 값 앞에 값을 지정할 때 반환됩니다AfterEndTime .
INVALID_NEXT_TOKEN	페이지 매김 토큰의 형식이 잘못되었거나, 만료되었거나, 해독할 수 없거나, 페이지 매김된 호출 간에 요청 파라미터가 변경된 경우에 반환됩니다.
DUPLICATE_CHARGES	요청에 중복된 요금 식별자가 포함되어 있거나 동일한 요금 식별자에 대해 다른 구매 주문 값이 제공되는 경우 반환됩니다.
UNSUPPORTED_FILTERS	제공된 필터 조합이 지원되지 않을 때 반환됩니다. 지원되는 필터 조합은 당사자 유형에 따라 다릅니다. 퍼블릭 설명서에서 지원되는 조합을 확인합니다.
INVALID_PURCHASE_ORDER_REFERENCE	구매 주문 참조가 형식 요구 사항을 충족하지 않거나 계정 관리자가 구성한 대로 필수 구매 주문이 제공되지 않은 경우 반환됩니다.
INVALID_CHARGE_AMOUNT	청구 금액이 유효하지 않을 때 반환됩니다. 가능한 시나리오: • 청구 금액이 해당 기간의 최대 총 청구 금액을 초과합니다. • 요금 금액 형식이 통화 요구 사항(예: 소수 자릿수가 너무 많음)과 일치하지 않습니다. • 계약이 최대 총 청구 금액에 도달했기 때문에 남은 금액이 없습니다.

오류 코드	설명
UNSUPPORTED_ACTION	요청된 작업이 리소스의 현재 상태에 대해 지원되지 않을 때 반환됩니다. 특정 시나리오는 API에 따라 다릅니다. • <code>UpdatePurchaseOrders</code> - 청구 날짜가 과거이고 자동 갱신이 비활성화되었거나, 청구 날짜가 3시간 이내이거나, 청구가 환불 요금이거나, 청구 수정이 최신이 아니거나, 구매 주문이 모두 동일한 계약에 속하지 않거나, 계약이 활성 상태가 아닙니다. • <code>CreateAgreementRequest</code> - 장래 날짜 계약 수정이 지원되지 않거나, 리소스 유형 및 기간 유형 조합이 요청된 의도를 지원하지 않거나, 활성 계약을 장래 날짜 계약으로 대체할 수 없거나, 수정 중에 계약 기간을 변경할 수 없거나, 현재 계약을 업데이트할 수 없거나(예: 보류 중인 권한 있음), 계약 제안이 소스 계약과 동일한 리소스를 참조하거나, 계약 제안이 소스 계약과 달라야 하거나, 계약 유형이이 작업을 지원하지 않습니다. • <code>AcceptAgreementRequest</code> - 리소스 유형 및 기간 유형 조합이이 작업을 지원하지 않거나, 계약 유형이이 작업을 지원하지 않거나, 활성 계약이 이미 동일한 리소스에 있거나, 현재 계약을 업데이트할 수 없습니다. • <code>CancelAgreement</code> - 계약에 보류 중인 권한이 있거나, 계약 유형이 취소를 지원하지 않거나, 와의 계약이 동일한 리소스에서 활성 상태 <code>ConfigurableUpfrontPricingTerm</code> <code>PaymentScheduleTerm</code> 인 동안 계약을 취소할 수 없거나, 리소스 유형과 기간 유형 조합이 취소를 지원하지 않습니다. • <code>GetBillingAdjustmentRequest</code> - 계약이 비구매 계약 유형이거나 호출자가 계약의 수락자입니다. • <code>SendAgreementCancellationRequest</code> - 계약 유형은이 작업을 지원하지 않습니다. • <code>AcceptAgreementCancellationRequest</code> - 계약 유형은이 작업을 지원하지 않습니다. • <code>RejectAgreementCancellationRequest</code> - 계약 유형은이 작업을 지원하지 않습니다.

오류 코드	설명
	• <code>CancelAgreementCancellationRequest</code> - 계약 유형은 이 작업을 지원하지 않습니다. • <code>GetAgreementCancellationRequest</code> - 계약은 비구매 계약 유형입니다.

ResourceNotFoundException

에는 찾을 수 없는 리소스를 식별하는 `resourceType` 및 `resourceId` 필드가 `ResourceNotFoundException` 포함되어 있습니다. 다음 표에서는 각 리소스 유형에 대해 이 예외를 일으킬 수 있는 시나리오를 설명합니다.

Note

보안상의 이유로 리소스가 없거나 호출자가 리소스에 액세스할 수 없는 경우에도 동일한 오류가 반환됩니다. 이렇게 하면 유효한 리소스 식별자가 열거되지 않습니다.

리소스 유형	가능한 시나리오
<code>AgreementProposal</code>	• 지정된 식별자에 대한 계약 제안을 찾을 수 없습니다. • 제안은 수락자를 대상으로 하지 않습니다. • 수락자의 국가에서는 제안을 사용할 수 없습니다.
<code>Agreement</code>	• 지정된 식별자에 대한 계약을 찾을 수 없습니다. • 호출자는 계약의 수락자도 제안자도 아닙니다.
<code>AgreementRequest</code>	• 지정된 식별자에 대한 계약 요청을 찾을 수 없습니다. • 계약 요청은 다른 호출자에게 속합니다. • 계약 요청이 만료되었습니다. • 계약 요청이 동일한 제품 또는 제안에 대해 이후에 수락된 요청으로 대체되었습니다.

리소스 유형	가능한 시나리오
	재게시 제안 또는 제안 만료로 인해 CreateAgreementRequest 및 AcceptAgreementRequest 호출 간에 계약 제안이 비활성화되었습니다.
Charge	- 지정된 식별자에 대한 요금이 없습니다. - 요금은 지정된 계약에 속하지 않습니다.
BillingAdjustmentRequest	- 지정된 식별자에 대한 결제 조정 요청을 찾을 수 없습니다. - 호출자는 조정과 관련된 계약의 수락자도 제안자도 아닙니다.
AgreementCancellationRequest	- 지정된 식별자에 대한 계약 취소 요청을 찾을 수 없습니다. - 호출자는 취소 요청과 관련된 계약의 수락자도 제안자도 아닙니다.
PaymentRequest	- 지정된 식별자에 대한 결제 요청을 찾을 수 없습니다. - 호출자는 결제 요청과 관련된 계약의 수락자도 제안자도 아닙니다.

AWS Marketplace 측정 API 사용

이 참조는 API에 AWS Marketplace Metering Service 대한 설명을 제공합니다. AWS Marketplace 판매자는 이 API를 사용하여 사용자 지정 사용 차원에 대한 데이터를 제출할 수 있습니다. 이 API를 사용하는 데 필요한 권한에 대한 자세한 내용은 판매자 안내서의 [AWS Marketplace 측정 및 권한 부여 API 권한](#)을 참조하세요. AWS Marketplace

측정 레코드 제출

[MeterUsage](#)

- AWS Marketplace 제품에 대한 측정 레코드를 제출합니다.
- 발신자: Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스 또는 Amazon Elastic Kubernetes Service(Amazon EKS) 또는 Amazon Elastic Container Service(Amazon ECS)에서 실행되는 컨테이너
- 지원되는 제품 유형: Amazon Machine Image(AMIs) 및 컨테이너
- 공급업체 측정 태그 지정: 지원되는 할당 태그 지정

[BatchMeterUsage](#)

- 고객 집합에 대한 측정 레코드를 제출합니다.는 BatchMeterUsage API 호출을 AWS CloudTrail 캡처합니다. CloudTrail을 사용하여 eventName의를 사용하여 레코드를 검색하여 전송한 구독형 소프트웨어(SaaS) 측정 레코드가 정확한지 확인합니다BatchMeterUsage. CloudTrail을 사용하여 시간 경과에 따른 레코드를 감사할 수도 있습니다. 자세한 내용은 [CloudTrail 개념](#)을 참조하세요.
- 호출 출처: SaaS 애플리케이션
- 지원되는 제품 유형: SaaS
- 공급업체 측정 태그 지정: 할당 태그 지정 지원

신규 고객 수락

[ResolveCustomer](#)

- 등록 프로세스 중에 구매자가 브라우저를 통해 제출하는 등록 토큰을 해결합니다. CustomerAWSAccountId 및 CustomerIdentifier 함께를 가져옵니다ProductCode.
- 발신자: 등록 프로세스 중 SaaS 애플리케이션
- 지원되는 제품 유형: SaaS
- 공급업체 측정 태그 지정: 해당 사항 없음

유료 컨테이너 제품에 대한 권한 부여 및 측정

[RegisteredUsage](#)

- 소프트웨어 권한 부여 및 측정 기능을 제공합니다. 를 통해 판매되는 유료 컨테이너 소프트웨어 제품은와 통합 AWS Marketplace Metering Service 되고 RegisterUsage 작업을 호출 AWS Marketplace 해야 합니다. Amazon ECS 또는 Amazon EKS용 무료 및 기존 보유 라이선스 사용 모델 (BYOL) 제품은를 호출할 필요가 없습니다RegisterUsage. 그러나 판매자 보고서에서 사용 데이터를 수신하려는 경우 그렇게 할 수 있습니다. RegisterUsage 작업 사용에 대한 자세한 내용은의 [컨테이너 기반 제품을 참조하세요 AWS Marketplace](#).
- 발신자: 유료 컨테이너 소프트웨어 제품
- 지원되는 제품 유형: 컨테이너
- 공급업체 측정 태그 지정: 해당 사항 없음

AWS Marketplace 권한 API 사용

AWS Marketplace 권한 부여 서비스는 지정된 제품에 대한 고객의 권한을 결정하는 데 사용됩니다. 권한은 고객이 소유한 제품의 용량을 나타냅니다. 예를 들어 고객이 SaaS 애플리케이션의 일부 사용자 또는 시트를 소유하거나 다중 테넌트 데이터베이스의 일부 데이터 용량을 소유할 수 있습니다.

AWS Marketplace 배포 API 사용

AWS Marketplace 배포 서비스는 서비스형 소프트웨어(SaaS) 제품의 배포 옵션인 빠른 시작을 지원합니다. 빠른 시작은 제품을 구성, 배포 및 시작하는 데 도움이 될 수 있습니다. AWS Marketplace 배포 서비스는 구매자에게 배포 파라미터(예: API 키 및 외부 IDs 전달할 수 있는 안전한 방법을 판매자에게 제공합니다.

Note

판매자를 위한 빠른 시작에 대한 자세한 내용은 [빠른 시작 구성](#)을 참조하세요.

AWS Marketplace Discovery API 사용

AWS Marketplace Discovery API는 AWS Marketplace 카탈로그에 대한 프로그래밍 방식의 액세스를 제공합니다. 이를 사용하여 제품 및 요금 정보를 검색하고, 통합 조달 경험을 구축하고, 사용자 지정 매장을 만들 수 있습니다.

서비스 엔드포인트

Discovery API는 다음 엔드포인트 형식을 사용합니다.

```
https://discovery-marketplace.region.api.aws
```

예를 들어 미국 동부(버지니아 북부)에서 API를 호출하려면

```
https://discovery-marketplace.us-east-1.api.aws
```

API 버전

현재 API 버전은 `2026-02-05`.

데이터 모델

Discovery API는 AWS Marketplace 카탈로그를 다음 개체로 구성합니다.

- **목록** - 구매자에게 표시되는 제품 또는 다중 제품 솔루션. 목록에는 설명, 하이라이트, 범주, 배지, 요금 모델, 요금 단위, 리뷰, 프로모션 미디어, 판매자 참여, 이행 옵션 유형, 관련 제품 및 제안에 대한 참조가 포함됩니다. `GetListing`를 사용하여 목록을 검색하거나 `SearchListings`를 사용하여 목록 간에 검색합니다.
- **제품** - 판매 중인 기본 소프트웨어 또는 서비스입니다. 제품에는 구매자가 제품을 배포하거나 액세스하는 방법을 설명하는 설명, 하이라이트, 범주, 프로모션 미디어, 판매자 참여 및 이행 옵션(예: AMI, SaaS, 컨테이너 또는 Helm)이 포함됩니다. `GetProduct`를 사용하여 제품 세부 정보를 검색하고 `ListFulfillmentOptions`를 사용하여 제품에 대한 세부 이행 옵션을 검색합니다.
- **제안** - 요금 모델, 기록 판매자, 가용성 날짜 및 배지를 포함한 제품의 요금 약정입니다. 제안에는 사용량 기반 요금, 고정 선결제 요금, 무료 평가판 기간, 법률 문서, 결제 일정 및 갱신 조건과 같은 상업적 조건이 포함됩니다. `ListPurchaseOptions`를 사용하여 제품에 사용할 수 있는 모든 제안을 찾고, 제안의 세부 정보를 `GetOffer` 검색하고, 제안의 특정 조건을 `GetOfferTerms` 검색합니다.
- **제안 세트** - 다중 제품 솔루션의 각 제품에 대한 비공개 제안의 그룹화된 모음입니다. 제안 세트를 사용하면 구매자가 모든 제안을 함께 검토하고 단일 작업으로 동시에 수락할 수 있습니다. `ListPurchaseOptions`를 사용하여 제품에 사용 가능한 모든 제안 세트를 찾고, 제안 세트의 세부 정보를 `GetOfferSet` 검색하고, 제안의 세부 정보를 `GetOffer` 검색하고, 제안의 특정 조건을 `GetOfferTerms` 검색합니다.

Authentication

Discovery API는 표준 AWS 서명 버전 4(SigV4) 인증을 사용합니다. API를 호출하려면 유효한 AWS 자격 증명과 적절한 IAM 권한이 있어야 합니다. 자세한 내용은 [AWS Marketplace Discovery API에 대한 액세스 제어](#)를 참조하세요.

요청하기

모든 Discovery API 작업은 JSON 요청 본문과 함께 HTTP POST 메서드를 사용합니다. 작업 이름은 URL 경로에 지정됩니다.

응답 형식

모든 응답은 JSON 형식으로 반환됩니다. 응답이 성공하면 HTTP 상태 코드 200이 반환됩니다. 오류 응답에는 오류 유형과 메시지가 포함됩니다. 자세한 내용은 [일반 오류를 참조하세요](#).

AWS SDK 사용

Discovery API를 호출하는 권장 방법은 AWS SDK를 사용하는 것입니다. SDK는 인증, 요청 서명, 직렬화 및 오류 처리를 자동으로 처리합니다.

```
# Python (Boto3) example
import boto3

client = boto3.client('marketplace-discovery', region_name='us-east-1')

response = client.get_listing(
    listingId='listing-saas-abc123'
)

print(response['listingName'])
```

```
// JavaScript (AWS SDK v3) example
import { MarketplaceDiscoveryClient, GetListingCommand } from "@aws-sdk/client-marketplace-discovery";

const client = new MarketplaceDiscoveryClient({ region: "us-east-1" });
const response = await client.send(new GetListingCommand({
    listingId: "listing-saas-abc123"
}));

console.log(response.listingName);
```

페이지 매김

목록을 반환하는 작업(예: ListPurchaseOptions 및 SearchFacets)은 사용하여 페이지 매김을 지원합니다. 응답에 nextToken 값이 포함된 경우 다음 요청에 전달하여 추가 결과를 검색합니다.

Throttling

Discovery API는 요청 속도 제한을 적용하여 서비스 가용성을 보장합니다. 속도 제한을 초과하면 API가 ThrottlingException (HTTP 429)를 반환합니다. 애플리케이션에서 지수 백오프 및 재시도 로직을 구현합니다.

AWS Marketplace API AWS 리전 에 대해 지원됨

다음 섹션에서는 지원되는에 대한 정보를 제공합니다 AWS 리전.

주제

- [AWS Marketplace 카탈로그 API에 AWS 리전 지원됨](#)
- [AWS Marketplace 계약 API에 AWS 리전 지원됨](#)
- [AWS Marketplace 측정 API에 AWS 리전 지원됨](#)
- [AWS Marketplace 권한 API에 AWS 리전 지원됨](#)
- [AWS Marketplace 배포 API AWS 리전 지원](#)
- [AWS Marketplace Discovery API AWS 리전 지원](#)

AWS Marketplace 카탈로그 API에 AWS 리전 지원됨

다음 엔드포인트를 AWS 리전 사용하여 미국 동부(버지니아 북부) AWS Marketplace Catalog API 에 서에 액세스할 수 있습니다.

```
catalog.marketplace.us-east-1.amazonaws.com
```

AWS Marketplace 계약 API에 AWS 리전 지원됨

다음 엔드포인트를 사용하여 미국 동부(버지니아 북부) AWS 리전 에서 AWS Marketplace 계약 서비스 에 액세스할 수 있습니다.

```
agreement-marketplace.us-east-1.api.aws (IPv4, IPv6)
```

AWS Marketplace 측정 API에 AWS 리전 지원됨

BatchMeterUsage 리전 지원

BatchMeterUsage는 다음에서 지원됩니다. AWS 리전

상용 리전

eu-north-1, me-south-1, ap-south-1, eu-west-3, ap-southeast-3, us-east-2, af-south-1, eu-west-1, me-central-1, eu-central-1, sa-east-1, ap-east-1, ap-south-2, us-east-1, ap-northeast-2, ap-northeast-3,

eu-west-2, ap-southeast-4, eu-south-1, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-1, ap-east-1, ap-

중국 리전

cn-northwest-1

Amazon EC2에 대한 MeterUsage 리전 지원

MeterUsage는 다음에서 지원됩니다. AWS 리전

상용 리전

eu-north-1, me-south-1, ap-south-1, eu-west-3, ap-southeast-3, us-east-2, af-south-1, eu-west-1, me-central-1, eu-central-1, sa-east-1, ap-east-1, ap-south-2, us-east-1, ap-northeast-2, ap-northeast-3, eu-west-2, ap-southeast-4, eu-south-1, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-1, ap-east-1, ap-

AWS GovCloud (US) 리전

us-gov-east-1, us-gov-west-1

Note

측정 AWS 리전 예를 추가하려면 [AWS Marketplace 판매자 작업에](#) 문의하세요.

Amazon ECS 및 Amazon EKS에 대한 MeterUsage 리전 지원

MeterUsage는 다음에서 Amazon ECS 및 Amazon EKS에 대해 지원됩니다. AWS 리전

상용 리전

us-east-1, us-east-2, us-west-1, us-west-2, eu-west-1, eu-central-1, eu-west-2, eu-west-3, eu-north-1, ap-east-1, ap-southeast-1, ap-northeast-1, ap-southeast-2, ap-northeast-2, ap-south-1, ca-central-1, sa-east-1.

Note

Amazon ECS 및 Amazon EKS AWS GovCloud (US) 의 경우에서 MeterUsage를 사용할 수 없습니다.

Amazon Bedrock AgentCore 런타임에 대한 MeterUsage 리전 지원

MeterUsage는 다음에서 Amazon Bedrock AgentCore 런타임에 대해 지원됩니다. AWS 리전

상용 리전

us-east-1

Note

Amazon Bedrock AgentCore 런타임 AWS GovCloud (US) 의 경우에서 MeterUsage를 사용할 수 없습니다.

RegisterUsage 리전 지원

RegisterUsage는 다음에서 지원됩니다. AWS 리전

상용 리전

eu-north-1, ap-south-1, eu-west-3, us-east-2, eu-west-1, eu-central-1, sa-east-1, ap-east-1, us-east-1, ap-northeast-2, eu-west-2, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-southeast-2, ca-central-1

ResolveCustomer 리전 지원

ResolveCustomer는 다음에서 지원됩니다. AWS 리전

상용 리전

eu-north-1, me-south-1, ap-south-1, eu-west-3, ap-southeast-3, us-east-2, af-south-1, eu-west-1, me-central-1, eu-central-1, sa-east-1, ap-east-1, ap-south-2, us-east-1, ap-northeast-2, ap-northeast-3, eu-west-2, ap-southeast-4, eu-south-1, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-1, ap-east-1, ap-

중국 리전

cn-northwest-1

AWS Marketplace 권한 API에 AWS 리전 지원됨

다음 엔드포인트를 AWS 리전 사용하여 미국 동부(버지니아 북부) AWS Marketplace Entitlement Service 에서에 액세스할 수 있습니다.


```
entitlement.marketplace.us-east-1.amazonaws.com
```

```
entitlement-marketplace.us-east-1.api.aws
```

AWS Marketplace 배포 API AWS 리전 지원

다음 엔드포인트를 사용하여 미국 동부(버지니아 북부) AWS 리전 에서 AWS Marketplace 배포 서비스에 액세스할 수 있습니다.

```
deployment-marketplace.us-east-1.amazonaws.com
```

AWS Marketplace Discovery API AWS 리전 지원

AWS Marketplace Discovery API는 AWS 리전다음에서 사용할 수 있습니다.

- 미국 동부(버지니아 북부) - us-east-1
- 미국 서부(오레곤) - us-west-2
- 유럽(아일랜드) – eu-west-1

AWS Marketplace API에 대한 권한

다음 섹션에서는 권한에 대한 정보를 제공합니다.

주제

- [AWS Marketplace 카탈로그 API에 대한 액세스 제어](#)
- [AWS Marketplace 계약 API에 대한 액세스 제어](#)
- [AWS Marketplace 측정 API에 대한 액세스 제어](#)
- [AWS Marketplace 권한 부여 API에 대한 액세스 제어](#)
- [AWS Marketplace 배포 API에 대한 액세스 제어](#)
- [AWS Marketplace 규정 준수 API에 대한 액세스 제어](#)
- [AWS Marketplace Discovery API에 대한 액세스 제어](#)
- [AWS Marketplace 보고 API에 대한 액세스 제어](#)

AWS Marketplace 카탈로그 API에 대한 액세스 제어

AWS Marketplace Catalog API를 사용하여 [에서 판매자 제품을 AWS Marketplace](#) 관리하거나 [프라이빗 마켓플레이스에서 경험을 관리할 수 있습니다](#). 그러나 먼저 사용자 또는 역할이 호출하려는 API 기능에 액세스할 수 있는지 확인합니다.

AWS Identity and Access Management(IAM)를 사용하여 사용자 및 역할을 생성하고 최종 사용자에게 제한된 권한을 부여하는 정책을 할당합니다. 정책은 사용자 또는 역할을 통해 리소스에 대해 수행할 수 있는 작업을 정의합니다 AWS Marketplace Catalog API.

예를 들어 엔지니어링, 마케팅 및 요금과 같은 역할을 정의할 수 있습니다. 그런 다음 조직의 사용자를 엔지니어링 역할에 추가할 수 있습니다. 해당 역할에서 새 버전의 판매자 제품을 게시하기 위한 변경 요청을 시작할 수 있는 권한이 부여될 수 있습니다. 그러나 엔지니어링 역할은 사용자가 모든 변경 세트를 나열하도록 허용하지 않습니다.

Note

제품을 판매하려면 AWS Marketplace를 판매자 계정으로 설정해야 AWS 계정합니다. AWS Marketplace 판매자가 되는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [판매자로 시작하기](#)를 참조하세요.

AWS 관리형 정책을 사용하거나 자체 IAM 정책을 생성하여 AWS 관리형 정책에서 사용할 수 있는 것보다 더 세부적으로 제어할 수 있습니다. 이러한 접근 방식에 대한 자세한 내용은 다음 주제를 참조하세요.

주제

- [AWS 관리형 정책을 사용하여 작업 허용](#)
- [모든 리소스에 대한 작업 허용](#)
- [특정 리소스에 대한 작업 허용](#)
- [특정 ChangeType 조건 키를 사용하여 작업 허용](#)
- [특정 aws:ResourceTag 조건 키를 사용하여 작업 허용](#)
- [사용자 지정 IAM 역할 생성](#)
- [리소스의 태그 관리](#)
- [리소스 변경 요청 시 태그 관리](#)
- [리소스의 태그를 관리할 수 있는 권한 부여](#)

- [리소스에 특정 태그가 있는 경우에만 리소스의 태그를 관리할 수 있는 권한 부여](#)
- [태그를 통해서만 엔터티를 생성하고 세트를 변경할 수 있는 권한 부여](#)

AWS관리형 정책을 사용하여 작업 허용

에서 관리하는 정책을 사용하여 사용자 또는 역할에 권한을 AWS부여할 수 있습니다.

판매하는 제품을 사용하려면 다른 권한 AWS Marketplace Catalog API외에도에 대한 전체 액세스 권한이 있는 AWSMarketplaceSellerFullAccess IAM 관리형 정책을 사용할 AWS Marketplace 수 있습니다. AWSMarketplaceSellerProductsReadOnly 정책을 사용하여 Catalog API에 대한 읽기 전용 액세스 권한을 부여할 수 있습니다. 자세한 내용은 판매자 안내서의 판매자 [에 대한 액세스 제어](#) [AWS Marketplace Management Portal](#), [AWS Marketplace 판매자에 대한 정책 및 권한](#), [AWS Marketplace 판매자에 대한 AWS 관리형 정책을 참조하세요](#). AWS Marketplace

프라이빗 마켓플레이스를 관리하려면 계정 또는 AWS 조직의 프라이빗 마켓플레이스를 생성하고 편집할 수 있는 전체 액세스 권한이 있는 AWSPrivateMarketplaceAdminFullAccess IAM 관리형 정책을 사용할 수 있습니다. 자세한 내용은 구매자 안내서의 [AWS Marketplace 구독에 대한 액세스 제어](#), [프라이빗 마켓플레이스 관리자 생성](#) 및 [AWS Marketplace 구매자를 위한 AWS 관리형 정책을 참조하세요](#). AWS Marketplace

또는 자체 IAM 정책을 생성하여 AWS관리형 정책에서 사용할 수 있는 것보다 더 세부적으로 제어할 수 있습니다. 다음 주제를 사용하여 자체 IAM 정책을 생성합니다.

모든 리소스에 대한 작업 허용

리소스는 작업이 수행할 수 있는 객체입니다. 모든 리소스 유형을 모든 작업에서 지정할 수 있는 것은 아닙니다. 일부 리소스 유형은 특정 작업에서만 유효합니다. 자세한 내용은 서비스 승인 참조 [의 AWS Marketplace 카탈로그에 사용되는 작업, 리소스 및 조건 키를](#) 참조하세요.

카탈로그 API에는 두 가지 리소스 유형이 있습니다.

- 개체 - 개체는 [의 판매자 제품 AWS Marketplace](#) 또는 [프라이빗 마켓플레이스에서의 경험입니다](#).
- ChangeSet - Catalog API를 사용하여 개체를 변경할 때마다 변경 세트가 반환됩니다. 변경 세트는 요청된 변경 사항과 상태를 설명합니다. 상태가 PREPARING 상태인 경우 변경 세트를 취소할 수 있습니다.

사용자 또는 역할에의 모든 엔터티를 변경할 수 있는 권한을 허용하려면 다음 IAM 정책을 추가할 AWS 계정수 있습니다. 이 정책을 사용하면 사용자 또는 역할이 모든 리소스()에서 StartChangeSet 작업을 사용할 수 있습니다 "*").

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

카탈로그 API에 사용할 수 있는 모든 작업에 대한 자세한 내용은 서비스 권한 부여 참조의 [AWS Marketplace 카탈로그에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.

특정 리소스에 대한 작업 허용

Note

StartChangeSet 작업에 대한 리소스 수준 권한 및 조건 컨텍스트 키는 카탈로그 API와 함께 사용되는 경우에만 지원됩니다. 와 함께 사용할 때는 지원되지 않습니다. [AWS Marketplace Management Portal](#).

모든 리소스에 대한 변경을 허용하는 대신 리소스 수준 권한을 사용하여 특정 리소스에 대한 변경을 허용할 수 있습니다.

예를 들어 모든 판매자 제품 AWS 계정대신의 특정 판매자 제품에 대한 변경을 허용할 수 있습니다. 이렇게 하려면 IAM 정책의에서 판매자 제품의 Amazon 리소스 이름(ARN)Resource를 지정합니다.

Note

새 변경 세트를 생성하는 작업을 통해 세분화된 리소스 수준 권한을 지정하려면 리소스 목록에 ChangeSet ARN도 포함해야 합니다. ChangeSet ARN에는 그림과 같이 생성된 새 변경 세트 ID와 일치하도록 와일드카드(/*)가 포함되어야 합니다.

```
{
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    "aws-marketplace:StartChangeSet"
  ],
  "Resource": [
    "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/example1-abcd-1234-5ef6-7890abcdef12",
    "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
  ]
}
```

또 다른 예로, 모든 경험 대신 프라이빗 마켓플레이스의 특정 경험에 대한 변경을 허용할 수 있습니다. 이렇게 하려면 IAM 정책의에서 경험Resource의 ARN을 지정합니다.

Note

새 변경 세트를 생성하는 작업을 통해 세분화된 리소스 수준 권한을 지정하려면 리소스 목록에 ChangeSet ARN도 포함해야 합니다. ChangeSet ARN에는 그림과 같이 생성된 새 변경 세트 ID와 일치하도록 와일드카드(/*)가 포함되어야 합니다.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Experience/exp-example12345",
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
      ]
    }
  ]
}
```

특정 ChangeType 조건 키를 사용하여 작업 허용

Note

StartChangeSet 작업에 대한 리소스 수준 권한 및 조건 컨텍스트 키는 카탈로그 API와 함께 사용되는 경우에만 지원됩니다. 와 함께 사용할 때는 지원되지 않습니다. [AWS Marketplace Management Portal](#).

카탈로그 API 작업 StartChangeSet에는 여러 가지 변경 유형이 있습니다. 특정 변경 유형에 대한 액세스만 허용할 수 있습니다.

예를 들어 제품 제목과 같은 판매자 제품의 메타데이터에 대한 변경만 허용하고 새 제품 버전 추가는 허용하지 않을 수 있습니다. 이 예제에서 변경 유형을 UpdateInformation 사용하면 제목을 포함하여 판매자 제품의 메타데이터를 변경할 수 있습니다. 다양한 변경 유형에 대한 자세한 내용은 AWS Marketplace Catalog API 참조의 [판매자 제품 작업](#) 및 [프라이빗 마켓플레이스 작업을](#) 참조하세요.

작업을 하나 이상의 변경 유형으로 제한하려면 조건 키 ChangeType에를 지정합니다. 다음 예제 IAM 정책에서 조건 연산자는가와 ChangeType 일치하는 경우에만 작업이 허용되도록 StringEquals 지정합니다 UpdateInformation. 조건 키에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서의 [조건 연산자를](#) 참조하세요.

Note

새 변경 세트를 생성하는 작업을 통해 세분화된 리소스 수준 권한을 지정하려면 리소스 목록에 ChangeSet ARN도 포함해야 합니다. ChangeSet ARN에는 그림과 같이 생성된 새 변경 세트 ID와 일치하도록 와일드카드(/*)가 포함되어야 합니다.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/AmiProduct/example1-abcd-1234-5ef6-7890abcdef12",
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
      ]
    }
  ]
}
```

```

    ],
    "Condition": {
      "StringEquals": {
        "catalog:ChangeType": "UpdateInformation"
      }
    }
  }
]
}

```

특정 aws:ResourceTag 조건 키를 사용하여 작업 허용

Note

StartChangeSet 작업에 대한 리소스 수준 권한 및 조건 컨텍스트 키는 카탈로그 API와 함께 사용되는 경우에만 지원됩니다. 와 함께 사용할 때는 지원되지 않습니다 [AWS Marketplace Management Portal](#).

정책을 계속 업데이트하고 엔터티 ARNs. 리소스 태그 지정을 사용하여이 작업을 수행할 수 있습니다. 리소스에 태그를 추가하면 태그를 기반으로 해당 리소스에 대한 액세스를 제어할 수 있습니다. 예를 들어 각 판매자 제품에 대해 개별 ARNs을 지정하지 않고 판매자 제품 그룹을 설명하는 것을 허용할 수 있습니다.

예를 들어 다음 IAM 정책은 태그 키가 product-team 이고 태그 값이 인 모든 엔터티 리소스("*")에 대한 DescribeEntity 작업을 허용합니다team-xyz.

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:DescribeEntity"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}

```

```
]
}
```

특정 태그로 생성된 변경 세트를 설명하고 취소하도록 허용할 수도 있습니다.

예를 들어 다음 IAM 정책은 태그 키가 DescribeChangeSet 이고 태그 값이 인 변경 세트 리소스 ("*")에 대해 product-team 및 CancelChangeSet 작업을 허용합니다team-xyz.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:DescribeChangeSet",
        "aws-marketplace:CancelChangeSet"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

또한 엔터티에 특정 태그가 있는 경우에만 엔터티에 대한 시작 변경 세트를 허용할 수 있습니다. 예를 들어 특정 태그가 있는 판매자 제품에 대한 변경을 허용할 수 있습니다.

예를 들어 다음 IAM 정책은 태그 키가 product-team 이고 태그 값이 인 모든 엔터티 리소스("*")에 대해 StartChangeSet 작업을 허용합니다team-xyz. 또한 변경 세트를 생성할 때 동일한 태그 키와 값으로 태그가 지정되도록 TagResource 작업이 필요합니다.

Note

StartChangeSet 작업을 허용하는 정책에 특정 태그와 일치하는 조건이 포함된 경우 동일한 정책에도 TagResource 작업이 포함되어야 합니다. 이는 정책 조건이 개체의 태그와 변경 요청으로 인해 새로 생성된 변경 세트의 태그 모두와 일치해야 하기 때문입니다. 따라서 사용자 또는 역할에 새로 생성된 변경 세트에 태그를 지정할 수 있는 권한도 있어야 합니다. 변경 세트

를 시작하고 변경 세트에 태그를 지정하는 예제는 섹션을 참조하세요 [the section called “예: 개체에 태그 추가 및 생성 중 세트 변경”](#).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet",
        "aws-marketplace:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

사용자 지정 IAM 역할 생성

재판매 권한 부여 ChangeType 또는 CPPO ChangeType을 사용하려는 고객은 사용자 지정AWS Identity and Access Management(IAM) 역할을 생성해야 합니다. 이는 재판매 권한 부여 제품 수명 주기의 생성을 지원합니다.

사용자 지정 IAM 역할을 생성하려면

1. IAM 콘솔(<https://console.aws.amazon.com/iam/>)에 로그인합니다.
2. 액세스 관리(Access management)에서 정책(Policies)을 선택합니다.
3. 정책 생성을 선택합니다.
4. 1단계: 권한 지정의 경우
 - a. 정책 편집기에서 JSON 버튼을 선택한 다음 다음 정책을 추가합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowResaleAuthorizationShareActionsRAMCreate",
      "Effect": "Allow",
      "Action": [
        "ram:CreateResourceShare",
        "ram:AssociateResourceShare"
      ],
      "Resource": [
        "arn:aws:ram:*:*:*"
      ],
      "Condition": {
        "ArnLikeIfExists": {
          "ram:ResourceArn": "arn:aws:aws-marketplace:*:*:AWSMarketplace/ResaleAuthorization/*"
        },
        "StringEqualsIfExists": {
          "ram:RequestedResourceType": "aws-marketplace:Entity"
        }
      }
    },
    {
      "Sid": "AllowResaleAuthorizationShareActionsRAMAccept",
      "Effect": "Allow",
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourceShareInvitations",
        "ram:GetResourcePolicies",
        "ram:GetResourceShareAssociations"
      ],
      "Resource": [
        "arn:aws:ram:*:*:*"
      ]
    },
    {
      "Sid": "AllowResaleAuthorizationShareActionsMarketplace",
      "Effect": "Allow",
      "Action": [
```

```

    "aws-marketplace:PutResourcePolicy",
    "aws-marketplace:GetResourcePolicy",
    "aws-marketplace:DescribeEntity"
  ],
  "Resource": "arn:aws:aws-marketplace:*:*:AWSMarketplace/
ResaleAuthorization/*"
}
]
}

```

b. 다음을 선택합니다.

5. 2단계: 검토 및 생성의 경우

- a. 정책 세부 정보에 정책 이름 **FullResaleAuthorizationAccess** 아래에를 입력하고 선택적 설명을 입력합니다.
- b. 이 정책에 정의된 권한을 검토하세요.
- c. 태그 추가에서 태그를 추가합니다(선택 사항).
- d. 정책 생성을 선택합니다.

FullResaleAuthorizationAccess 정책을 생성했습니다.

6. 액세스 관리에서 역할을 선택합니다.

7. 역할 생성을 선택합니다.

8. 1단계: 신뢰할 수 있는 엔터티 선택의 경우

- a. 신뢰할 수 있는 엔터티 유형(Trusted entity type)에서 사용자 지정 정책(Custom trust policy)을 선택합니다.
- b. 다음 사용자 지정 신뢰 정책을 복사하여 JSON 편집기에 붙여넣습니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "resale-
authorization.marketplace.amazonaws.com"
      },

```

```

      "Action": "sts:AssumeRole"
    }
  ]
}

```

c. 다음을 선택합니다.

9. 2단계: 권한 추가의 경우

a. 검색줄에 **FullResaleAuthorizationAccess**를 입력합니다.

b. FullResaleAuthorizationAccess 권한 정책을 선택한 후 다음을 선택합니다.

10. 3단계: 이름 지정, 검토 및 생성의 경우

1. 역할 세부 정보에 역할 이름으로 **FullResaleAuthorizationAccess**를 입력하고 선택적 설명을 입력합니다.
2. 1단계: 신뢰할 수 있는 엔터티 선택에서 선택한 정책 이름이 역할에 연결되어 있는지 확인합니다.
3. 2단계: 권한 추가에서 정책 이름을 검토합니다.
4. 3단계: 태그 추가에서 태그를 추가합니다(선택 사항).
5. 역할 생성을 선택합니다.

FullResaleAuthorizationAccess 역할을 생성했습니다.

리소스의 태그 관리

기존 엔터티 또는 변경 세트에서 태그를 추가, 나열 및 제거할 수 있습니다.

리소스에 태그 추가

개체에 태그를 추가하거나 세트를 변경하려면 TagResource API 작업을 사용합니다.

요청

```

POST /TagResource HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ResourceArn": "string",
  "Tags": [
    {

```

```

    "Key": "string",
    "Value": "string"
  }
  ...
]
}

```

요청 파라미터에는 다음이 포함됩니다.

- 카탈로그(문자열) - (필수) 이어야 합니다AWSMarketplace.
- ResourceArn(문자열) - 변경 세트 또는 엔터티의 (필수) ARN입니다. 변경 세트는 Catalog API를 사용하여 수행한 변경 사항을 설명합니다. 엔터티는 [의 판매자 제품AWS Marketplace](#)이거나 [프라이빗 마켓플레이스에서의 경험](#)일 수 있습니다.
- 태그(객체 배열) - (필수) 각 태그 키와 값을 지정하는 객체 목록입니다. 허용되는 객체 수: 1~50개.
 - 키(문자열) - (필수) 태그의 이름입니다.
 - 정규식 패턴 - `^([\p{L}\p{Z}\p{N}_:/+=\-\@]*)$`
 - 문자 길이 - 1~128
 - 값(문자열) - (필수) 태그의 값입니다.
 - 정규식 패턴 - `^([\p{L}\p{Z}\p{N}_:/+=\-\@]*)$`
 - 문자 길이 - 0~256

응답

```
{}
```

리소스에서 태그 제거

개체 또는 변경 세트에서 태그 또는 태그 목록을 제거하려면 UntagResource API 작업을 사용합니다.

요청

```

POST /UntagResource HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ResourceArn": "string",

```

```
"TagKeys": [
  "string"
  ...
]
```

요청 파라미터에는 다음이 포함됩니다.

- 카탈로그(문자열) - (필수) 이어야 합니다AWSMarketplace.
- ResourceArn(문자열) - 변경 세트 또는 엔터티의 (필수) ARN입니다. 변경 세트는 Catalog API를 사용하여 수행한 변경 사항을 설명합니다. 엔터티는 [의 판매자 제품AWS Marketplace](#)이거나 [프라이빗 마켓플레이스에서의 경험](#)일 수 있습니다.
- 태그(객체 배열) - (필수) 제거할 태그의 키 이름 목록입니다. 허용되는 문자열 수: 0~256.

응답

```
{}
```

리소스의 모든 태그 나열

변경 세트 또는 개체에 추가되었지만 아직 제거되지 않은 모든 태그를 나열하려면 ListTagsForResource API 작업을 사용합니다.

요청

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ResourceArn": "string"
}
```

요청 파라미터에는 다음이 포함됩니다.

- 카탈로그(문자열) - (필수) 이어야 합니다AWSMarketplace.
- ResourceArn(문자열) - 변경 세트 또는 개체의 (필수) ARN입니다. 변경 세트는 Catalog API를 사용하여 수행한 변경 사항을 설명합니다. 엔터티는 [의 판매자 제품AWS Marketplace](#)이거나 [프라이빗 마켓플레이스에서의 경험](#)일 수 있습니다.

응답

```
{
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
    ...
  ]
}
```

리소스 변경 요청 시 태그 관리

개체 또는 변경 세트가 생성될 때 태그를 추가할 수 있습니다.

예: 변경 세트를 생성할 때 변경 세트에 태그 추가

다음은 판매자 제품의 제품 메타데이터를 업데이트하는 StartChangeSet 요청의 예입니다. 이 요청은 ChangeSetTags 속성에 태그를 포함하여이 요청으로 생성된 변경 세트에 태그를 추가합니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "AmiProduct@1.0"
      },
      "Details": "{\"ProductTitle\": \"My updated title\"}"
    }
  ],
  "ChangeSetTags": [
    {
      "Key": "product-team",
      "Value": "team-xyz"
    }
  ]
}
```

```
]
}
```

판매자 제품 관리에 대한 자세한 내용은 AWS Marketplace Catalog API 참조의 [판매자 제품 작업을](#) 참조하세요.

예: 개체에 태그 추가 및 생성 중 세트 변경

다음은 프라이빗 마켓플레이스 경험 엔터티를 생성하는 StartChangeSet 요청의 예입니다. 요청은 EntityTags 및 ChangeSetTags 속성에 태그를 포함하여이 요청으로 생성된 개체 리소스와 변경 세트 리소스 모두에 태그를 추가합니다. 이러한 태그를 사용하면 사용자 또는 역할의 권한 정책을 지정하여이 요청이 생성하는 변경 세트의 설명 또는 취소만 허용하거나이 요청이 생성하는 엔터티에 대한 추가 변경 세트의 생성만 허용할 수 있습니다. 자세한 내용은 [태그를 통해서만 엔터티를 생성하고 세트를 변경할 수 있는 권한 부여](#) 단원을 참조하십시오.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateExperience",
      "Entity": {
        "Type": "Experience@1.0"
      },
      "Details": "{\"Name\": \"ExamplePrivateMarketplace\"}",
      "EntityTags": [
        {
          "Key": "product-team",
          "Value": "team-xyz"
        }
      ]
    }
  ],
  "ChangeSetTags": [
    {
      "Key": "product-team",
      "Value": "team-xyz"
    }
  ]
}
```


프라이빗 마켓플레이스 관리에 대한 자세한 내용은 AWS Marketplace Catalog API참조의 [프라이빗 마켓플레이스 작업을](#) 참조하세요.

리소스의 태그를 관리할 수 있는 권한 부여

사용자 또는 역할이 모든 엔터티 또는 변경 세트에 태그를 추가, 제거 및 나열하도록 허용하려면 다음 IAM 정책이 필요합니다.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:TagResource",
        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}
```

리소스에 특정 태그가 있는 경우에만 리소스의 태그를 관리할 수 있는 권한 부여

사용자 또는 역할이 개체에 태그를 추가, 제거 및 나열하거나 특정 태그가 있는 세트를 변경하도록 허용할 수 있습니다. 다음 IAM 정책은 태그 키가 product-team 이고 태그 값이 인 모든 엔터티 리소스 ("*")에서 이러한 작업을 허용합니다team-xyz.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:TagResource",
        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

태그를 통해서만 엔터티를 생성하고 세트를 변경할 수 있는 권한 부여

Note

StartChangeSet 작업에 대한 리소스 수준 권한 및 조건 컨텍스트 키는 카탈로그 API와 함께 사용되는 경우에만 지원됩니다. 와 함께 사용할 때는 지원되지 않습니다 [AWS Marketplace Management Portal](#).

개체 또는 변경 세트가 생성될 때 태그 지정을 적용할 수 있습니다. 태그 키 일치 product-team 및 태그 값 일치를 지정하는 조건과 함께 StartChangeSet 및 TagResource 작업을 허용하려면 다음 정책을 추가합니다 team-xyz. 이 정책 조건은 새로 생성된 개체의 태그와 생성 요청으로 인해 새로 생성된 변경 세트의 태그와 일치해야 합니다. 생성 시 개체에 태그를 지정하는 예제는 섹션을 참조하세요 [the section called “예: 개체에 태그 추가 및 생성 중 세트 변경”](#).

기존 엔터티의 경우 이 정책은 해당 엔터티에 대한 변경을 요청할 때 태그 지정 변경 세트도 적용합니다. 또한 기존 엔터티에 이 기존 태그가 있어야 합니다. 이는 정책 조건이 기존 엔터티의 태그와 변경 요청으로 인해 새로 생성된 변경 세트와 모두 일치해야 하기 때문입니다. 변경 요청에 태그를 추가하는 예제는 섹션을 참조하세요 [the section called “예: 변경 세트를 생성할 때 변경 세트에 태그 추가”](#).

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet",
        "aws-marketplace:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}

```

```
}
```

AWS Marketplace 계약 API에 대한 액세스 제어

계약 서비스를 AWS Marketplace 사용하여의 계약을 관리하려면 AWS Identity and Access Management (IAM) 정책 및 역할이 설정되어 있는지 확인해야 합니다. 사용자가 작업을 수행할 수 있도록 허용하려면 다음 정책/권한이 있어야 합니다.

- `CreateAgreementRequest` - 사용자에게 새 계약을 생성하거나 기존 계약을 업데이트하기 위한 새 요청을 시작할 수 있는 권한을 부여합니다.
- `AcceptAgreementRequest` - 사용자에게 이전에 생성한 요청을 수락하여 계약을 구체화할 수 있는 권한을 부여합니다. 참고: 계약 요청을 수락하는 동안 사용자가 구매 주문을 제공하는 경우 작업을 수행할 수 있는 `UpdatePurchaseOrders` 권한도 있어야 합니다.
- `GetAgreementEntitlements` - 사용자에게 계약의 권한을 나열할 수 있는 권한을 부여합니다.
- `CancelAgreement` - 사용자에게 참여 중인 활성 계약을 취소할 수 있는 권한을 부여합니다.
- `ListAgreementCharges` - 사용자에게 계약에 대한 요금을 나열할 수 있는 권한을 부여합니다.
- `UpdatePurchaseOrders` - 사용자에게 계약 요금에 대한 구매 주문을 업데이트할 수 있는 권한을 부여합니다.
- `SendAgreementCancellationRequest` - 사용자에게 취소 요청을 보낼 수 있는 권한을 부여합니다.
- `AcceptAgreementCancellationRequest` - 수락자로 참여하는 계약에 대해 판매자가 시작한 취소 요청을 수락할 수 있는 권한을 사용자에게 부여합니다. 참고: 취소 요청을 승인하면 계약이 취소되므로 사용자에게 `CancelAgreement` 권한도 필요합니다.
- `RejectAgreementCancellationRequest` - 수락자로 참여하는 계약에 대해 판매자가 시작한 취소 요청을 거부할 수 있는 권한을 사용자에게 부여합니다.
- `SendAgreementPaymentRequest` - 사용자에게 결제 요청을 보낼 수 있는 권한을 부여합니다.
- `AcceptAgreementPaymentRequest` - 사용자가 수락자로 참여하는 계약에 대해 판매자가 시작한 결제 요청을 수락할 수 있는 권한을 부여합니다.
- `RejectAgreementPaymentRequest` - 수락자로 참여하는 계약에 대해 판매자가 시작한 결제 요청을 거부할 수 있는 권한을 사용자에게 부여합니다.
- `GetAgreementCancellationRequest` - 수락자로 참여하는 계약의 특정 취소 요청에 대한 세부 정보를 검색할 수 있는 권한을 사용자에게 부여합니다.
- `ListAgreementCancellationRequests` - 수락자로 참여하는 계약에 대한 취소 요청을 나열할 수 있는 권한을 사용자에게 부여합니다.

- `GetAgreementPaymentRequest` - 수락자로 참여하는 계약의 특정 결제 요청에 대한 세부 정보를 검색할 수 있는 권한을 사용자에게 부여합니다.
- `ListAgreementPaymentRequests` - 수락자로 참여하는 계약에 대한 결제 요청을 나열할 수 있는 권한을 사용자에게 부여합니다.
- `DescribeAgreement` - 사용자에게 계약에 대한 자세한 메타데이터를 가져올 수 있는 권한을 부여합니다.
- `GetAgreementTerms` - 사용자에게 계약 조건에 대한 세부 정보를 얻을 수 있는 권한을 부여합니다.
- `SearchAgreements` - 사용자에게 모든 계약을 검색할 수 있는 권한을 부여합니다.

Note

이러한 권한에 대한 자세한 내용은 판매자 안내서의 [AWS Marketplace 판매자에 대한 정책 및 권한](#) AWS Marketplace 과 구매자 안내서의 [AWS Marketplace 구독에 대한 액세스 제어를 참조하세요](#). AWS Marketplace

AWS 관리형 정책을 사용하여 작업 허용

AWS Marketplace 는 페르소나별 관리형 정책을 정의합니다. 기본적으로 지정된 역할에 대한 정책의 범위를 좁히는 IAM 자격 증명에 관리형 정책을 사용할 수 있습니다. 관리형 정책을 사용하면 Marketplace에서 새 기능을 사용할 수 있게 되면 모범 사례를 따르고 정책 업데이트를 받을 수 있습니다.

이러한 권한에 대한 자세한 내용은 판매자 안내서의 [AWS Marketplace 판매자를 위한 정책 및 권한](#) AWS Marketplace 과 구매자 안내서의 [AWS Marketplace 구독에 대한 액세스 제어를 참조하세요](#). AWS Marketplace

여기에 설명된 AWS 관리형 정책의 권한으로 제한되지 않습니다. IAM을 이용하여 사용자 지정 권한이 있는 정책을 생성한 다음, 해당 정책을 IAM 역할에 추가할 수 있습니다. 자세한 정보는 IAM 사용 설명서에서 [IAM 정책 관리](#) 및 [IAM ID 권한 추가 및 제거](#)를 참조하세요.

고객 관리형 정책을 사용하여 작업 허용

고객 관리형 정책을 정의하여 계약 서비스 APIs, 효과, 작업 및 지원되는 조건을 혼합하여 정책을 사용자 지정할 수 있습니다.

제품 구독 관리를 위한 AWS Marketplace 계약 API에 액세스하려면 다음 정책을 생성하여 액세스를 허용할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:CreateAgreementRequest",
        "aws-marketplace:AcceptAgreementRequest"
      ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws-marketplace:AgreementType": ["PurchaseAgreement"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:CancelAgreement",
        "aws-marketplace:GetAgreementEntitlements",
        "aws-marketplace:UpdatePurchaseOrders",
        "aws-marketplace:ListAgreementCharges"
      ],
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws-marketplace:AgreementType": ["PurchaseAgreement"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:AcceptAgreementCancellationRequest",
        "aws-marketplace:RejectAgreementCancellationRequest",
        "aws-marketplace:GetAgreementCancellationRequest",
        "aws-marketplace:ListAgreementCancellationRequests",
        "aws-marketplace:CancelAgreement"
      ],
    },
  ]
}
```

```

    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "aws-marketplace:AgreementType": ["PurchaseAgreement"]
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:AcceptAgreementPaymentRequest",
      "aws-marketplace:RejectAgreementPaymentRequest",
      "aws-marketplace:GetAgreementPaymentRequest",
      "aws-marketplace:ListAgreementPaymentRequests"
    ],
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "aws-marketplace:AgreementType": ["PurchaseAgreement"]
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:DescribeAgreement",
      "aws-marketplace:SearchAgreements",
      "aws-marketplace:GetAgreementTerms"
    ],
    "Resource": "*",
    "Condition": {
      "ForAllValues:StringEquals": {
        "aws-marketplace:AgreementType": ["PurchaseAgreement"]
      },
      "StringEquals": {
        "aws-marketplace:PartyType": "Acceptor"
      }
    }
  }
]
}

```

서비스 정의 조건 키를 사용하여 정책 사용자 지정

AgreementType 조건 키 사용

아래 문은 PurchaseAgreement 계약 유형에 대해서만 UpdatePurchaseOrders 작업에 대한 액세스를 제한합니다.

```
{
  "Effect": "Allow",
  "Action": [
    "aws-marketplace:UpdatePurchaseOrders"
  ],
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "aws-marketplace:AgreementType": ["PurchaseAgreement"]
    }
  }
}
```

ProductId 조건 키 사용

문 아래는 CreateAgreementRequest 및에 대한 액세스를 지정된 AWS Marketplace 제품 IDAcceptAgreementRequest로만 제한합니다. Bedrock 기본 모델로 제한하는 등 사전 승인된 제품 ID만 생성할 수 있도록 이 문을 포함할 수 있습니다.

```
{
  "Effect": "Allow",
  "Action": [
    "aws-marketplace:CreateAgreementRequest",
    "aws-marketplace:AcceptAgreementRequest"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws-marketplace:ProductId": [
        "model-product-id-1",
        "model-product-id-2"
      ]
    }
  }
}
```

PartyType 조건 키 사용

아래 문은 사용자가 계약의 제안자인 경우 계약의 세부 정보를 읽을 수 있는 액세스 권한을 부여합니다.

```
{
  "Effect": "Allow",
  "Action": [
    "aws-marketplace:DescribeAgreement"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "aws-marketplace:PartyType": "Proposer"
    }
  }
}
```

AWS Marketplace 측정 API에 대한 액세스 제어

AWS Marketplace 측정 API를 AWS Marketplace와 통합하려면 서비스 계정에 제약 AWS Identity and Access Management (IAM) 정책이 있어야 합니다. 자세한 내용은 [AWS Marketplace 측정 및 권한 부여 API 권한을 참조하세요](#).

AWS Marketplace 권한 부여 API에 대한 액세스 제어

AWS Marketplace 권한 부여 API를 AWS Marketplace와 통합하려면 서비스 계정에 제약 AWS Identity and Access Management (IAM)이 있어야 합니다. 자세한 내용은 [AWS Marketplace 측정 및 권한 부여 API 권한을 참조하세요](#).

AWS Marketplace 배포 API에 대한 액세스 제어

에서 배포를 관리하려면 필요한 AWS Identity and Access Management (IAM) 역할 및 권한이 있는지 확인해야 AWS Marketplace합니다.

PutDeploymentParameter 작업을 호출하기 전에 구매자는 AWSServiceRoleForMarketplaceDeployment 서비스 연결 역할을 생성해야 합니다. 이렇게 하면 구매자 계정 AWS Marketplace에서 필요한 배포 파라미터 관련 리소스를 생성, 관리 및 태그 지정하는 데 필요한 권한이 제공됩니다. 구매자는 빠른 시작 환경의 구성 프로세스를 진행하면서 프롬프트를 사용하

여이 역할을 생성합니다. 자세한 내용은 AWS Marketplace 구매자 안내서의 [역할을 사용하여 제품 구성 및 시작](#)을 참조하세요.

를 호출하려면 PutDeploymentParameter 판매자는 다음 작업에 대한 IAM 권한이 있어야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aws-marketplace:PutDeploymentParameter",
        "aws-marketplace:TagResource"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

aws-marketplace:PutDeploymentParameter 작업을 통해 사용자는 PutDeploymentParameter API를 호출할 수 있습니다. API는 선택적 tags 속성도 허용합니다. tags 속성이 요청에 포함된 경우 호출자에게 관련 리소스 aws-marketplace:TagResource에 대한 권한도 있어야 합니다. 사용자 생성에 대한 자세한 내용은 IAM 사용 설명서의 [AWS 계정에서 사용자 생성을 참조하세요](#). 정책 생성 및 할당에 대한 자세한 내용은 [IAM 사용자의 권한 변경](#)을 참조하세요.

AWS Marketplace 규정 준수 API에 대한 액세스 제어

AWS Marketplace 규정 준수 API를 사용하여에서 판매자 규정 준수를 관리할 수 있습니다 AWS Marketplace. 그러나 먼저 사용자 또는 역할이 호출하려는 API 기능에 액세스할 수 있는지 확인합니다.

AWS Identity and Access Management (IAM)를 사용하여 사용자 및 역할을 생성하고 최종 사용자에게 제한된 권한을 부여하는 정책을 할당합니다. 정책은 사용자 또는 역할이 AWS Marketplace 규정 준수 API를 통해 리소스에 대해 수행할 수 있는 작업을 정의합니다.

Note

제품을 판매하려면 AWS Marketplace를 판매자 계정으로 설정해야 AWS 계정입니다. AWS Marketplace 판매자가 되는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [판매자로 시작하기](#)를 참조하세요.

주제

- [AWS 관리형 정책을 사용하여 작업 허용](#)
- [모든 리소스에 대한 작업 허용](#)
- [특정 리소스에 대한 작업 허용](#)
- [특정 aws:ResourceTag 조건 키를 사용하여 작업 허용](#)
- [리소스의 태그 관리](#)
- [리소스의 태그를 관리할 수 있는 권한 부여](#)
- [리소스에 특정 태그가 있는 경우에만 리소스의 태그를 관리할 수 있는 권한 부여](#)
- [인보이스 제출 작업을 시작할 때 태그 요구](#)
- [확인 증거 작업](#)

AWS 관리형 정책을 사용하여 작업 허용

에서 관리하는 정책을 사용하여 사용자 또는 역할에 권한을 AWS 부여할 수 있습니다.

에서 인보이스 제출 작업을 AWS Marketplace수행하려면 다른 권한 외에도 AWS Marketplace 규정 준수 API 작업에 대한 전체 액세스를 포함하는 AWSMarketplaceSellerFullAccess IAM 관리형 정책을 사용할 수 있습니다. 자세한 내용은 [AWS Marketplace 판매자 안내서의 판매자에 대한 정책 및 권한](#)과 [AWS Marketplace 판매자에 대한 AWS 관리형 정책을 참조하세요](#). AWS Marketplace

또는 자체 IAM 정책을 생성하여 AWS 관리형 정책에서 사용할 수 있는 것보다 더 세부적으로 제어할 수 있습니다. 다음 주제를 사용하여 자체 IAM 정책을 생성합니다.

모든 리소스에 대한 작업 허용

리소스는 작업이 수행할 수 있는 객체입니다. 규정 준수 API에는 다음과 같은 리소스 유형이 있습니다.

- InvoiceSubmissionTask - 인보이스 제출 작업은 판매자가 제출한 인보이스의 처리를 추적합니다 AWS Marketplace.
- IssuedTaxInvoice - 판매자를 대신하여 발급한 AWS Marketplace 세금 계산서입니다.

- **VerificationEvidence** - 확인 중인 특정 확인 범주 및 주제에 대한 확인 데이터를 포함합니다.

사용자 또는 역할이 인보이스 제출 작업 작업에 대한 전체 액세스 권한을 허용하려면 다음 IAM 정책을 추가할 수 있습니다. 이 정책을 사용하면 사용자 또는 역할이 모든 리소스()에서 모든 인보이스 제출 작업 작업을 사용할 수 있습니다"".

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartInvoiceSubmissionTask",
        "aws-marketplace:GetInvoiceSubmissionTask",
        "aws-marketplace:ListInvoiceSubmissionTasks",
        "aws-marketplace:ListPayables"
      ],
      "Resource": "*"
    }
  ]
}
```

사용자 또는 역할이 발행된 세금 계산서 작업에 대한 전체 액세스 권한을 허용하려면 다음 IAM 정책을 추가할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:ListIssuedTaxInvoices",
        "aws-marketplace:GetIssuedTaxInvoice"
      ],
      "Resource": "*"
    }
  ]
}
```

사용자 또는 역할이 확인 작업에 대한 모든 액세스 권한을 허용하려면 다음 IAM 정책을 추가할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:CreateVerificationEvidence",
        "aws-marketplace:UpdateVerificationEvidence",
        "aws-marketplace:GetVerificationEvidence",
        "aws-marketplace:ListVerificationEvidence",
        "aws-marketplace:StartVerification",
        "aws-marketplace:GetVerification",
        "aws-marketplace:ListVerifications"
      ],
      "Resource": "*"
    }
  ]
}
```

규정 준수 API에 사용할 수 있는 모든 작업에 대한 자세한 내용은 서비스 승인 참조의 규정 [AWS Marketplace 준수에 사용되는 작업, 리소스 및 조건 키](#)를 참조하세요.

특정 리소스에 대한 작업 허용

리소스 수준 권한을 사용하여 모든 리소스 대신 특정 리소스에 대한 작업을 허용할 수 있습니다. 이렇게 하려면 IAM 정책의에서 리소스Resource의 Amazon 리소스 이름(ARN)을 지정합니다.

다음 예시에서는 특정 인보이스 제출 작업에 대한 GetInvoiceSubmissionTask 작업을 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetInvoiceSubmissionTask"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:catalog/example-catalog/invoice-submission-task/example-task-id"
      ]
    }
  ]
}
```

```

    }
  ]
}
```

다음 예시에서는 발행된 특정 세금 계산서에 대한 `GetIssuedTaxInvoice` 작업을 허용합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetIssuedTaxInvoice"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:catalog/AWSMarketplace/issued-
tax-invoice/example-invoice-id"
      ]
    }
  ]
}
```

다음 예시에서는 특정 확인 증거 리소스에 대한 `GetVerificationEvidence` 작업을 허용합니다.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetVerificationEvidence"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:verification-type/business-
verification/verification-evidence/evidence-a1b2c3d4e5f6g"
      ]
    }
  ]
}
```

특정 aws:ResourceTag 조건 키를 사용하여 작업 허용

개별 ARNs. 리소스에 태그를 추가하면 태그를 기반으로 해당 리소스에 대한 액세스를 제어할 수 있습니다.

예를 들어 다음 IAM 정책은 태그 키가 product-team 이고 태그 값이 인 모든 인보이스 제출 작업 리소스("*")에 대한 GetInvoiceSubmissionTask 작업을 허용합니다team-xyz.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetInvoiceSubmissionTask"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

마찬가지로 다음 IAM 정책은 태그 키가 Department 이고 태그 값이 인 발급된 세금 인보이스 리소스("*")에 대한 GetIssuedTaxInvoice 작업을 허용합니다Tax.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetIssuedTaxInvoice"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "Tax"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

다음 IAM 정책은 태그 키가 Department 이고 태그 값이 인 모든 확인 증거 리소스("*")에 대한 GetVerificationEvidence 작업을 허용합니다Compliance.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetVerificationEvidence"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Department": "Compliance"
        }
      }
    }
  ]
}

```

리소스의 태그 관리

인보이스 제출 작업 및 발행된 세금 인보이스와 같은 기존 규정 준수 API 리소스에서 태그를 추가, 나열 및 제거할 수 있습니다.

리소스에 태그 추가

리소스에 태그를 추가하려면 TagResource API 작업을 사용합니다.

요청

```

POST /TagResource HTTP/1.1
Content-type: application/json

{
  "ResourceArn": "string",
  "Tags": [

```

```
{
  "Key": "string",
  "Value": "string"
}
```

요청 파라미터에는 다음이 포함됩니다.

- ResourceArn(문자열) - 리소스의 (필수) ARN입니다.
- 태그(객체 배열) - (필수) 각 태그 키와 값을 지정하는 객체 목록입니다. 허용되는 객체 수: 1~50개.
 - 키(문자열) - (필수) 태그의 이름입니다. 정규식 패턴: `^([\p{L}\p{Z}\p{N}_:/=+\-@]*)$`. 문자 길이: 1~128.
 - 값(문자열) - (필수) 태그의 값입니다. 정규식 패턴: `^([\p{L}\p{Z}\p{N}_:/=+\-@]*)$`. 문자 길이: 0~256.

리소스에서 태그 제거

리소스에서 태그 또는 태그 목록을 제거하려면 UntagResource API 작업을 사용합니다.

요청

```
POST /UntagResource HTTP/1.1
Content-type: application/json
```

```
{
  "ResourceArn": "string",
  "TagKeys": [
    "string"
  ]
}
```

요청 파라미터에는 다음이 포함됩니다.

- ResourceArn(문자열) - 리소스의 (필수) ARN입니다.
- TagKeys(문자열 배열) - (필수) 제거할 태그의 키 이름 목록입니다.

리소스의 모든 태그 나열

리소스의 모든 태그를 나열하려면 ListTagsForResource API 작업을 사용합니다.

요청

```
POST /ListTagsForResource HTTP/1.1
Content-type: application/json
```

```
{
  "ResourceArn": "string"
}
```

응답

```
{
  "ResourceArn": "string",
  "Tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

리소스의 태그를 관리할 수 있는 권한 부여

사용자 또는 역할이 모든 규정 준수 API 리소스에 태그를 추가, 제거 및 나열하도록 허용하려면 다음 IAM 정책이 필요합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:TagResource",
        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
      ],
      "Resource": "*"
    }
  ]
}
```

리소스에 특정 태그가 있는 경우에만 리소스의 태그를 관리할 수 있는 권한 부여

사용자 또는 역할이 특정 태그가 있는 규정 준수 API 리소스에 태그를 추가, 제거 및 나열하도록 허용할 수 있습니다. 다음 IAM 정책은 태그 키가 `product-team` 이고 태그 값이 인 모든 리소스("*")에서 이러한 작업을 허용합니다 `team-xyz`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:TagResource",
        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/product-team": "team-xyz"
        }
      }
    }
  ]
}
```

인보이스 제출 작업을 시작할 때 태그 요구

`StartInvoiceSubmissionTask` 작업과 함께 `aws:RequestTag` 및 `aws:TagKeys` 조건 키를 사용하여 인보이스 제출 작업이 생성될 때 태그 지정에 적용할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartInvoiceSubmissionTask"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
```

```

        "aws:RequestTag/product-team": "team-xyz"
    },
    "ForAllValues:StringEquals": {
        "aws:TagKeys": [
            "product-team"
        ]
    }
}
]
}

```

확인 증거 작업

AWS Marketplace 규정 준수 API는 확인 증거 및 확인 프로세스를 관리하기 위한 작업을 제공합니다. IAM 정책을 사용하여 이러한 작업에 대한 액세스를 제어할 수 있습니다.

확인 작업을 위한 IAM 작업

확인 작업에 사용할 수 있는 IAM 작업은 다음과 같습니다. 모든 작업은 `aws-marketplace` 네임스페이스를 사용합니다.

증거 관리 작업

- `aws-marketplace:CreateVerificationEvidence` - 새 확인 증거를 생성합니다.
- `aws-marketplace:UpdateVerificationEvidence` - 기존 확인 증거를 업데이트합니다.
- `aws-marketplace:GetVerificationEvidence` - 확인 증거의 세부 정보를 가져옵니다.
- `aws-marketplace:ListVerificationEvidence` - 확인 증거 리소스를 나열합니다.

확인 프로세스 작업

- `aws-marketplace:StartVerification` - 증거를 제출하고 데이터 공유를 활성화합니다.
- `aws-marketplace:GetVerification` - 자세한 확인 상태를 가져옵니다.
- `aws-marketplace:ListVerifications` - 모든 확인 상태를 나열합니다.

리소스 유형

`VerificationEvidence` 리소스에는 특정 확인 범주 및 확인 대상에 대한 확인 데이터가 포함되어 있습니다. 이 리소스의 ARN 형식은 다음과 같습니다.

```
arn:aws:aws-marketplace:region:account-id:verification-type/type-value/verification-evidence/evidence-id
```

다음은 비즈니스 확인을 위한 ARN의 예입니다.

```
arn:aws:aws-marketplace:us-east-1:123456789012:verification-type/business-verification/verification-evidence/evidence-a1b2c3d4e5f6g
```

VerificationType 조건 키 사용

aws-marketplace:VerificationType 조건 키는 유형별로 확인 프로세스 작업을 필터링합니다. 이 조건 키는 다음 작업에 적용됩니다.

- StartVerification
- GetVerification
- ListVerifications

이 조건 키의 유효한 값은 다음과 같습니다.

- BusinessVerification

Note

검증 유형이 리소스 ARN에 이미 인코딩되어 있으므로 증거 관리 작업 (CreateVerificationEvidence, UpdateVerificationEvidence, GetVerificationEvidence, ListVerificationEvidence)은 이 조건 키를 사용하지 않습니다.

AWS 관리형 정책을 사용하여 작업 허용

AWSMarketplaceSellerFullAccess IAM 관리형 정책에는 다른 권한 외에도 7개의 확인 권한이 모두 포함됩니다. 자세한 내용은 AWS Marketplace 판매자 안내서의 [AWS Marketplace 판매자를 위한 AWS 관리형 정책을 참조하세요](#).

모든 확인 작업에 대한 전체 액세스 허용

모든 확인 작업에 대한 사용자 또는 역할의 전체 액세스를 허용하려면 다음 IAM 정책을 추가할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:CreateVerificationEvidence",
        "aws-marketplace:UpdateVerificationEvidence",
        "aws-marketplace:GetVerificationEvidence",
        "aws-marketplace:ListVerificationEvidence",
        "aws-marketplace:StartVerification",
        "aws-marketplace:GetVerification",
        "aws-marketplace:ListVerifications"
      ],
      "Resource": "*"
    }
  ]
}
```

비즈니스 확인 증거에 대한 읽기 전용 액세스 허용

사용자 또는 역할이 비즈니스 확인 증거에 대한 읽기 전용 액세스를 허용하려면 ARN 기반 필터링을 사용하여 특정 확인 유형에 대한 액세스를 제한할 수 있습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetVerificationEvidence",
        "aws-marketplace:ListVerificationEvidence"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:*:123456789012:verification-type/business-verification/verification-evidence/*"
      ]
    }
  ]
}
```

유형별 확인 프로세스 작업 제한

확인 프로세스 작업을 특정 확인 유형으로 제한하려면 `aws-marketplace:VerificationType` 조건 키를 사용합니다. 다음 예제에서는 비즈니스 확인을 위해서만 확인 프로세스 작업을 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartVerification",
        "aws-marketplace:GetVerification",
        "aws-marketplace:ListVerifications"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws-marketplace:VerificationType": "BusinessVerification"
        }
      }
    }
  ]
}
```

특정 확인 증거 리소스에 대한 작업 허용

특정 확인 증거 리소스에 대한 작업을 허용하려면 IAM 정책의 `Resource` 요소에 리소스의 ARN을 지정합니다. 다음 예시에서는 특정 확인 증거 리소스에 대한 모든 증거 관리 작업을 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:GetVerificationEvidence",
        "aws-marketplace:UpdateVerificationEvidence"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:verification-type/business-verification/verification-evidence/evidence-id"
      ]
    }
  ]
}
```

```

    }
  ]
}
```

AWS Marketplace Discovery API에 대한 액세스 제어

Discovery API는 인증 및 권한 부여에 AWS Identity and Access Management (IAM)을 사용합니다. 모든 AWS 고객은 적절한 IAM 권한을 구성하여 Discovery API를 호출할 수 있습니다.

Discovery API 작업을 호출하려면 사용자에게 다음 권한이 있어야 합니다.

- `GetListing` - 목록에 대한 정보를 검색할 수 있는 권한을 부여합니다.
- `GetProduct` - 제품에 대한 정보를 검색할 수 있는 권한을 부여합니다.
- `GetOffer` - 제안에 대한 정보를 검색할 수 있는 권한을 부여합니다.
- `GetOfferTerms` - 제안에 대한 조건을 검색할 수 있는 권한을 부여합니다.
- `GetOfferSet` - 제안 세트에 대한 정보를 검색할 수 있는 권한을 부여합니다.
- `ListPurchaseOptions` - 구매자가 사용할 수 있는 구매 옵션을 나열할 수 있는 권한을 부여합니다.
- `ListFulfillmentOptions` - 제품의 이행 옵션을 나열할 수 있는 권한을 부여합니다.
- `SearchListings` - 제품 목록을 검색할 수 있는 권한을 부여합니다.
- `SearchFacets` - 필터링 목록에 대한 패킷 값을 검색할 수 있는 권한을 부여합니다.

IAM 권한

Discovery API 작업을 호출하려면 IAM 보안 주체(사용자 또는 역할)에 적절한 `aws-marketplace` 권한이 있어야 합니다. Discovery API는 리소스 수준 권한을 지원하므로 IAM 정책의 특정 리소스 유형에 대한 액세스 범위를 지정할 수 있습니다.

다음 IAM 정책은 리소스 수준 범위를 지정하여 모든 Discovery API 작업에 대한 액세스 권한을 부여합니다.

```

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

```

    "Action": [
      "aws-marketplace:GetListing",
      "aws-marketplace:GetProduct",
      "aws-marketplace:GetOffer",
      "aws-marketplace:GetOfferTerms",
      "aws-marketplace:GetOfferSet",
      "aws-marketplace:ListPurchaseOptions",
      "aws-marketplace:ListFulfillmentOptions",
      "aws-marketplace:SearchFacets",
      "aws-marketplace:SearchListings"
    ],
    "Resource": [
      "arn:aws:aws-marketplace::catalog/<catalog-name>/product/*",
      "arn:aws:aws-marketplace::catalog/<catalog-name>/listing/*",
      "arn:aws:aws-marketplace::catalog/<catalog-name>/offer/*",
      "arn:aws:aws-marketplace::catalog/<catalog-name>/offerSet/*",
      "arn:aws:aws-marketplace::catalog/<catalog-name>/purchaseOption/*"
    ]
  }
}

```

를 적절한 카탈로그 식별자(예: AWSMarketplace)<catalog-name>로 바꿉니다.

리소스 ARN 형식

다음 표에는 Discovery API 작업에 사용되는 리소스 유형과 해당 ARN 형식이 나와 있습니다.

리소스 유형	ARN 형식	사용처
제품	arn:aws:aws-marketplace::catalog/ <i>catalog-name</i> /product/ <i>product-id</i>	GetProduct, ListFulfillmentOptions
목록	arn:aws:aws-marketplace::catalog/ <i>catalog-name</i> /listing/ <i>listing-id</i>	GetListing, SearchListings, SearchFacets
제안	arn:aws:aws-marketplace::catalog/ <i>catalog-name</i> /offer/ <i>offer-id</i>	GetOffer, GetOfferTerms

리소스 유형	ARN 형식	사용처
제안 세트	arn:aws:aws-marketplace:::catalog/ <i>catalog-name</i> /offerSet/ / <i>offerSet-id</i>	GetOfferSet
구매 옵션	arn:aws:aws-marketplace:::catalog/ <i>catalog-name</i> /purchase Option/ <i>purchaseOption-id</i>	ListPurchaseOptions

특정 리소스로 작업 범위 지정

Get 작업(GetListing, GetProduct, GetOffer, GetOfferTerms, GetOfferSet) 및 ListFulfillmentOptions 작업의 경우 와일드카드를 사용하는 대신 특정 리소스 IDs로 액세스 범위를 축소할 수 있습니다. 예를 들어 단일 목록으로 액세스를 제한하려면

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:GetListing",
      "Resource": "arn:aws:aws-marketplace:::catalog/AWSMarketplace/
listing/listing-id"
    }
  ]
}
```

이 패턴은 모든 Get 작업 및 ListFulfillmentOptions에서 작동합니다. 리소스 유형과 ID를 적절히 바꿉니다.

- arn:aws:aws-marketplace:::catalog/*catalog-name*/product/*product-id*
- arn:aws:aws-marketplace:::catalog/*catalog-name*/listing/*listing-id*
- arn:aws:aws-marketplace:::catalog/*catalog-name*/offer/*offer-id*
- arn:aws:aws-marketplace:::catalog/*catalog-name*/offerSet/*offerSet-id*

SearchListings, SearchFacets 및 ListPurchaseOptions 작업은 여러 항목에서 작동하므로 단일 리소스의 범위 조정을 지원하지 않습니다. 이러한 작업에는 다음 와일드카드 ARN 패턴을 사용합니다.

- SearchListings 및 SearchFacets의 경우: `arn:aws:aws-marketplace:::catalog/catalog-name/listing/*`
- ListPurchaseOptions의 경우: `arn:aws:aws-marketplace:::catalog/catalog-name/purchaseOption/*`

서비스 연결 역할

Discovery API는 서비스 연결 역할을 사용하지 않습니다. 모든 액세스는 표준 IAM 정책을 통해 제어됩니다.

크로스 계정 액세스

IAM 역할을 사용하여 Discovery API에 대한 교차 계정 액세스 권한을 부여할 수 있습니다. 적절한 Discovery API 권한을 사용하여 대상 계정에서 역할을 생성한 다음 소스 계정에서 역할을 수입합니다.

AWS Marketplace 보고 API에 대한 액세스 제어

AWS Marketplace 보고 API를 사용하여 조달 인사이트 대시보드를 가져옵니다. API는 [GetBuyerDashboard](#) 작업을 지원합니다.

API를 사용하려면 먼저 `AWSServiceRoleForProcurementInsightsPolicy` 서비스 연결 역할을 생성해야 합니다. 역할은 다음을 수행합니다.

- 구매자 조직의 모든 계정에 대한 데이터에 AWS Marketplace 액세스하고 설명할 수 있습니다.
- 조달 인사이트 대시보드 가져오기
- 위임된 관리자를 등록 및 등록 취소할 수 있습니다.

구매자는 AWS Marketplace 콘솔을 사용하여 조달 인사이트 대시보드에 대한 신뢰할 수 있는 액세스를 활성화할 때 역할을 생성합니다. 해당 프로세스에 대한 자세한 내용은 AWS Marketplace 구매자 안내서의 [대시보드 활성화](#)를 참조하세요.

Important

- API 또는 CLI를 사용하는 경우 대시보드에 대한 신뢰할 수 있는 액세스를 활성화하기 전에 서비스 연결 역할을 생성해야 합니다.
- 서비스 연결 역할 외에도 조직의 모든 기능을 활성화하고 관리자 계정에 속해야 합니다. 자세한 내용은 AWS Organizations 사용 설명서의 다음 주제를 참조하세요.

- [AWS Organizations를 사용하여 조직의 모든 기능 활성화](#)
- [자습서: 조직 생성 및 구성](#)
- [를 사용하여 관리 계정 관리 AWS Organizations](#)

[GetBuyerDashboard](#) 작업을 호출하고 위임된 관리자를 등록 및 등록 취소하려면 다음 IAM 권한이 `AWSServiceRoleForProcurementInsightsPolicy` 있어야 합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CreateServiceLinkedRoleForProcurementInsights",
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "arn:aws:iam::*:role/aws-service-role/procurement-insights.marketplace.amazonaws.com/AWSServiceRoleForProcurementInsights*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "procurement-insights.marketplace.amazonaws.com"
        }
      }
    },
    {
      "Sid": "EnableAWSServiceAccessForProcurementInsights",
      "Effect": "Allow",
      "Action": [
        "organizations:EnableAWSServiceAccess"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "organizations:ServicePrincipal": "procurement-insights.marketplace.amazonaws.com"
        }
      }
    },
    {
      "Sid": "ManageDelegatedAdministrators",
```

```

        "Effect": "Allow",
        "Action": [
            "organizations:ListDelegatedAdministrators",
            "organizations:DeregisterDelegatedAdministrator",
            "organizations:RegisterDelegatedAdministrator"
        ],
        "Resource": "*",
        "Condition": {
            "StringLike": {
                "organizations:ServicePrincipal": "procurement-
insights.marketplace.amazonaws.com"
            }
        }
    },
    {
        "Sid": "GetBuyerDashboardStatement",
        "Effect": "Allow",
        "Action": "aws-marketplace:GetBuyerDashboard",
        "Resource": "*"
    },
    {
        "Sid": "ViewOrganizationDetails",
        "Effect": "Allow",
        "Action": [
            "organizations:DescribeOrganization",
            "organizations:ListAWSServiceAccessForOrganization"
        ],
        "Resource": "*"
    }
]
}

```

정책 생성에 대한 자세한 내용은 IAM 사용 설명서 [의 AWS Identity and Access Management의 정책 및 권한을](#) 참조하세요.

정책이 사용하는 AWS Organizations 작업에 대한 자세한 내용은 [AWS Organizations API 참조](#)를 참조하세요.

AWS Marketplace API 할당량

다음 섹션에서는 서비스 할당량에 대한 정보를 제공합니다.

주제

- [AWS Marketplace 카탈로그 API의 서비스 할당량](#)
- [AWS Marketplace 계약 API에 대한 서비스 할당량](#)
- [AWS Marketplace 권한 부여 API에 대한 서비스 할당량](#)
- [AWS Marketplace 배포 API에 대한 서비스 할당량](#)
- [AWS Marketplace 측정 API에 대한 서비스 할당량](#)
- [AWS Marketplace Discovery API의 서비스 할당량](#)

AWS Marketplace 카탈로그 API의 서비스 할당량

AWS Marketplace 카탈로그 API에는 다음과 같은 할당량이 있습니다.

요청 할당량

API 작업	요청 속도(당 AWS 계정)
ListEntities	초당 10개
DescribeEntity	초당 20
StartChangeSet	초당 5
ListChangeSets	초당 5
DescribeChangeSet	초당 10개
CancelChangeSet	초당 5
TagResource	초당 5
UntagResource	초당 5
ListTagsForResource	초당 5
PutResourcePolicy	초당 5
GetResourcePolicy	초당 5
DeleteResourcePolicy	초당 5

제품 요금 차원 할당량

설명	제한
비공개 제안 또는 공개 제안에 포함될 수 있는 고유한 계약 차원(Entitled 차원이라고도 함)의 최대 수입니다. 제안의 계약 차원은 제품당 고유한 계약 차원의 최대 수에 포함됩니다.	200
제품당 고유한 계약 차원(Entitled 차원이라고도 함)의 최대 수입니다. 여기에는 비공개 제안과 공개 제안의 모든 고유한 계약 차원이 포함됩니다. 제한된 상태의 차원은이 제한에 포함됩니다.	200
비공개 제안 또는 공개 제안에 포함할 수 있는 고유한 차원(ExternallyMetered 차원이라고도 함)의 최대 수입니다. 제안의 이러한 사용량 차원은 제품당 최대 고유 사용량 차원 수에 포함됩니다.	200
제품당 고유 사용 차원(ExternallyMetered 차원이라고도 함)의 최대 수입니다. 여기에는 비공개 제안과 공개 제안의 모든 고유한 사용 차원이 포함됩니다. 제한된 상태의 차원은이 제한에 포함됩니다.	200

계정 할당량

할당량	설명
계정당 최대 미결 StartChangeSet 요청 수	250
계정당 동시에 생성되거나 업데이트된 최대 제안 수	20
계정당 동시에 생성되거나 업데이트된 최대 OfferSets 수	20

요청 기록 보존 할당량

설명	할당량
변경 요청의 보존 기간입니다. 이는 각 변경 요청의 종료 시간 이후에 적용됩니다.	90일

 Note

변경 요청과 엔터티는 다릅니다. 개체는 유형 또는 상태에 관계없이 영구적으로 존재합니다. 예를 들어 [판매자 제품](#), [제안](#), [재판매 권한](#) 부여 및 [프라이빗 마켓플레이스 경험](#) 엔터티는 초안 상태에 있더라도 삭제되지 않습니다.

AWS Marketplace 계약 API에 대한 서비스 할당량

AWS 계정에는 AWS Marketplace 계약 서비스와 관련된 다음과 같은 할당량이 있습니다.

요청 할당량

API 연산	요청 속도(계정당 AWS)
AcceptAgreementCancellationRequest	2/초
AcceptAgreementPaymentRequest	초당 5
AcceptAgreementRequest	초당 3개
BatchCreateBillingAdjustmentRequest	2/초
CancelAgreement	초당 3개
CancelAgreementCancellationRequest	2/초
CancelAgreementPaymentRequest	초당 5
CreateAgreementRequest	초당 4개

API 연산	요청 속도(계정당 AWS)
DescribeAgreement	초당 5
GetAgreementCancellationRequest	초당 5
GetAgreementEntitlements	초당 10개
GetAgreementPaymentRequest	초당 5
GetAgreementTerms	초당 10개
GetBillingAdjustmentRequest	초당 5
ListAgreementCancellationRequests	초당 5
ListAgreementCharges	초당 15개
ListAgreementInvoiceLineItems	초당 10개
ListAgreementPaymentRequests	초당 5
ListBillingAdjustmentRequests	초당 5
RejectAgreementCancellationRequest	2/초
RejectAgreementPaymentRequest	초당 5
SearchAgreements	초당 5
SendAgreementCancellationRequest	2/초
SendAgreementPaymentRequest	초당 5
UpdatePurchaseOrders	초당 3개

계약 할당량

설명	할당량
대상 제안에 지정할 수 있는 최대 계약 값(USD)입니다.	10억 달러(또는 요청된 통화로 이에 상응하는 금액)
일반 제안에 지정할 수 있는 최대 계약 값(USD)입니다.	1백만 달러(또는 요청된 통화로 이에 상응하는 금액)
계정당 리소스당 허용되는 최대 무료 평가판 수입니다.	1
계정당 리소스당 생성할 수 있는 최대 활성 계약 수입니다.	100

AWS Marketplace 권한 부여 API에 대한 서비스 할당량

AWS 계정에는 AWS Marketplace 권한 부여 서비스와 관련된 다음과 같은 할당량이 있습니다.

요청 할당량

API 작업	요청 속도(당 AWS 계정)	설명
GetEntitlements	초당 20	현재 리전의이 계정에서 초당 수행할 수 있는 최대 GetEntitlements 요청 수입니다.

AWS Marketplace 배포 API에 대한 서비스 할당량

AWS 계정에는 AWS Marketplace 배포 서비스와 관련된 다음과 같은 할당량이 있습니다.

요청 할당량

할당량	기본값	설명
Deployment parameter	10	구매자 및 제품 조합당 최대 배포 파라미터 수

할당량	기본값	설명
Deployment parameter update frequency	100	24시간당 배포 파라미터를 업데이트할 수 있는 최대 횟수
PutDeploymentParameter throttle	1	초당 수행할 수 있는 최대 PutDeploymentParameter 요청 수
ListTagsForResource throttle	20	초당 수행할 수 있는 최대 ListTagsForResource 요청 수
TagResource throttle	20	초당 수행할 수 있는 최대 TagResource 요청 수
UntagResource throttle	20	초당 수행할 수 있는 최대 UntagResource 요청 수
Deployment parameter SecretString name length	15,000	배포 파라미터 SecretString 문자열의 최대 문자 수
Deployment parameter name length	400	배포 파라미터 이름의 최대 문자 수
TagList	50	배포 파라미터 요청당 최대 태그 수
ClientToken	64	ClientToken 문자열의 최대 문자 수

AWS Marketplace 측정 API에 대한 서비스 할당량

AWS 계정에는 AWS Marketplace 측정 서비스와 관련된 다음과 같은 할당량이 있습니다.

요청 할당량

API 작업	요청 속도(당 AWS 계정)	설명
BatchMeterUsage	초당 10개	현재 리전의이 계정에서 초당 수행할 수 있는 최대 BatchMeterUsage 요청 수입니다.
MeterUsage	초당 10개	현재 리전의이 계정에서 초당 수행할 수 있는 최대 MeterUsage 요청 수입니다.
RegisterUsage	초당 5	현재 리전의이 계정에서 초당 수행할 수 있는 최대 RegisterUsage 요청 수입니다.

AWS Marketplace Discovery API의 서비스 할당량

AWS 계정에는 AWS Marketplace Discovery API와 관련된 다음과 같은 할당량이 있습니다.

요청 할당량

API 연산	요청 속도(AWS 계정당)
GetListing	초당 5
GetProduct	초당 5
GetOffer	초당 5
GetOfferTerms	초당 5
GetOfferSet	초당 5
ListPurchaseOptions	초당 5
ListFulfillmentOptions	초당 5

API 연산	요청 속도(AWS 계정당)
SearchListings	초당 10개
SearchFacets	초당 10개

AWS Marketplace API에 대한 로깅

다음 섹션에서는 로깅에 대한 정보를 제공합니다.

주제

- [CloudTrail을 사용하여 AWS Marketplace 카탈로그 API 호출 로깅](#)
- [CloudTrail을 사용하여 AWS Marketplace 계약 API 호출 로깅](#)
- [CloudTrail을 사용하여 AWS Marketplace 측정 API 호출 로깅](#)
- [CloudTrail을 사용하여 AWS Marketplace 배포 API 호출 로깅](#)
- [를 사용하여 AWS Marketplace Discovery API 호출 로깅 AWS CloudTrail](#)

CloudTrail을 사용하여 AWS Marketplace 카탈로그 API 호출 로깅

AWS Marketplace 카탈로그 API는 사용자 AWS CloudTrail, 역할 또는가 수행한 작업에 대한 레코드를 제공하는 서비스와 통합됩니다 AWS 서비스. CloudTrail은의 호출을 포함하여 카탈로그 API에 대한 모든 호출을 이벤트로 캡처합니다 AWS Marketplace Management Portal.

추적을 생성하는 경우 CloudTrail 이벤트를 Amazon Simple Storage Service(Amazon S3) 버킷으로 지속적으로 전송할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

AWS Marketplace CloudTrail의 카탈로그 API 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. AWS Marketplace Catalog API에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 이벤트와 함께 CloudTrail AWS 서비스 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은AWS CloudTrail 사용 설명서에서 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

이 이벤트를 지속적으로 기록하려면 추적을 AWS 계정생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든

AWS 리전에 트레일이 적용됩니다. 추적은 AWS 파티션 AWS 리전 의 모든에서 이벤트를 로깅하고 지정한 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취 AWS 서비스 하도록 다른를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- [트레일 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에서 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 수신](#)
- [여러 계정에서 CloudTrail 로그 파일 수신](#)

모든 AWS Marketplace 카탈로그 API 작업은 CloudTrail에서 로깅되며 API 참조에 설명되어 있습니다. 예를 들어 StartChangeSet, DescribeChangeSet 및 ListChangeSets API 작업에 대한 호출은 CloudTrail 로그 파일에 항목을 생성합니다. 모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에게 관한 정보가 포함됩니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부

자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail userIdentity 요소](#)를 참조하세요.

AWS Marketplace 카탈로그 로그 파일 항목 이해

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함하고 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출에 대한 순서 지정된 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

Note

이들 예제는 서식을 조정하여 가독성을 높인 것입니다. CloudTrail 로그 파일에서는 모든 항목 및 이벤트가 한 줄로 연결되어 있습니다. 또한 이 예제는 단일 AWS Marketplace 카탈로그 API 항목으로 제한되었습니다. 실제 CloudTrail 로그 파일에는 여러의 항목과 이벤트가 표시됩니다 AWS 서비스.

다음 예제에서는 ListEntities 작업을 보여주는 AWS Marketplace 카탈로그 API 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": "2019-10-17T21:49:23Z",
    "eventSource": "marketplacecatalog.amazonaws.com",
    "eventName": "ListEntities",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "PostmanRuntime/7.18.0",
    "requestParameters": {
      "catalog": "AWSMarketplace",
      "entityType": "EntityProduct",
      "sort": {
        "sortBy": "LastUpdateTimeInMillis",
        "sortOrder": "DESC"
      },
      "maxResults": 20
    },
    "responseElements": null,
    "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789010"
  }
]
```

CloudTrail을 사용하여 AWS Marketplace계약 API 호출 로깅

계약 API는 사용자AWS CloudTrail, 역할 또는 AWS 서비스에서 수행한 작업에 대한 레코드를 제공하는 서비스와 통합됩니다AWS Marketplace. CloudTrail은 계약 API에 대한 API 호출을 이벤트로 캡처합

니다. 캡처되는 호출에는 계약 API를 활용하는 AWS Marketplace 웹 사이트, 콘솔 및 기타 인터페이스의 호출과 계약 API 작업에 대한 직접 코드 호출이 포함됩니다.

추적을 생성하면 계약 API에 대한 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 계약 API에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용자 안내서](#)를 참조하세요.

CloudTrail의 계약 API 정보

CloudTrail은 계정을 생성할 AWS 계정때에서 활성화됩니다. 계약 API에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 이벤트와 함께 CloudTrail AWS 서비스이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다AWS 계정. 자세한 내용은 AWS CloudTrail 사용 설명서에서 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

계약 API에 대한 이벤트를 AWS 계정포함하여에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든 AWS 리전에 트레일이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취AWS 서비스하도록 다른를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기](#) 및 [여러 계정에서 CloudTrail 로그 파일 받기](#)

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부

자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail userIdentity 요소](#)를 참조하세요.

계약 API 작업 이해

계약 API는 서비스형 소프트웨어(SaaS), 서버(컨테이너 포함) 및 전문 서비스 제품을 구매하는 데 사용됩니다. AWS Marketplace. 또한에 대한 결과 계약 또는 구독을 관리하는 데에도 사용됩니다. AWS Marketplace.

Note

AWS Data Exchange 제품 구매는 CloudTrail에서 기록하지 않습니다.

`AcceptAgreementRequest` 작업은의 AWS Identity and Access Management(IAM) 사용자 또는 역할이 해당 제품을 AWS 계정구매할 때 사용됩니다. AWS Marketplace. 마찬가지로 `CancelAgreement` 작업은 IAM 사용자 또는 역할이 계약 또는 구독을 취소할 때 사용됩니다. 계약 API에서 CloudTrail 로그를 모니터링하여 구매자는 AWS 계정에서 발생하는 가장 중요한 구매 관련 작업을 모니터링할 수 있습니다. AWS Marketplace.

이 `DescribeAgreement` 작업은 고객이 특정 계약에 대한 메타 데이터를 특별히 볼 때 사용됩니다. `GetAgreementTerms` 작업은 특정 계약의 조건을 볼 때 사용됩니다. `SearchAgreements` 작업은 IAM 사용자 또는 역할이 모든 계약의 전체 목록에서 계약의 하위 집합을 나열하거나 필터링할 때 사용됩니다.

Note

`AcceptAgreementRequest` 및 `CancelAgreement` 작업은 구매자가 사용할 수 있지만 판매자는 사용할 수 없습니다. 그러나 구매자와 판매자 모두 `DescribeAgreement`, `GetAgreementTerms`, 및 `SearchAgreements` 작업을 사용할 수 있습니다.

구매자는 CloudTrail 로그에서 계약의 계약 ID를 식별할 수도 있습니다. 계약에 대한 자세한 내용을 보려면 AWS Marketplace 콘솔에서 구독 관리 탭을 선택합니다. 여기서 계약 ID는 세부 정보 보기에 제공됩니다. 계약 ID는 `AcceptAgreementRequest` API 작업의 `responseElements` 경우에서, API 작업의 `requestParameters` 경우에서 찾을 수 `CancelAgreement` 있습니다.

계약 API 로그 파일 항목 이해

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스로부터의 단일 요청을 나타

내며 요청 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보가 들어 있습니다. CloudTrail 로그 파일은 퍼블릭 API 호출의 순서가 지정된 스택 추적이 아니므로 특정 순서로 표시되지 않습니다.

다음은 AcceptAgreementRequest 작업을 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP123"
  },
  "eventTime": "2023-08-11T17:13:50Z",
  "eventSource": "agreement-marketplace.amazonaws.com",
  "eventName": "AcceptAgreementRequest",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Coral/Netty4",
  "requestParameters": {
    "agreementRequestId": "ar-6xbrddjzym594imkrrezrn5wa"
  },
  "responseElements": {
    "agreementId": "agmt-1lnrq6riwpg2tczhv378zknlc"
  },
  "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
  "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789010",
  "eventCategory": "Management"
}
```

다음은 CancelAgreement 작업을 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
```

```

    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP1234"
  },
  "eventTime": "2023-08-14T03:11:42Z",
  "eventSource": "agreement-marketplace.amazonaws.com",
  "eventName": "CancelAgreement",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Coral/Netty4",
  "requestParameters": {
    "agreementId": "agmt-enitbfqjebjmwmomzrucf032t"
  },
  "responseElements": null,
  "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
  "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789010",
  "eventCategory": "Management"
}

```

다음은 DescribeAgreement 작업을 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudtrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP123",
  },
  "eventTime": "2023-10-30T22:45:24Z",
  "eventSource": "agreement-marketplace.amazonaws.com",
  "eventName": "DescribeAgreement",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36",
  "requestParameters": {
    "agreementId": "agmt-6wy4rhp7l6iyuu2jrcgd1shdi"
  },
}

```

```

    "responseElements": null,
    "requestID": "bEXAMPLE-347f-4c07-9645-cd2EXAMPLE61",
    "eventID": "dEXAMPLE-d891-42a5-8da6-1cdEXAMPLE34",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789010",
    "eventCategory": "Management",
  }

```

다음은 GetAgreementTerms 작업을 설명하는 CloudTrail 로그 항목을 보여 주는 예시입니다.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudtrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP123",
  },
  "eventTime": "2023-10-30T22:48:37Z",
  "eventSource": "agreement-marketplace.amazonaws.com",
  "eventName": "GetAgreementTerms",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36",
  "requestParameters": {
    "agreementId": "agmt-6wy4rhp7l6iyuu2jrcgd1shdi"
  },
  "responseElements": null,
  "requestID": "eEXAMPLE-fc57-4127-bbda-bc1EXAMPLE03",
  "eventID": "bEXAMPLE-5345-4634-8b58-925EXAMPLE3e",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789010",
  "eventCategory": "Management",
}

```

다음은 SearchAgreements 작업을 보여주는 CloudTrail 로그 항목이 나타난 예시입니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudtrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP123",
  },
  "eventTime": "2023-10-30T18:41:10Z",
  "eventSource": "agreement-marketplace.amazonaws.com",
  "eventName": "SearchAgreements",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "127.0.0.1",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/117.0.0.0 Safari/537.36",
  "requestParameters": {
    "catalog": "AWSMarketplace",
    "filters": [
      {
        "name": "PartyType",
        "values": [
          "Proposer"
        ]
      },
      {
        "name": "ResourceType",
        "values": [
          "SaaSProduct"
        ]
      },
      {
        "name": "Status",
        "values": [
          "ACTIVE"
        ]
      },
      {
        "name": "AgreementType",
        "values": [
          "PurchaseAgreement"
        ]
      }
    ]
  }
}
```

```

    ],
    "maxResults": 5
  },
  "responseElements": null,
  "requestID": "fEXAMPLE-0aa6-4e42-8715-6a1EXAMPLE95",
  "eventID": "0EXAMPLE-8ce8-4814-bcf1-636EXAMPLEb5",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789010",
  "eventCategory": "Management",
}

```

CloudTrail을 사용하여 AWS Marketplace 측정 API 호출 로깅

측정 API는 사용자 AWS CloudTrail, 역할 또는에서 수행한 작업의 레코드를 제공하는 서비스와 통합됩니다 AWS 서비스 AWS Marketplace. CloudTrail은 측정 API에 대한 API 호출을 이벤트로 캡처합니다. 캡처되는 호출에는 측정 API를 활용하는 AWS Marketplace 웹 사이트, 콘솔 및 기타 인터페이스의 호출과 측정 API 작업에 대한 직접 코드 호출이 포함됩니다.

추적을 생성하면 측정 API에 대한 이벤트를 포함하여 CloudTrail 이벤트를 지속적으로 Amazon S3 버킷에 배포할 수 있습니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 측정 API에 수행된 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

CloudTrail에 대한 자세한 내용은 [AWS CloudTrail 사용자 안내서](#)를 참조하세요.

CloudTrail에서 API 정보 측정

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. 측정 API에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 이벤트와 함께 CloudTrail AWS 서비스 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 AWS CloudTrail 사용 설명서에서 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

측정 API에 대한 이벤트를 AWS 계정에 포함하여 이벤트를 지속적으로 기록하려면 추적을 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든 AWS 리전에 트레일이 적용됩니다. 추적은 AWS 파티션에 있는 모든 리전의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취 AWS 서비스 하도록 다른 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하세요.

- [추적 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에 대한 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 받기](#) 및 [여러 계정에서 CloudTrail 로그 파일 받기](#)

SaaS 기반 제품에 대한 BatchMeterUsage 작업, 컨테이너 기반 제품에 대한 RegisterUsage 작업, Amazon Machine Image(AMI) 기반 및 컨테이너 기반 제품에 대한 MeterUsage 작업은 CloudTrail에서 로깅합니다.

모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에 관한 정보가 포함됩니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부

자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail userIdentity 요소](#)를 참조하세요.

측정 API 로그 파일 항목 이해

예: SaaS 기반 제품의 **BatchMeterUsage** 경우

다음 예제는 AWS Marketplace 측정 API의 BatchMeterUsage 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다. 판매자가 나열된 서비스형 소프트웨어(SaaS) 제품에 대한 [고객의 사용을 보고하기 위해 측정 레코드를 보내](#)면 AWS Marketplace이 CloudTrail 로그 항목이 판매자의에 기록됩니다 AWS 계정.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::123456789012:user/*****",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "userName": "*****"
  },
  "eventTime": "2018-04-19T16:32:51Z",
  "eventSource": "metering-marketplace.amazonaws.com",
```

```

    "eventName": "BatchMeterUsage",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "*****",
    "userAgent": "Coral/Netty14",
    "requestParameters": {
      "usageRecords": [
        {
          "dimension": "Dimension1",
          "timestamp": "Apr 19, 2018 4:32:50 PM",
          "customerIdentifier": "customer1",
          "customerAWSAccountID": "987654321098",
          "quantity": 1
        }
      ],
      "productCode": "EXAMPLE_proCode"
    },
    "responseElements": {
      "results": [
        {
          "usageRecord": {
            "dimension": "Dimension1",
            "timestamp": "Apr 19, 2018 4:32:50 PM",
            "customerIdentifier": "customer1",
            "customerAWSAccountID": "987654321098",
            "quantity": 1
          },
          "meteringRecordId": "bEXAMPLE-98f0-4e90-8bd2-bf0EXAMPLE1e",
          "status": "Success"
        }
      ],
      "unprocessedRecords": [ ]
    },
    "requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
    "eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
  }
}

```

예: Amazon EKS에 배포된 컨테이너 기반 제품의 **RegisterUsage** 경우

다음 예제는 AWS Marketplace 측정 API의 RegisterUsage 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다. 의 시간당 가격 컨테이너 제품이 구매자의 AWS Marketplace 배포되면 컨테이너

의 AWS 계정소프트웨어는 구매자의 RegisterUsage 내에서를 호출 AWS 계정 하여 해당 Amazon Elastic Container Service(Amazon ECS) 작업 또는 Amazon Elastic Kubernetes Service(Amazon EKS) 포드에 대한 시간당 측정을 시작합니다. 이 CloudTrail 로그 항목은 구매자의 AWS 계정에 로깅 됩니다.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EX_PRINCIPAL_ID:botocore-session-1111111111",
    "arn": "arn:aws:sts::123456789012:assumed-role/Alice/botocore-session-1111111111",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Alice",
        "accountId": "123456789012",
        "userName": "Alice"
      },
      "webIdFederationData": {
        "federatedProvider": "arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-east-1.amazonaws.com/id/EXAMPLEFA1C58F08CDB049167EXAMPLE",
        "attributes": {}
      },
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-07-23T02:19:34Z"
      }
    }
  },
  "eventTime": "2020-07-23T02:19:46Z",
  "eventSource": "metering-marketplace.amazonaws.com",
  "eventName": "RegisterUsage",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "1.2.3.4",
  "userAgent": "aws-cli/1.18.103 Python/3.8.2 Linux/4.14.181-142.260.amzn2.x86_64 botocore/1.17.26",
  "requestParameters": {
    "productCode": "EXAMPLE_proCode",
    "publicKeyVersion": 1
  }
}
```



```

    },
    "responseElements": {
      "signature": "eyJhbGciOiJIUzI1Ni..."
    },
    "requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
    "eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
  }
}

```

예: Amazon EKS에 배포된 컨테이너 기반 제품의 **MeterUsage** 경우

다음 예제는 Amazon EKS의 컨테이너에 대한 AWS Marketplace 측정 API의 MeterUsage 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다. 의 [사용자 지정 측정](#) AWS Marketplace 기능이 있는 컨테이너 제품이 구매자의에 배포되면 컨테이너의 AWS 계정소프트웨어가 구매자의 MeterUsage 내에서를 호출 AWS 계정 하여 매시간 보고합니다. 이 CloudTrail 로그 항목은 구매자의 AWS 계정에 로깅됩니다.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EX_PRINCIPAL_ID:botocore-session-1111111111",
    "arn": "arn:aws:sts::123456789012:assumed-role/Alice/botocore-session-1111111111",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Alice",
        "accountId": "123456789012",
        "userName": "Alice"
      },
      "webIdFederationData": {
        "federatedProvider": "arn:aws:iam::123456789012:oidc-provider/oidc.eks.us-east-1.amazonaws.com/id/EXAMPLEFA1C58F08CDB049167EXAMPLE",
        "attributes": {}
      },
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-07-23T01:03:26Z"
      }
    }
  }
}

```

```

    }
  }
},
"eventTime": "2020-07-23T01:38:13Z",
"eventSource": "metering-marketplace.amazonaws.com",
"eventName": "MeterUsage",
"awsRegion": "us-east-1",
"sourceIPAddress": "1.2.3.4",
"userAgent": "aws-cli/1.18.103 Python/3.8.2 Linux/4.14.181-142.260.amzn2.x86_64
botocore/1.17.26",
"requestParameters": {
  "timestamp": "Jul 23, 2020 1:35:44 AM",
  "usageQuantity": 1,
  "usageDimension": "Dimension1",
  "productCode": "EXAMPLE_proCode"
},
"responseElements": {
  "meteringRecordId": "bEXAMPLE-98f0-4e90-8bd2-bf0EXAMPLE1e"
},
"requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
"eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}

```

예: AMI 기반 제품의 **MeterUsage** 경우

다음 예제는 Amazon Machine Image(AMIs)용 AWS Marketplace 측정 API의 MeterUsage 작업을 보여주는 CloudTrail 로그 항목을 보여줍니다. 에서 사용자 지정 측정을 사용하는 AMI 제품이 구매자의 에 AWS Marketplace 배포되면 AMI의 AWS 계정소프트웨어는 구매자의 MeterUsage 내에서 호출 AWS 계정 하여 매시간 사용량을 보고합니다. 이 CloudTrail 로그 항목은 구매자의 AWS 계정에 로그인됩니다.

```

{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "EX_PRINCIPAL_ID:i-exampled859aa775c",
    "arn": "arn:aws:sts::123456789012:assumed-role/Alice/i-exampled859aa775c",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "sessionContext": {
      "sessionIssuer": {

```

```

        "type": "Role",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:role/Alice",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "webIdFederationData": {},
    "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2020-07-10T23:05:20Z"
    },
    "ec2RoleDelivery": "1.0"
}
},
"eventTime": "2020-07-10T23:06:42Z",
"eventSource": "metering-marketplace.amazonaws.com",
"eventName": "MeterUsage",
"awsRegion": "us-east-1",
"sourceIPAddress": "1.2.3.4",
"userAgent": "aws-cli/1.16.102 Python/2.7.16 Linux/4.14.133-113.112.amzn2.x86_64
botocore/1.12.92",
"requestParameters": {
    "productCode": "EXAMPLE_proCode",
    "timestamp": "Jul 10, 2020 11:06:41 PM",
    "usageDimension": "Dimension1",
    "usageQuantity": 1,
    "dryRun": false
},
"responseElements": {
    "meteringRecordId": "bEXAMPLE-98f0-4e90-8bd2-bf0EXAMPLE1e"
},
"requestID": "dEXAMPLE-251d-11e7-8d11-1f3EXAMPLE8b",
"eventID": "cEXAMPLE-e6c2-465d-b47f-150EXAMPLE97",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}

```

CloudTrail을 사용하여 AWS Marketplace 배포 API 호출 로깅

AWS Marketplace 배포 서비스는 사용자 AWS CloudTrail, 역할 또는가 수행한 작업의 레코드를 제공하는 서비스와 통합됩니다 AWS 서비스.

추적을 생성하는 경우 CloudTrail 이벤트를 Amazon Simple Storage Service(Amazon S3) 버킷으로 지속적으로 전송할 수 있습니다. 추적을 구성하지 않은 경우에도 이벤트 기록에서 CloudTrail 콘솔의 최신 이벤트를 볼 수 있습니다. CloudTrail에서 수집한 정보를 사용하여 요청, 요청이 수행된 IP 주소, 요청을 수행한 사람, 요청이 수행된 시간 및 추가 세부 정보를 확인할 수 있습니다.

AWS Marketplace CloudTrail의 배포 서비스 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. AWS Marketplace 배포 서비스에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 이벤트와 함께 CloudTrail AWS 서비스 이벤트에 기록됩니다. 에서 최근 이벤트를 보고 검색하고 다운로드할 수 있습니다 AWS 계정. 자세한 내용은 AWS CloudTrail 사용 설명서에서 [CloudTrail 이벤트 기록을 사용하여 이벤트 보기](#)를 참조하세요.

이 이벤트를 지속적으로 기록하려면 추적을 AWS 계정 생성합니다. CloudTrail은 추적을 사용하여 Amazon S3 버킷으로 로그 파일을 전송할 수 있습니다. 콘솔에서 트레일을 생성하면 기본적으로 모든 AWS 리전에 트레일이 적용됩니다. 추적은 AWS 파티션 AWS 리전 의 모든에서 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한 CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 조치를 취 AWS 서비스 하도록 다른를 구성할 수 있습니다. 자세한 내용은 다음을 참조하세요.

- [트레일 생성 개요](#)
- [CloudTrail 지원 서비스 및 통합](#)
- [CloudTrail에서 Amazon SNS 알림 구성](#)
- [여러 리전에서 CloudTrail 로그 파일 수신](#)
- [여러 계정에서 CloudTrail 로그 파일 수신](#)

모든 배포 서비스 작업은 CloudTrail에서 로깅되며 API 참조에 문서화됩니다. 예를 들어 PutDeploymentParameterAPI 작업을 호출하면 CloudTrail 로그 파일에 항목이 생성됩니다. 모든 이벤트 또는 로그 항목에는 요청을 생성했던 사용자에게 관한 정보가 포함됩니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 사용자 자격 증명으로 했는지 여부
- 역할 또는 페더레이션 사용자에게 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지 여부

자세한 내용은 AWS CloudTrail 사용 설명서의 [CloudTrail userIdentity 요소](#)를 참조하세요.

AWS Marketplace 로그 파일 항목 이해

트레일이란 지정한 S3 버킷에 이벤트를 로그 파일로 입력할 수 있게 하는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함될 수 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜와 시간, 요청 파라미터 등에 대한 정보를 포함하고 있습니다. CloudTrail 로그 파일은 퍼블릭 API 직접 호출에 대한 순서 지정된 스택 추적이 아니기 때문에 특정 순서로 표시되지 않습니다.

Note

이들 예제는 서식을 조정하여 가독성을 높인 것입니다. CloudTrail 로그 파일에서는 모든 항목 및 이벤트가 한 줄로 연결되어 있습니다. 또한 이 예제는 단일 배포 서비스 항목으로 제한되었습니다. 실제 CloudTrail 로그 파일에는 여러의 항목과 이벤트가 표시됩니다 AWS 서비스.

다음 예제에서는 PutDeploymentParameter 작업을 보여주는 배포 서비스 로그 항목을 보여줍니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHJKLMNOP12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHJKLMNOP123"
  },
  "eventTime": "2023-11-16T16:32:48Z",
  "eventSource": "deployment-marketplace.amazonaws.com",
  "eventName": "PutDeploymentParameter",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
  "userAgent": "aws-sdk-java/2.20.162 Mac_OS_X/13.5.2 OpenJDK_64-Bit_Server_VM/18.0.1+10-FR Java/18.0.1 vendor/Amazon.com_Inc. io/sync http/URLConnection cfg/retry-mode/legacy",
  "requestParameters": {
    "productId": "prod-fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "catalog": "AWSMarketplace",
    "clientToken": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "agreementId": "agmt-fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "deploymentParameter": {
      "name": "PutDeploymentParameterCloudTrailTest-secret",
      "secretString": "****"
    }
  }
}
```

```

    },
    "expirationDate": "2023-11-30T03:02:26.779241Z"
  },
  "responseElements": {
    "agreementId": "agmt-fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "deploymentParametersId": "dp-fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "resourceArn": "arn:aws:aws-marketplace:us-east-1:123456789010:DeploymentParameter:catalogs/AWSMarketplace/products/prod-fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1/dp-fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1"
  },
  "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
  "eventID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789010",
  "eventCategory": "Management"
}

```

를 사용하여 AWS Marketplace Discovery API 호출 로깅 AWS CloudTrail

AWS Marketplace Discovery API는 사용자 AWS CloudTrail, 역할 또는가 수행한 작업의 레코드를 제공하는 서비스와 통합됩니다 AWS 서비스. CloudTrail은 AWS SDK 및 AWS CLI의 호출을 포함하여 모든 Discovery API 호출을 이벤트로 캡처합니다.

CloudTrail의 Discovery API 정보

CloudTrail은 계정을 생성할 AWS 계정 때에서 활성화됩니다. Discovery API에서 활동이 발생하면 해당 활동이 이벤트 기록의 다른 이벤트와 함께 CloudTrail AWS 서비스 이벤트에 기록됩니다.

모든 Discovery API 작업은 CloudTrail에서 로깅됩니다. 다음 작업은 CloudTrail 로그 파일에 항목을 생성합니다.

- GetListing
- GetProduct
- GetOffer
- GetOfferTerms
- GetOfferSet
- ListPurchaseOptions

- ListFulfillmentOptions
- SearchFacets
- SearchListings

Discovery API 이벤트의 이벤트 소스는 `discovery-marketplace.amazonaws.com`.

Discovery API 로그 항목의 예

다음은 GetListing 작업을 보여주는 CloudTrail 로그 항목이 나타낸 예시입니다.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/ExampleUser",
    "accountId": "123456789012",
    "accessKeyId": "EXAMPLE",
    "userName": "ExampleUser"
  },
  "eventTime": "2026-04-01T15:30:00Z",
  "eventSource": "discovery-marketplace.amazonaws.com",
  "eventName": "GetListing",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.50",
  "userAgent": "aws-sdk-python/1.34.0 Python/3.12.0",
  "requestParameters": {
    "listingId": "listing-saas-abc123"
  },
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "readOnly": true,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "123456789012",
  "eventCategory": "Management"
}
```

AWS Marketplace API에 대한 알림

다음 섹션에서는 알림에 대한 정보를 제공합니다.

주제

- [AWS Marketplace 카탈로그 API 이벤트에 대한 Amazon EventBridge 알림](#)

AWS Marketplace 카탈로그 API 이벤트에 대한 Amazon EventBridge 알림

AWS Marketplace 는 이전에 Amazon CloudWatch Events라고 했던 Amazon EventBridge와 통합됩니다. EventBridge는 애플리케이션을 다양한 소스의 데이터와 연결하는 데 사용할 수 있는 이벤트 버스 서비스입니다.

판매자, 채널 파트너 및 프라이빗 마켓플레이스 관리자가 EventBridge를 사용하여 ChangeSet 상태 이벤트를 수신하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [변경 세트 이벤트를 참조](#)하세요.

AWS Marketplace MCP 서버 정보

AWS Marketplace 모델 컨텍스트 프로토콜(MCP)은 AWS Marketplace 검색, 평가, 연구 및 제안 생성을 위한 AI 기반 도구를 제공합니다. MCP는 AI 어시스턴트가 다음을 통해 외부 시스템과 상호 작용할 수 있도록 하는 표준화된 프로토콜입니다.

- 도구: AI 모델이 작업을 수행하기 위해 호출할 수 있는 함수입니다.
- 리소스: AI 모델과 공유할 수 있는 데이터 및 컨텍스트입니다.

모든 도구는 상태 비저장입니다. 각 호출은 서버 측 대화 상태 없이 독립적입니다.

사용 사례 예

다음은 AWS Marketplace MCP 서버를 사용하는 일반적인 방법입니다.

- 자연어 쿼리를 사용하여 비즈니스 요구 사항을 해결하는 제품을 찾습니다.
- AI 생성 비교 보고서를 사용하여 여러 공급업체를 빠르게 평가합니다.
- 비즈니스 요구 사항에 맞는 포괄적인 구매 제안을 생성합니다.
- 경쟁 솔루션을 side-by-side 분석합니다.

엔드포인트: <https://marketplace-mcp.us-east-1.api.aws/mcp>

프로토콜: 스트리밍 가능한 HTTP를 통한 MCP(JSON-RPC 2.0). 모든 응답은 단일 JSON 페이로드입니다.

인증: 필요하지 않습니다.

주제

- [사용 가능한 도구](#)
- [통합 가이드](#)
- [도구 세부 정보](#)
- [워크플로](#)
- [종합 예제](#)
- [오류 처리](#)

사용 가능한 도구

6가지 도구 모두 상태 비저장 도구이므로 대화 컨텍스트나 세션 관리가 필요하지 않습니다. 전체 입력/출력 스키마를 포함한 모든 도구 정의는 표준 MCP tools/list 엔드포인트를 통해 프로그래밍 방식으로 사용할 수 있습니다. 에서 반환되는 스키마는 파라미터 이름, 유형 및 제약 조건에 대한 정식 최신 정보 `tools/list`입니다.

다음 표에는 각 도구와 읽기 전용인지 여부가 요약되어 있습니다.

도구 이름	설명	읽기 전용
get_aws_marketplace_report_guidelines	권장 사항, 비교 또는 평가에 대한 워크플로 지침과 보고서 형식 템플릿을 반환합니다. 먼저를 호출하여 보고서 워크플로를 확인합니다.	예
search_aws_marketplace_solutions	쿼리별로 AWS Marketplace 카탈로그를 검색합니다. 배치 쿼리(최대 10개) 및 커서 기반 페이지 매김을 지원합니다.	예
get_aws_marketplace_solution	솔루션 IDs(평가, 요금, 감정, 순위, CTAs)별로 자세한 솔루션 메타데이터를 가져옵니다.	예
get_aws_marketplace_related_solutions	지정된 솔루션 IDs 대한 관련 솔루션을 관계 점수별로 순위가 매겨집니다.	예
research_aws_marketplace_solution	솔루션(기능, 리뷰, 제한 사항, 경쟁자, 요금, 통합, 사례 연구)에 대한 심층 웹 조사를 수행합니다. 기본 웹 검색이 있는 클라이언트에서 숨겨집니다.	예
submit_aws_marketplace_feedback	보고서 품질에 대한 사용자 피드백(긍정적/부정적)을 제출합니다.	아니요

통합 가이드

코드 없는 Claude Connector 또는 프로그래밍 방식 AWS Marketplace MCP 클라이언트를 통해 MCP 서버와 통합할 수 있습니다.

Claude 커넥터(권장)

AWS Marketplace MCP 도구를 사용하는 가장 빠른 방법은 AWS Marketplace Claude용 커넥터를 사용하는 것입니다. 코딩이나 MCP 클라이언트 설정이 필요하지 않음 - Claude는 도구 검색 및 호출을 자동으로 처리합니다.

설정:

1. Claude [AWS Marketplace 커넥터](#) 디렉터리에서 커넥터를 엽니다.
2. 연결을 선택하여 활성화합니다.
3. Claude 대화에서 + 버튼(왼쪽 하단)을 선택하거나 /를 입력하여 메뉴를 엽니다.
4. 커넥터 위로 마우스를 가져가서를 클릭합니다AWS Marketplace.
5. Claude에게 AWS Marketplace 제품을 검색, 비교 또는 평가하도록 요청합니다. 자동으로 MCP 도구를 사용합니다.

커넥터 작동 방식에 대한 자세한 내용은 [커넥터를 사용하여 Claude의 기능 확장을 참조하세요](#).

프롬프트 예제:

- “AWS Marketplace 중견기업에 가장 적합한 SIEM 솔루션 찾기”
- "Datadog과 New Relic 비교" AWS Marketplace
- "Terraform Cloud에 대한 평가 보고서 작성" AWS Marketplace

MCP 클라이언트 통합

프로그래밍 방식 통합 또는 사용자 지정 에이전트 프레임워크의 경우 MCP 엔드포인트에 직접 연결합니다.

사전 조건

시작하려면 다음이 필요합니다.

1. MCP 호환 에이전트 프레임워크 또는 클라이언트.
2. JSON-RPC 2.0 요청을 수행할 수 있는 HTTP 클라이언트입니다.

1단계: MCP 클라이언트 설정

HTTP 전송을 지원하는 표준 MCP 클라이언트를 사용합니다.

엔드포인트:

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

지원되는 메서드는 다음과 같습니다.

- POST with method: "initialize"- 세션을 초기화하고를 가져옵니다Mcp-Session-Id.
- POST를 사용하여 method: "tools/list"- 사용 가능한 도구와 스키마를 검색합니다.
- POST with method: "tools/call"- 도구를 호출합니다.

인증: 인증이 필요하지 않습니다. HTTP 요청에서 인증 헤더를 생략합니다.

2단계: 도구 가져오기 및 등록(도구 검색)

모든 도구 정의는 표준 MCP tools/list 메서드를 통해 프로그래밍 방식으로 사용할 수 있습니다. 도구 스키마는이 설명서와 독립적으로 업데이트될 수 있으므로 현재 입력/출력 스키마를 가져오는 것이 좋습니다. 초기화 중예를 tools/list 한 번 호출하고 에이전트의 도구 매니페스트 또는 시스템 프롬프트에 반환된 도구를 등록합니다.

다음 예제에서는 tools/list 요청을 보여줍니다.

요청:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/list",
  "params": {}
}
```

응답(약어):

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

```

"result": {
  "tools": [
    {
      "name": "get_aws_marketplace_report_guidelines",
      "description": "...",
      "inputSchema": { ... }
    },
    {
      "name": "search_aws_marketplace_solutions",
      "description": "...",
      "inputSchema": { ... }
    },
    {
      "name": "get_aws_marketplace_solution",
      "description": "...",
      "inputSchema": { ... }
    },
    {
      "name": "get_aws_marketplace_related_solutions",
      "description": "...",
      "inputSchema": { ... }
    },
    {
      "name": "research_aws_marketplace_solution",
      "description": "...",
      "inputSchema": { ... }
    },
    {
      "name": "submit_aws_marketplace_feedback",
      "description": "...",
      "inputSchema": { ... }
    }
  ]
}
}

```

각 도구 항목에는 도구 name, 사용 지침 및 예제description가 자세히 설명된 , 파라미터 유형, 제약 조건 및 검증 규칙이 inputSchema 포함된다 포함됩니다.

3단계: 원격 측정 구성

AWS Marketplace MCP에는 요청 상관관계 및 원격 측정을 위한 모든 도구 호출에 안정적인 식별자가 필요합니다. 이는 인증 자격 증명이 아니며 진단 및 분석에만 사용됩니다.

사용자 지정 HTTP 헤더에 식별자를 전달합니다.

```
Mcp-Session-Id: <stable-id>
```

- 브라우저 클라이언트: 기존 세션 쿠키 또는 프런트엔드 생성 UUID에서 파생합니다. 모든 도구 호출에서 동일한 값을 사용합니다.
- 프로그래밍 방식 클라이언트:와 POST 함께 method: "initialize"를 사용하여 가져옵니다Mcp-Session-Id.

HTTP 전송

AWS Marketplace MCP는 서버가 여러 클라이언트 연결을 처리할 수 있는 독립 HTTP 서비스로 작동하는 HTTP 전송을 사용합니다.

다음은 주요 전송 특성입니다.

- 요청에 대한 HTTP POST: 모든 클라이언트 메시지는 새 HTTP POST 요청으로 전송됩니다.
- JSON 응답: 모든 도구는 JSON 응답()을 반환합니다Content-Type: application/json.
- 세션 관리: 요청 상관관계 및 원격 측정을 위한 Mcp-Session-Id 헤더를 통한 세션 IDs.
- 프로토콜 버전: MCP-Protocol-Version: 2025-06-18 헤더를 통해 지정됩니다.

필수 HTTP 헤더:

```
Content-Type: application/json
Mcp-Session-Id: <stable-uuid>
MCP-Protocol-Version: 2025-06-18
```

다음 예제에서는 도구 호출 요청 및 응답을 보여줍니다.

요청 예제:

```
POST /mcp HTTP/1.1
Content-Type: application/json
Mcp-Session-Id: 1868a90c-5b12-4e8a-9f3d-2a1b3c4d5e6f
MCP-Protocol-Version: 2025-06-18

{
```

```

"jsonrpc": "2.0",
"id": 1,
"method": "tools/call",
"params": {
  "name": "search_aws_marketplace_solutions",
  "arguments": {
    "queries": ["SIEM solutions"]
  }
}
}

```

응답 예제:

```

{
  "jsonrpc": "2.0",
  "id": 1,
  "result": {
    "content": [
      {
        "type": "text",
        "text": "{\"results\":[...],\"next_cursor\":null}"
      }
    ],
    "isError": false
  }
}

```

도구 세부 정보

이 섹션에서는 각 도구의 입력 파라미터, 출력 형식 및 사용 지침을 설명합니다.

get_aws_marketplace_report_guidelines

지정된 보고서 유형에 대한 워크플로 지침과 출력 형식 템플릿을 반환합니다. 권장 사항, 비교 또는 평가 워크플로를 보려면 먼저 호출합니다.

이 도구는 클라이언트 기능에 따라 다른 프롬프트를 반환합니다. 기본 웹 검색이 있는 클라이언트는 `research_aws_marketplace_solution` 도구 참조 대신 임베디드 검색 쿼리가 있는 프롬프트를 가져옵니다.

입력:

```
{
  "workflow_type": "recommendations" | "comparison" | "evaluation"
}
```

다음 표에서는 입력 파라미터를 설명합니다.

Field	유형	필수	설명
workflow_type	enum	예	"recommendations" 순위가 매겨진 제품 검색의 경우, side-by-side 분석 "comparison" 의 경우, 단일 제품 평가의 "evaluation" 경우.

출력:

```
{
  "workflow_type": "recommendations",
  "guidelines": "# AWS Marketplace Recommendations Workflow\n\n..."
}
```

다음 표에서는 출력 필드를 설명합니다.

Field	유형	설명
workflow_type	문자열	요청된 워크플로 유형입니다.
guidelines	문자열	워크플로 지침과 마크다운 형식 템플릿을 결합했습니다.

각 워크플로를 사용해야 하는 경우:

- 권장 사항 - "모니터링 도구 찾기", "CI/CD 솔루션이 필요합니다", "월 5,000 USD 미만의 SOC2 준수 보안 스캔".
- 비교 - "Datadog 대 New Relic 비교", "로그 관리를 위한 Splunk 대 Elastic".
- 평가 - "Terraform Cloud에 대한 평가 보고서 작성", "내 CTO에 New Relic 구매 정당화".

search_aws_marketplace_solutions

직접 AWS Marketplace 카탈로그 API 검색을 수행합니다. AI 처리 또는 관련성 점수 없이 순위가 지정되지 않은 필터링되지 않은 원시 결과를 반환합니다. 단일 쿼리에 대해 배치 쿼리(최대 10개) 및 커서 기반 페이지 매김을 지원합니다.

입력:

```
{
  "queries": ["Drata compliance", "Vanta compliance"],
  "max_results": 5,
  "cursor": null
}
```

다음 표에서는 입력 파라미터를 설명합니다.

Field	유형	필수	설명
queries	string[]	예	검색 쿼리 배열(1~10). 또한 레거시query(단일 문자열, 배열로 자동 변환됨)를 허용합니다.
max_results	number	아니요	쿼리당 최대 결과 수(1~100, 기본값: 10).
cursor	문자열	No	페이지 매김 커서(단일 쿼리에만 작동하며 배치에는 무시됨).

출력:

```
{
  "results": [
    {
      "solution_id": "prodview-abc123",
      "solution_name": "Drata",
      "vendor_name": "Drata Inc",
      "vendor_url": "https://aws.amazon.com/marketplace/seller-profile?id=...",
      "solution_description": "Continuous compliance automation...",
      "solution_url": "https://aws.amazon.com/marketplace/pp/prodview-abc123",
      "reviews_summary": {
```

```

      "reviews_count": 150,
      "average_rating": 4.7
    }
  ],
  "next_cursor": "eyJvZmZzZXQiOjEwLCJwYXd1U2l6ZSI6MTB9"
}

```

다음 표에서는 출력 필드를 설명합니다.

Field	유형	설명
results	MinimalSolutionMetadata[]	일치하는 솔루션의 배열입니다.
next_cursor or	문자열 null	더 이상 결과가 없는 null 경우 다음 페이지의 페이지 매김 커서(단일 쿼리만 해당).

배치 동작: 여러 쿼리의 경우 결과에는 의해 결합 및 중복 제거됩니다 solution_id. 배치 요청에 대해서는 페이지 매김 커서가 반환되지 않습니다.

페이지 매김: next_cursor이 반환되면(단일 쿼리 요청만 해당) cursor 파라미터로 다시 전달하여 추가 결과를 검색합니다. next_cursor가 이면 일치하는 null모든 솔루션이 검색된 것입니다.

get_aws_marketplace_solution

직접 AWS Marketplace 카탈로그 API 호출을 수행합니다. AI 처리 없이 솔루션 ID별로 포괄적인 솔루션 메타데이터를 검색합니다. 배치 요청(최대 10IDs)을 지원합니다.

입력:

```

{
  "solution_ids": ["prodview-abc123", "prodview-xyz789"]
}

```

다음 표에서는 입력 파라미터를 설명합니다.

Field	유형	필수	설명
solution_ids	string[]	예	솔루션 IDs 배열(1~10). IDs 로 시작합니다prodview- . 또한 레거시solution_id (단일 문자열, 자동 변환됨)를 허용합니다.

출력:

```
{
  "solutions": [
    {
      "solution_id": "prodview-abc123",
      "solution_name": "Datadog",
      "solution_description": "Comprehensive monitoring and analytics...",
      "solution_url": "https://aws.amazon.com/marketplace/pp/prodview-abc123?ref_=...",
      "vendor_name": "Datadog, Inc.",
      "vendor_url": "https://aws.amazon.com/marketplace/seller-profile?id=...&ref_=...",
      "highlights": ["Real-time monitoring", "700+ integrations"],
      "reviews_summary": {
        "reviews_count": 245,
        "average_rating": 4.6
      },
      "review_sentiments": [
        {
          "sentiment_category": "functionality",
          "overall_polarity": "positive",
          "positive_score": "85%",
          "negative_score": "5%",
          "mixed_score": "10%",
          "sample_reviews": [
            {
              "review_text": "Excellent monitoring capabilities...",
              "rating": 5,
              "source_name": "G2"
            }
          ]
        }
      ]
    },
    {
      "free_trial_available": true,
      "pricing_options": [
```

```

    {
      "name": "Infrastructure Monitoring",
      "description": "$15 per host per month"
    }
  ],
  "fulfillment_options_types": ["SaaS"],
  "categories": ["Monitoring & Observability"],
  "security_certificates": ["SOC 2 Type II", "ISO 27001"],
  "rankings": [{ "category": "Monitoring", "rank": 3 }],
  "has_standard_contract": true,
  "call_to_action": {
    "procurement_url": "https://...",
    "request_for_demo_url": "https://...",
    "request_for_private_offer_url": "https://...",
    "procure_free_trial_url": "https://..."
  }
}
]
}

```

다음 표에서는 출력 필드를 설명합니다.

Field	유형	설명
solutions	SolutionOutput[]	솔루션 세부 정보 배열(단일 ID의 경우에도 항상 배열).

다음 표에서는 각 SolutionOutput 객체의 필드를 설명합니다.

Field	유형	설명
solution_id	문자열	AWS Marketplace 솔루션 ID입니다.
solution_name	문자열	제품 이름입니다.
solution_description	문자열	길거나 짧은 설명입니다.
solution_url	문자열	참조 태그가 있는 Marketplace URL입니다.
vendor_name	문자열	공급업체 표시 이름입니다.

Field	유형	설명
vendor_url	문자열	참조 태그가 있는 공급업체 프로필 URL입니다.
highlights	string[]	제품 하이라이트.
reviews_summary	객체	{ reviews_count, average_rating }
review_sentiments	ReviewSentiment[]	극성 점수 및 샘플 리뷰가 포함된 범주별 감성(기능, 사용 편의성, 고객 서비스, 비용 효율성).
free_trial_available	부울?	무료 평가판을 사용할 수 있는지 여부입니다.
pricing_options	PricingOption[]	{ name, description } 페어.
fulfillment_options_types	string[]?	SaaS, 컨테이너 또는 AMI와 같은 이행 유형.
categories	string[]?	"모니터링 및 관찰성"과 같은 범주.
security_certificates	string[]?	"SOC 2 Type II" 또는 "ISO 27001"과 같은 인증.
rankings	순위[]?	{ category, rank } - 범주 내 순위(1~100).
has_standard_contract	부울?	제품이 AWS 표준 계약을 지원하는지 여부입니다.
call_to_action	CallToAction	{ procurement_url, request_for_demo_url, request_for_private_offer_url, procure_free_trial_url }

get_aws_marketplace_related_solutions

직접 AWS Marketplace 카탈로그 API 호출을 수행합니다. AI 분석 또는 관련성 점수 없이 관계 점수(내림차순)별로 AWS Marketplace 정렬된 지정된 솔루션 IDs에 대한 관련 솔루션을 반환합니다. 배치 요청(최대 10IDs)을 지원합니다.

입력:

```
{
  "solution_ids": ["prodview-abc123"]
}
```

다음 표에서는 입력 파라미터를 설명합니다.

Field	유형	필수	설명
solution_ids	string[]	예	솔루션 IDs 배열(1~10). 또한 레거시 solution_id (단일 문자열, 자동 변환됨)를 허용합니다.

출력:

```
{
  "results": [
    {
      "solution_id": "prodview-abc123",
      "related_solutions": [
        {
          "solution_id": "prodview-xyz789",
          "solution_name": "New Relic",
          "vendor_name": "New Relic, Inc.",
          "vendor_url": "https://aws.amazon.com/marketplace/seller-profile?id=...",
          "solution_description": "Full-stack observability platform...",
          "solution_url": "https://aws.amazon.com/marketplace/pp/prodview-xyz789"
        }
      ]
    }
  ]
}
```

다음 표에서는 출력 필드를 설명합니다.

Field	유형	설명
results	SingleRelatedSolutionsResult[]	입력 솔루션 ID당 하나의 항목입니다.

Field	유형	설명
<code>results[].solution_id</code>	문자열	입력 솔루션 ID입니다.
<code>results[].related_solutions</code>	MinimalSolutionMetadata[]	관련성을 기준으로 정렬된 관련 솔루션.

research_aws_marketplace_solution

AWS Marketplace 솔루션에 대한 심층 웹 조사를 수행합니다. 요청된 각 섹션에 대한 웹 검색을 실행하고 인라인 인용 링크가 있는 콘텐츠를 반환합니다. 배치 요청(최대 5IDs)을 지원합니다.

가시성:이 도구는 기본 웹 검색 기능이 있는 클라이언트(예: Anthropic/ClaudeAI)에서 숨겨집니다. 이러한 클라이언트는 대신 임베디드 검색 쿼리가 포함된 워크플로 프롬프트를 가져옵니다.

입력:

```
{
  "solution_ids": ["prodview-abc123", "prodview-xyz789"],
  "sections": ["features", "integrations"]
}
```

다음 표에서는 입력 파라미터를 설명합니다.

Field	유형	필수	설명
<code>solution_ids</code>	string[]	예	prodview-* 형식(1~5)의 솔루션 IDs.
<code>sections</code>	ResearchSection[]	아니요	조사할 섹션입니다. 기본값: 7개 섹션 모두.

다음 표에서는 사용 가능한 각 연구 섹션을 설명합니다.

섹션	반환되는 항목
features	공식 설명서, 기능, 사양
reviews	G2, 가트너, TrustRadius 고객 피드백
limitations	알려진 문제, 단점, 비판
competitors	경쟁 포지셔닝, 대안
pricing	비용 정보, 요금 모델, 티어
integrations	AWS 에코시스템, APIs, 타사 도구
case_studies	고객 성공 사례, 추천, ROI

출력:

```
{
  "results": [
    {
      "solution_id": "prodview-abc123",
      "solution_name": "Datadog",
      "research": {
        "features": {
          "content": "Datadog provides 700+ integrations...",
          "citations": [
            {
              "index": "1",
              "title": "Datadog Integrations",
              "url": "https://docs.datadoghq.com/integrations/"
            }
          ]
        }
      }
    }
  ]
}
```

다음 표에서는 출력 필드를 설명합니다.

Field	유형	설명
results	SingleResearchResult[]	입력 솔루션 ID당 하나의 항목입니다.
results[].solution_id	문자열	조사된 솔루션 ID입니다.
results[].solution_name	문자열	솔루션 ID에서 확인된 제품 이름입니다.
results[].research	객체	요청된 각 섹션에 대한 키가 있는 객체입니다.
results[].research[section].content	문자열	인라인 인용 링크를 사용하여 콘텐츠를 조사합니다.
results[].research[section].citations	인용[]	{ index, title, url } 각 인용에 대해

동시성: 솔루션은 병렬로 연구됩니다. 각 솔루션 내의 섹션은 다운스트림 서비스가 압도되지 않도록 순차적입니다.

submit_aws_marketplace_feedback

AWS Marketplace 보고서에 대한 사용자 피드백을 제출합니다. 권장 사항, 비교 또는 평가 워크플로를 완료한 후이 도구를 호출합니다.

입력:

```
{
  "sentiment": "negative",
  "categories": ["incomplete", "other"],
  "feedback_text": "Missing pricing details"
}
```

다음 표에서는 입력 파라미터를 설명합니다.

Field	유형	필수	설명
sentiment	enum	예	"positive" 또는 "negative"
categories	string[]	아니요	부정적인 피드백의 경우: "irrelevant" , "incomplete" , "inaccurate" , "other".
feedback_text	문자열	No	자유 형식 텍스트(최대 1,000자).

출력:

```
{
  "success": true,
  "message": "Thank you for your feedback. We'll use it to improve."
}
```

워크플로

AWS Marketplace MCP 서버는 제품 연구 및 직접 카탈로그 조회를 위한 세 가지 기본 워크플로를 지원합니다.

보고서 생성(권장 사항, 비교, 평가)

이는 AI 지원 제품 연구를 위한 기본 워크플로입니다. 를 오케스트레이션 진입점 `get_aws_marketplace_report_guidelines`으로 사용합니다.

흐름:

1. 적절한 `get_aws_marketplace_report_guidelines`를 사용하여 호출합니다 `다workflow_type`.
2. 도구는 step-by-step 워크플로 지침과 출력 형식 템플릿을 반환합니다.
3. AI 어시스턴트는 반환된 지침을 따르며, 일반적으로 카탈로그 API 및 연구 도구를 호출하여 데이터를 수집합니다.
4. 어시스턴트는 출력 템플릿에 따라 형식이 지정된 보고서를 생성합니다.
5. 보고서를 전송한 후 어시스턴트는를 통해 사용자 피드백을 수집합니다 `다submit_aws_marketplace_feedback`.

권장 사항 워크플로:

1. `get_aws_marketplace_report_guidelines({ workflow_type: "recommendations" })`
2. `search_aws_marketplace_solutions({ queries: ["Product A", "Product B"], max_results: 3 })`
3. 형식 템플릿에 따라 보고서를 작성합니다.
4. `submit_aws_marketplace_feedback({ sentiment: "positive" })`

비교 워크플로:

1. `get_aws_marketplace_report_guidelines({ workflow_type: "comparison" })`
2. `get_aws_marketplace_solution({ solution_ids: ["prodview-abc", "prodview-xyz"] })`
3. `research_aws_marketplace_solution({ solution_ids: [...], sections: ["features", "integrations"] })`
4. 형식 템플릿에 따라 비교 매트릭스를 빌드하고 보고서를 작성합니다.
5. `submit_aws_marketplace_feedback({ sentiment: "positive" })`

평가 워크플로:

1. `get_aws_marketplace_report_guidelines({ workflow_type: "evaluation" })`
2. `get_aws_marketplace_solution({ solution_ids: ["prodview-target"] })`
3. `research_aws_marketplace_solution({ solution_ids: ["prodview-target"], sections: ["features", "reviews", "limitations", "case_studies"] })`
4. 형식 템플릿에 따라 평가 보고서를 작성합니다.
5. `submit_aws_marketplace_feedback({ sentiment: "positive" })`

직접 카탈로그 조회

AI 분석이 필요하지 않은 프로그래밍 방식 통합 또는 빠른 조회의 경우 없이 카탈로그 API 도구를 직접 호출할 수 있습니다 `get_aws_marketplace_report_guidelines`.

검색 후 세부 정보 조회:

1. 하나 이상의 키워드 쿼리 `search_aws_marketplace_solutions`를 사용하여 호출합니다.
2. 반환된 `solution_id` 값을 사용하여 전체 세부 정보를 `get_aws_marketplace_solution` 호출합니다.
3. 선택적으로 호출 `get_aws_marketplace_related_solutions`하여 관련 제품을 검색합니다.

피드백 수집

워크플로를 완료한 후 어시스턴트는 사용자 피드백을 수집해야 합니다.

1. 응답이 도움이 되었는지 사용자에게 질문합니다.
2. 양수인 경우 `submit_aws_marketplace_feedback`를 사용하여 호출합니다 `sentiment: "positive"`.
3. 음수인 경우 해당하는 범주(`irrelevant`, `incomplete`, `inaccurate`, `other`)를 묻고 선택적으로 자유 텍스트 피드백을 수집하여 제출합니다.

종합 예제

다음 예제에서는 초기 쿼리에서 피드백 제출까지 전체 권장 사항 워크플로를 보여줍니다.

1단계 - 워크플로 지침 가져오기:

```
{
  "jsonrpc": "2.0",
  "id": 1,
  "method": "tools/call",
  "params": {
    "name": "get_aws_marketplace_report_guidelines",
    "arguments": {
      "workflow_type": "recommendations"
    }
  }
}
```

2단계 - 후보 솔루션 검색:

```
{
  "jsonrpc": "2.0",
  "id": 2,
```

```

"method": "tools/call",
"params": {
  "name": "search_aws_marketplace_solutions",
  "arguments": {
    "queries": ["SIEM security monitoring", "log management SIEM"],
    "max_results": 10
  }
}
}

```

3단계 - 상위 후보에 대한 세부 메타데이터 검색:

```

{
  "jsonrpc": "2.0",
  "id": 3,
  "method": "tools/call",
  "params": {
    "name": "get_aws_marketplace_solution",
    "arguments": {
      "solution_ids": ["prodview-abc123", "prodview-def456", "prodview-ghi789"]
    }
  }
}

```

4단계 - 사용자 피드백 제출:

```

{
  "jsonrpc": "2.0",
  "id": 4,
  "method": "tools/call",
  "params": {
    "name": "submit_aws_marketplace_feedback",
    "arguments": {
      "sentiment": "positive",
      "feedback_text": "Recommendations were relevant and well-structured"
    }
  }
}

```

오류 처리

오류는 프로토콜 오류, 도구 실행 오류, 검증 오류의 세 가지 범주로 나뉩니다.

프로토콜 오류

요청이 JSON-RPC 또는 MCP 사양(예: 잘못된 JSON, 누락된 필수 필드 또는 알 수 없는 방법)을 위반하면 프로토콜 오류가 반환됩니다.

```
{
  "jsonrpc": "2.0",
  "id": 10,
  "error": {
    "code": -32602,
    "message": "Invalid params",
    "data": {
      "details": "Missing required field: queries"
    }
  }
}
```

도구 실행 오류

런타임에 도구 호출이 실패하면(예: 속도 제한, 잘못된 솔루션 IDs 또는 업스트림 서비스 오류) 도구 실행 오류가 반환됩니다. 도구 오류는 MCP 프로토콜 수준 오류가 아닌 결과 객체 내에 보고되므로 AI 어시스턴트는 오류를 보고 처리할 수 있습니다.

```
{
  "jsonrpc": "2.0",
  "id": 11,
  "result": {
    "content": [
      {
        "type": "text",
        "text": "Failed to search listings: Rate limit exceeded"
      }
    ],
    "isError": true
  }
}
```

유효성 검사 오류

입력이 스키마 검증에 실패하면 검증 오류가 반환됩니다(예: 잘못된 솔루션 ID 형식, 최대 배열 길이 초과 또는 필수 필드 누락). 이는에서 도구 실행 오류로 표시됩니다 `isError: true`.

AWS Marketplace API를 판매자로 사용

다음 섹션에서는 API를 판매자로 사용하는 방법에 대한 정보를 제공합니다.

주제

- [판매자 제품 작업](#)
- [AWS Marketplace APIs를 사용하여 제안 작업](#)
- [다중 제품 솔루션 작업](#)
- [카탈로그 데이터를 사용하여 인터페이스 구축](#)

판매자 제품 작업

를 사용하여 판매자 제품 작업을 위한 작업을 자동화AWS Marketplace Catalog API할 수 있습니다. 여기에는 제품을 생성, 업데이트, 보기, 나열 및 정렬하는 기능이 포함됩니다. 이를 통해 제품 관리를 자동화할 수 있습니다. 예를 들어에서 셀프 서비스 게시 기능을 제공할 수 있습니다AWS Marketplace Management Portal.

제품은 판매하려는 단위 또는 리소스로AWS Marketplace, 종종 기본 제품이라고 합니다. 구매자는 제품 정보, 배포 속성 및 결제 정보를 추가할 때까지 기본 제품을 사용할 수 없습니다.

제품은 판매하려는 목록의 제품 정보, 소프트웨어 배포 속성 및 결제 메커니즘을 설명합니다. 판매할 수 있고 구매자가 사용할 수 있는 거래 가능한 단위가 되려면 제품을 제안과 페어링해야 합니다AWS Marketplace.

를 사용하여 다음을 수행할 수도 있습니다AWS Marketplace Catalog API.

- [AWS MarketplaceAPIs를 사용하여 비공개 제안 작업](#)
- [AWS MarketplaceAPIs를 사용하여 재판매 권한 부여 작업](#)
- [AWS MarketplaceAPIs를 사용하여 채널 파트너 비공개 제안 작업](#)

각 제품 유형에는 서로 다른 제품 개체가 있습니다. 엔터티는 제품 또는 제안일 수 있습니다AWS Marketplace. 지원되는 제품 유형 및 개체는 다음과 같습니다.

제품 유형	개체
Amazon Machine Image(AMI) 제품	AmiProduct@1.0

제품 유형	개체
컨테이너 제품	ContainerProduct@1.0
서비스형 소프트웨어(SaaS) 제품	SaaSProduct@1.0
기계 학습(ML) 제품	MachineLearningProduct@1.0

Note

CloudFormation 제품 유형, AWS Data Exchange 데이터 제품 및 전문 서비스 제품이 포함된 단일 AMI는 지원되지 않습니다.

다음 주제에서는에 설명된 대로 API에 액세스할 수 있고 판매자 사전 조건을 완료했다고 가정합니다. [AWS Marketplace 카탈로그 API에 대한 액세스 제어](#).

다음 리소스를 참조하세요.

- AWS Marketplace 카탈로그 API 사용의 기본 사항을 이해하려면 섹션을 참조하세요. [AWS Marketplace 카탈로그 API 사용](#).
- 작업 코드 예제가 포함된 end-to-end 랩은 AWS Marketplace 판매자 워크숍의 [API로 제품 관리를](#) 참조하세요.
- API 요청의 코드 예제는 GitHub의 AWS 샘플에서 [Python](#) 및 [Java](#) 예제를 참조하세요.

다음 주제에서는 카탈로그 API를 사용하여 단일 AMI 제품, 컨테이너 기반 제품 또는 SaaS 제품에 대한 작업을 수행하는 방법을 설명합니다.

주제

- [제품 생성](#)
- [제품 세부 정보 업데이트](#)
- [요금 차원 추가](#)
- [요금 차원 업데이트](#)
- [요금 차원 제한](#)
- [대상 지정 구성 업데이트](#)

- [제품 표시 여부 업데이트](#)
- [제품 게시](#)
- [제품 ID 찾기](#)
- [설정 상태 및 오류 변경](#)
- [AMI 기반 제품 작업](#)
- [EC2 Image Builder 구성 요소 제품 작업](#)
- [AWS Marketplace APIs를 사용하여 컨테이너 기반 제품 작업](#)
- [AWS Marketplace APIs를 사용하여 SaaS 제품 작업](#)
- [AWS Marketplace APIs 사용하여 기계 학습 제품 작업](#)

제품 생성

Note

이 변경 유형은 AWS Marketplace 카탈로그에 새로운 제품 개체를 생성하려는 경우에만 필요합니다. 기존 제품을 업데이트할 때는 필요하지 않습니다.

카탈로그 API를 사용하여 식별자(제품 코드 및 제품 ID)가 포함된 AMI, 컨테이너, 기계 학습 또는 SaaS 제품 문서를 생성할 수 있습니다 AWS Marketplace.

CreateProduct 변경 유형으로 StartChangeSet API 작업을 호출하여 Draft 상태의 제품을 생성합니다.

요청이 성공적으로 처리되면 Draft 상태의 제품을 AWS Marketplace Catalog API 생성합니다. 불완전한 제품이며 구매자가 볼 수 없습니다 AWS Marketplace.

그런 다음 Update 변경 유형을 사용하여 제품 생성 프로세스인 [UpdateInformation](#), [UpdateDimensions](#), [UpdateTargeting](#) 및 [UpdateVisibility](#)를 완료합니다.

제품이 완료되면 [ReleaseProduct](#) 변경 유형을 사용하여 제품 생성 프로세스를 완료한 다음 제안을 릴리스할 수 있습니다. 이 프로세스는 전체 제품을 검증하고 제품을 Limited 상태로 이동합니다.

Note

를 사용하여 제품을 생성하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 다음 주제를 AWS Marketplace Management Portal 참조하세요.

- [단일 AMI 제품 생성](#)

버전의 AMI는 업데이트할 수 없습니다. AMI를 업데이트해야 하는 경우 대신 새 버전을 생성합니다.

- [컨테이너 제품 생성](#)

- [SaaS 제품 생성](#)

- [기계 학습 제품 생성](#)

AWS Marketplace Management Portal를 사용하여 제품을 생성하는 경우 제품은 Staging 상태가 됩니다.

Draft 상태의 제품을 생성하려면 다음 예제와 같이 CreateProduct 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0" // choose from ["AmiProduct@1.0",
        "ContainerProduct@1.0", "SaaSProduct@1.0", "MachineLearningProduct@1.0"]
      },
      "DetailsDocument": {
        "ProductTitle": "Test product title set in CreateProduct"
      }
    }
  ]
}
```

CreateProduct 변경 유형을 추가할 필드에 대한 정보를 제공합니다. 이 변경 유형은 UpdateInformation 변경 유형으로 전송된 것과 동일한 제한이 적용되는 ProductTitle 속성에 포함될 수 있습니다.

- Entity (객체)(필수) - 생성 중인 명명된 객체 유형입니다.
- Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다 SaaSProduct@1.0, AmiProduct@1.0, ContainerProduct@1.0, MachineLearningProduct@1.0 또는 . 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument (객체)(필수) - 비어 있을 수 있습니다.
- ProductTitle (선택 사항) - 제품의 제목이며 최대 길이는 72자입니다. 나중에 UpdateInformation 변경 유형을 통해 제품 제목을 설정하거나 업데이트할 수도 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

요청이 완료되면(가 Status인 경우SUCCEEDED) 새 ProductId가 생성됩니다.

동기 검증

다음 스키마 검증은의 CreateProduct 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
ProductTitle(문자열)	최대 길이: 72	400

비동기 오류

다음 오류는 CreateProduct 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_INPUT	ProductTitle 필드에 부적절한 콘텐츠 '{InappropriateContent}'가 있습니다. ProductTitle에 부적절한 콘텐츠를 제공하지 않습니다.

제품 세부 정보 업데이트

이미 제품이 있는 경우 카탈로그 API를 사용하여 AMI, AWS Marketplace, 컨테이너, ML 또는 SaaS 제품의 제품 세부 정보를 업데이트할 수 있습니다.

Note

를 사용하여 제품 세부 정보를 업데이트하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 다음 주제를 AWS Marketplace Management Portal 참조하세요.

- AMI 기반 제품: [제품 정보 업데이트](#)
- 컨테이너 기반 제품: [컨테이너 제품에 대한 제품 정보 생성 또는 업데이트](#)
- SaaS 기반 제품: [제품 정보 업데이트](#)
- 기계 학습 제품: [제품 정보 업데이트](#)

제품 세부 정보를 업데이트하려면 다음 예제와 같이 UpdateInformation 변경 유형과 변경하려는 세부 정보를 사용하여 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
```

```
[
  {
    "ChangeType": "UpdateInformation",
    "Entity":
    {
      "Identifier": "prod-example12345",
      "Type": "AmiProduct@1.0"
    },
    "DetailsDocument":
    {
      "ProductTitle": "My Product Title",
      "ShortDescription": "My product short description.",
      "LongDescription": "My product longer description.",
      "Sku": "123example456",
      "LogoUrl": "https://awsmp-
logos.s3.amazonaws.com/ca60b754fe05a24257176cdbf31c4e0d",
      "VideoUrls":
      [
        "https://example.com/my-video"
      ],
      "Highlights":
      [
        "123example45"
      ],
      "AdditionalResources":
      [
        {
          "Text": "123example456",
          "Url": "https://example.com/some-link"
        }
      ],
      "SupportDescription": "Need help? Contact our experts at support@example.com \n
\nYour purchase includes 24x7 support.",
      "Categories":
      [
        "Operating Systems",
        "Network Infrastructure",
        "Application Development"
      ],
      "SearchKeywords":
      [
        "123example456"
      ]
    }
  }
]
```

```

    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateInformation 변경 유형을 추가합니다.

- Entity (객체)(필수) - 생성 중인 명명된 엔터티 유형입니다.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다. AmiProduct@1.0, ContainerProduct@1.0MachineLearningProduct@1.0, 또는 SaaSProduct@1.0.
- DetailsDocument (객체)(필수) - 제품에 대해 업데이트하려는 정보를 포함한 요청의 세부 정보입니다. 각 필드는 선택 사항이지만 업데이트할 변경 사항을 하나 이상 포함해야 합니다.
 - ProductTitle (문자열) - 구매자에게 표시할 제품의 이름입니다.
 - ShortDescription (문자열) - 구매자에게 표시할 제품의 주요 측면에 대한 설명입니다. 이는 일반적으로 2~3개의 문장입니다.
 - LongDescription (문자열) - 구매자에게 표시할 제품에 대한 더 긴 설명입니다. 이는 일반적으로 1~3개의 단락입니다.
 - Sku (문자열 또는 null) - 사용자가 직접 사용하기 위한 참조로 정의하는 자유 형식 문자열입니다. null를 사용하여이 필드의 설정을 해제합니다.
 - LogoUrl (문자열) - 공개적으로 액세스할 수 있는 Amazon Simple Storage Service(Amazon S3) 버킷의 이미지에 대한 URL입니다. 자세한 내용은 [회사 및 제품 로고 요구 사항](#)을 참조하세요.
 - VideoUrls (문자열 배열) - 제품 정보에서 구매자에게 참조로 제공할 공개적으로 사용 가능한 외부 호스팅 비디오의 URLs 목록입니다.

Note

현재는 배열에서 하나의 URL을 AWS Marketplace지원합니다.

- Highlights (문자열 배열) - 주요 제품 기능에 대한 짧은 콜아웃 목록입니다.
- AdditionalResources (구조 배열) - 제품에 대해 알아볼 수 있는 추가 리소스에 대한 참조 목록입니다. 각 참조는 텍스트 이름과 URL로 구성됩니다.
 - Text (문자열) - 리소스의 이름 또는 제목입니다.
 - Url (문자열) - 구매자가 제품을 이해하는 데 도움이 될 수 있는 리소스의 URL입니다.
- SupportDescription (문자열) - 제품에 대한 지원 제공에 대한 세부 정보입니다.

- **Categories** (문자열 배열) - 제품을 설명하는 정의된 제품 범주 목록AWS Marketplace입니다. 자세한 내용은 AWS Marketplace 구매자 안내서의 [제품 범주](#)를 참조하세요.
- **SearchKeywords** (문자열 배열) - 검색 환경을 개선하기 위한 제품의 키워드 목록입니다. 판매자 이름, 제품 이름 및 제품 범주는 검색 키워드에 자동으로 포함되므로 여기에서 반복할 필요가 없습니다.

Note

Draft 제품에 대한 제품 정보(메타데이터)를 처음 채울 때는 UpdateInformation 변경 유형의 DetailsDocument 객체에 ProductTitle, ShortDescription, LongDescription, LogoUrl, HighlightsAdditionalResources, SupportDescription, 및 Categories를 모두 제공해야 합니다SearchKeywords. CreateProduct 변경 유형 중에 이미 제공된 경우를 생략할 ProductTitle 수 있습니다. 그러나 제품의 기존 필드를 업데이트할 때 변경 유형의 DetailsDocument 객체에 UpdateInformation 변경해야 하는 속성만 포함할 수 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다.

요청 상태를 확인하려면를 사용하거나 [DescribeChangeSet](#) API를 AWS Marketplace Management Portal호출합니다.

동기 검증

다음 스키마 검증은의 UpdateInformation 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
문자열(일반)	제어 문자 "\x00-\x08\x0B-\x1F" 없음	400
ProductTitle(문자열)	최대 길이: 72 필수	400
ShortDescription(문자열)	최대 길이: 1000 필수	400
LongDescription(문자열)	최대 길이: 5000 필수	400
Sku(문자열)	최대 길이: 100 선택 사항	400
LogoUrl(문자열)	URL 패턴: ^https://(www\.)?[-a-zA-Z0-9@.]{1,256}\.[a-zA-Z0-9()]{2,63}\b([-a-zA-Z0-9@+./]*) 필수	400
VideoUrls(문자열 배열)	URL 패턴: https://(www\.)?[-a-zA-Z0-9@._]{1,256}\.[a-zA-Z0-9()]{2,63}\b([-a-zA-Z0-9@_+.V] 선택 사항	400
강조 표시(문자열 배열)	필수: 최소 1~최대 3	400
AdditionalResources(구조 배열)	최대 길이: 500	400

입력 필드	검증 규칙	HTTP 코드
	선택 사항	
SupportDescription(문자열)	최대 길이: 2000 필수	400
범주(문자열 배열)	최소 1~최대 3 필수	400
SearchKeywords(문자열 배열)	최소 1~최대 15 각 항목에 대해 최대 50자 필수	400

비동기 오류

다음 오류는 UpdateInformation 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
누락_데이터	업데이트를 수행하기 위해 제공된 데이터가 없습니다. 제품의 하나 이상의 필드에 대한 데이터를 제공합니다.
INVALID_INPUT	LogoUrl을 제공합니다.
INVALID_INPUT	ProductTitle을 제공합니다.
INVALID_INPUT	ShortDescription을 제공합니다.
INVALID_INPUT	LongDescription을 제공합니다.
INVALID_INPUT	SupportDescription을 제공합니다.

오류 코드	오류 메시지
INVALID_INPUT	하나 이상의 검색 키워드를 제공합니다.
INVALID_INPUT	하나 이상의 강조 표시를 제공합니다.
INVALID_INPUT	1~3개의 제품 범주를 제공합니다.
INVALID_INPUT	ProductTitle 필드에 부적절한 콘텐츠 '{InappropriateContent}'가 있습니다. ProductTitle에 부적절한 콘텐츠를 제공하지 않습니다.
INVALID_INPUT	ShortDescription 필드에 부적절한 콘텐츠 '{InappropriateContent}'가 있습니다. 부적절한 콘텐츠 없이 ShortDescription을 제공합니다.
INVALID_INPUT	LongDescription 필드에 부적절한 콘텐츠 '{InappropriateContent}'가 있습니다. 부적절한 콘텐츠 없이 LongDescription을 제공합니다.
INVALID_INPUT	SupportDescription 필드에 부적절한 콘텐츠 '{InappropriateContent}'가 있습니다. 부적절한 콘텐츠 없이 SupportDescription을 제공합니다.
INVALID_INPUT	ProductTitle 필드가 잘못되었습니다. 상표 기호 앞에 공백을 제거합니다.
INVALID_INPUT	ShortDescription 필드가 잘못되었습니다. 상표 기호 앞에 공백을 제거합니다.
INVALID_INPUT	잘못된 LongDescription 필드입니다. 상표 기호 앞에 공백을 제거합니다.
INVALID_INPUT	잘못된 SupportDescription 필드입니다. 상표 기호 앞에 공백을 제거합니다.
INVALID_INPUT	ProductTitle 필드가 잘못되었습니다. 지원되지 않는 문자[UnsupportedCharacters 제거합니다.

오류 코드	오류 메시지
INVALID_INPUT	ShortDescription 필드가 잘못되었습니다. 지원되지 않는 문자[UnsupportedCharacters 제거합니다.
INVALID_INPUT	잘못된 LongDescription 필드입니다. 지원되지 않는 문자[UnsupportedCharacters 제거합니다.
INVALID_INPUT	잘못된 SupportDescription 필드입니다. 지원되지 않는 문자[UnsupportedCharacters 제거합니다.
INVALID_INPUT	검색 키워드는 250자를 초과할 수 없습니다.
INVALID_INPUT	이 변경 유형에 대한 입력을 읽을 수 없습니다. 올바른 형식의 입력을 제출합니다.
INVALID_ADDITIONAL_RESOURCES	AdditionalResources의 잘못된 URLs: [InvalidAdditionalResourcesUrls] 유효한 URLs 제공합니다.
INVALID_CATEGORY_NAMES	에서 지원하는 유효한 범주 이름을 제공합니다 AWS Marketplace.
InvalidImageProperties	확인 오류 발견: 파일이 이미지 유형이 아닙니다. 지원되는 이미지 유형: [png jpg gif].
EXPLICIT_CONTENT	명시적 콘텐츠: '{ExplicitContent}'가 감지되었습니다. 미디어에 명시적 콘텐츠를 제공하지 않습니다.
INVALID_MEDIA	잘못된 URL: {MediaUrl} S3에 저장된 미디어의 새 URL을 제공합니다.
INVALID_MEDIA	잘못된 URL: {MediaUrl} 2048자를 초과하지 않는 유효한 URL을 제공합니다.

오류 코드	오류 메시지
INVALID_MEDIA	제공된 위치에 액세스할 수 없음: {MediaUrl} S3에 저장된 미디어에 대한 액세스 가능한 URL을 제공합니다.
INVALID_MEDIA	S3에서 미디어를 복사하는 동안 문제가 발생했습니다. 이미지 크기가 5MB를 초과합니다. 5MB 미만의 이미지를 제공합니다.
INVALID_MEDIA	미디어에서 맬웨어가 감지되었습니다. 맬웨어 없이 미디어를 다시 제출하십시오.
TOO_MANY_MEDIA	15개 이하의 미디어 항목을 제공합니다.
DUPLICATE_MEDIA	제품에 중복 미디어를 사용할 수 없습니다. 중복이 없는 미디어를 제공하십시오.

요금 차원 추가

[AWS Marketplace 카탈로그 서비스 작업을](#) 사용하여 사용자에게 AMI, 컨테이너 또는 SaaS 제품에 대한 요금을 청구할 수 있는 청구 가능한 요금 차원을 추가할 수 있습니다.

요금 차원은 판매자가 구매자에게 청구하기 위해 정의하는 측정 단위입니다. 판매자는 사용량 기반 요금 모델이든 계약 기반 요금 모델이든 상관없이 제품 사용에 대한 요금을 구매자에게 청구하도록 정보를 설정해야 합니다. 차원 유형은 제품의 요금 모델에 따라 다릅니다.

Note

무료 요금 모델이 있는 SaaS 제품의 경우 UsageBasedPricingTerm 또는 ConfigurableUpfrontPricingTerm을 사용하여 차원을 하나 이상 생성해야 하며 모든 차원의 가격은 0.00 USD여야 합니다. 이 요구 사항은 SaaS 제품에 고유하며 AMI, 컨테이너 또는 기계 학습 제품에 적용되지 않습니다.

Note

새로운 요금 차원은 SaaS 구매자에게 다음과 같은 영향을 미칩니다.

- 공개 제안에서 생성된 계약이 있는 구매자의 경우 계약이 생성될 때 제안에 존재하지 않았더라도 새 차원에 대한 소비를 보고할 수 있습니다.
- 비공개 제안에서 생성된 계약이 있는 구매자의 경우 계약이 생성될 때 비공개 제안에 존재하지 않았기 때문에 새 차원에 대한 소비를 보고할 수 없습니다. [BatchMeterUsage](#) API 호출은 성공하지만 구매자에게 요금이 청구되지 않으므로 어떤 구매자에게 새로운 차원에 대한 요금이 청구될 수 있는지, 청구될 수 없는지 추적해야 합니다. [GetAgreementTerms](#) API를 사용하여 각 구매자 계약에 포함된 차원을 볼 수도 있습니다.

새 차원에 대한 소비를 보고하고 구매자에게 청구하려면 차원이 포함된 [대체 제안을 연장](#)해야 하며 구매자는 제안을 수락해야 합니다.

제품 요금에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 다음 주제를 참조하세요.

- [AMI 제품 요금](#)
- [컨테이너 제품 요금](#)
- [SaaS 제품 요금](#)
- [기계 학습 제품 요금](#)

를 사용하여 요금 차원을 추가하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 다음 주제를 AWS Marketplace Management Portal 참조하세요.

- AMI 기반 제품: [요금 업데이트](#).
- 컨테이너 기반 제품: [요금 차원 추가](#).
- SaaS 기반 제품: [요금 차원을 추가합니다](#).
- ML 제품: 지원되지 않습니다. 기계 학습 제품에는 고정 요금 차원이 있습니다. 하지만 [요금을 업데이트할](#) 수 있습니다.

요금 차원을 추가하려면 다음 예제와 같이 AddDimensions 변경 유형으로 StartChangeSet API를 호출합니다.

Note

사용량, 계약 또는 소비 계약 등 요금 모델의 유형을 지정하는 차원이 있는 첫 번째 AddDimensions 변경 유형을 제출한 후에는 AWS Marketplace 판매자 운영 팀과 협력해야 합니다. 이를 통해 원래 요금 모델을 벗어나는 유형의 차원을 추가할 수 있습니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "AddDimensions",
      "DetailsDocument":
      [
        {
          "Description": "Description of the dimension",
          "Key": "UniqueApiKey",
          "Unit": "HostHrs",
          "Name": "First Dimension",
          "Types":
          [
            "ExternallyMetered"
          ]
        }
      ],
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "SaaSProduct@1.0"
      }
    }
  ]
}
```

AddDimensions 변경 유형에 대해 다음 필드를 제공합니다.

- **DetailsDocument** (객체 배열)(필수) - 요청의 세부 정보입니다.
- **Description** (문자열)(필수) - 구매자 보기 페이지의 긴 설명인 차원의 전체 세부 정보입니다.
- **Key** (문자열)(필수) - 제안에서 요금을 정의하는 데 사용할 패킷을 입력합니다. 또한 차원을 외부에서 측정할 수 없는 경우 AWS Marketplace Metering Service(MMS)에 게시된 차원을 입력합니다. 차원이 생성된 후에는 변경할 수 없습니다.
- **Units** (문자열)(필수) - 차원의 단위 유형입니다. 가능한 단위는 사용자, 호스트, GB, MB, TB, Gbps, Mbps, 요청, 단위, UserHrs, UnitHrs, Units, HostHrs, TierHrs 및 TaskHrs.
- **Name** (문자열)(필수) - 웹 사이트의 차원 및 고객의 청구서에 대한 표시 이름입니다.
- **Types** (문자열 배열)(필수)(태그라고도 함) - 차원이 측정, 권한 부여 또는 외부 측정 지원을 포함하는지 여부를 나타냅니다. 차원이 생성된 후에는 변경할 수 없습니다.
- **Metered** - 이 차원에 대해 측정이 발생할 수 있도록 상거래 플랫폼 사용 유형을 생성해야 함을 나타냅니다.
- **ExternallyMetered** - 판매자가 AWSSDK를 통해 AWS Marketplace 측정할 수 있도록 게시 중에 측정 서비스(MMS) 차원을 생성해야 함을 나타냅니다.
- **Entitled** - 제품 또는 제안 게시 중에 차원에 대한 권한을 부여할 수 있음을 나타냅니다.

다음 표에는 지원되는 요금 차원 및 제품 조합이 나열되어 있습니다.

요금 차원 유형	제품 유형
[Metered]	AMI
[Metered, ExternallyMetered]	SaaS, AMI/유연 소비 요금(FCP) ExternallyMetered 이 나타나면 Metered가 마스킹/유추됩니다.

요금 차원 유형	제품 유형
[Entitled]	SaaS 계약, ProServ 제품 Entitled 태그는 사용에 대한 소프트웨어/서비스 시작일 및 종료일을 사용할 수 있는 권한을 부여합니다. 또한 AMI 연간 제품에 대한 사용 할인을 받을 수 있는 권한을 부여합니다. 각 권한은 권한을 생성하거나 업데이트하기 AWS Marketplace Entitlement Service를 위해 Dimension Key의 로 식별됩니다.
[Metered, ExternallyMetered, Entitled]	차원을 선불하거나 측정할 수 있는 소비요금이 적용되는 계약은 [ExternallyMetered]와 []의 조합입니다Entitled.

- Entity (객체)(필수) - 생성 중인 명명된 엔터티 유형입니다.
- Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다. SaaSProduct@1.0. AmiProduct@1.0 또는 .

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 AddDimensions 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
설명	최대 길이: 1000 필수	400
Key(키)	최대 길이: 100 패턴: [A-Za-z0-9_-.]+\$ 필수	400
차원 단위	최대 길이: 20 필수	400
이름	최대 길이: 500 필수	400
유형(태그)	필수: 최소 1~최대 3 입력: Entitled, Metered, ExternallyMetered 필수	400

비동기 오류

다음 오류는의 AddDimensions 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#)[변경 세트 작업](#).

오류 코드	오류 메시지
누락_데이터	업데이트를 수행하기 위해 제공된 데이터가 없습니다. 1개 이상의 차원에 대한 데이터를 제공합니다.
INVALID_DIMENSION	차원을 200개 이하로 제공하십시오.
INVALID_DIMENSION	중복 차원을 추가할 수 없습니다.
INVALID_DIMENSION	현재 상태 '%s'에서는 차원을 추가할 수 없습니다. 차원 업데이트를 지원하는 상태는 %s입니다.
INVALID_DIMENSION	차원을 추가할 수 없습니다. '%s' 필드에 다른 차원의 중복 값 '%s'가 있습니다.
INVALID_DIMENSION	각 차원에 대해 비어 있지 않은 필드(키, 단위, 이름, 유형)를 제공합니다.
INVALID_TYPE	잘못된 유형 '%s'를 제거합니다. 유효한 유형은 ["Metered", "Entitled", "ExternallyMetered"]입니다.
INVALID_UNIT	잘못된 단위 '%s'를 제거합니다. 유효한 단위는 ["GB", "Gbps", "HostHrs", "Hosts", "MB", "Mbps", "Requests", "TaskHrs", "TB", "TierHrs", "UnitHrs", "Units", "UserHrs", "Users"]입니다.
INVALID_INPUT	%s 필드에 부적절한 '%s' 콘텐츠가 있습니다. %s에 부적절한 콘텐츠를 제공하지 않습니다.
INVALID_INPUT	'%s' 필드가 잘못되었습니다. 상표 기호 앞에 공백을 제거합니다.
INVALID_INPUT	'%s' 필드가 잘못되었습니다. 지원되지 않는 문자 %s를 제거합니다.

오류 코드	오류 메시지
INVALID_DIMENSION	잘못된 차원 유형 조합 %s를 제거합니다. 허용되는 값은 %s입니다.
INVALID_DIMENSION	측정된 차원에 대해 잘못된 차원 키 '%s'를 제거합니다.
INVALID_DIMENSION	productCode '%s'에 대해 '%s'라는 차원이 AWS Marketplace 측정 서비스 검증 %s를 통과하지 못했습니다.
INVALID_DIMENSION	productCode '%s'에 대해 '%s'라는 차원에는 측정 서비스에 측정 레코드가 없습니다. 제품이 테스트를 위해 시작된 적이 없거나 잘못 구성되었으며 AWS Marketplace 측정 서비스에 적절한 호출을 하지 않습니다.

요금 차원 업데이트

카탈로그 API를 사용하여 AMI, 컨테이너 또는 SaaS 제품의 기존 요금 차원을 업데이트할 수 있습니다 AWS Marketplace.

각 차원은 업데이트를 수행할 차원 키 및 차원 유형으로 고유하게 식별됩니다. 차원을 업데이트해도 원래 차원이 생성한 활성 제안이나 고객에게는 영향을 주지 않습니다.

Note

를 사용하여 요금 차원을 업데이트하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 다음 주제를 AWS Marketplace Management Portal 참조하세요.

- AMI 기반 제품: [요금 업데이트](#)
- 컨테이너 기반 제품: [차원 정보 업데이트](#)
- SaaS 기반 제품: [요금 차원 업데이트](#)
- 기계 학습 제품: [제품 요금 업데이트](#)

요금 차원을 업데이트하려면 다음 예제와 같이 UpdateDimensions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateDimensions",
      "DetailsDocument":
      [
        {
          "Key": "UniqueApiKey",
          "Types":
          [
            "ExternallyMetered"
          ],
          "Name": "First Dimension",
          "Description": "Description of the dimension"
        }
      ],
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "SaaSProduct@1.0"
      }
    }
  ]
}
```

다음 필드를 UpdateDimensions 변경 유형과 함께 사용합니다.

- DetailsDocument (객체 배열)(필수) - 요청의 세부 정보입니다.
 - Key (문자열)(필수) - 제품에서 기존 차원의 키를 제공하여 설명과 이름을 변경합니다. 이 경우 UpdateDimension이 필드는 변경할 차원을 식별하기 위한 것입니다.
 - Types (문자열 배열)(필수)(태그라고도 함) - 차원이 측정, 권한 부여 또는 외부 측정 지원을 포함하는지 여부를 나타냅니다. 차원이 생성된 후에는 변경할 수 없습니다.

- Metered -이 차원에 대해 측정이 발생할 수 있도록 상거래 플랫폼 사용 유형을 생성해야 함을 나타냅니다.
- ExternallyMetered - 판매자가 AWSSDK를 통해 측정할 수 있도록 게시 중에 AWS Marketplace Metering Service(MMS) 차원을 생성해야 함을 나타냅니다.
- Entitled - 제품/제안 게시 중에 차원에 대한 권한을 부여할 수 있음을 나타냅니다.

유효한 요금 차원 유형 조합

요금 차원 유형	제품
[Metered]	AMI
[ExternallyMetered]	SaaS, AMI/유연 소비 요금(FCP) ExternallyMetered 가 나타나면 측정됨이 마스킹/유추됩니다.
[Entitled]	SaaS 계약, ProServe 제품 Entitled 태그는 사용에 대한 소프트웨어/서비스 세트 시작 및 종료 날짜를 사용할 수 있는 권한을 부여합니다. 또한 AMI 연간 제품에 대한 사용 할인을 받을 수 있는 권한을 부여합니다. 각 권한은 권한을 생성하거나 업데이트하기 AWS Marketplace Entitlement Service위해의 차원 키로 식별됩니다.
[ExternallyMetered, Entitled]	차원을 선불하거나 측정할 수 있는 소비 요금이 적용되는 계약은 [ExternallyMetered]와 []의 조합입니다Entitled.

- Description (문자열) (선택 사항 - 구매자 보기 페이지의 긴 설명인 차원에 대한 전체 설명입니다.
- Name (문자열) 선택 사항 - 웹 사이트의 차원 및 고객의 청구서에 대한 Display 이름입니다.
- Entity (객체)(필수) - 생성 중인 명명된 엔터티 유형입니다.
- Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다SaaSProduct@1.0. AmiProduct@1.0 또는 .

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하기 위한 정보 검증이 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdateDimensions 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
설명	최대 길이: 1000 필수	400
Key(키)	최대 길이: 100 패턴: [A-Za-z0-9_-.]+\$ 필수	400
이름	최대 길이: 5 필수	400
유형(태그)	필수: 최소 1~최대 3	422

입력 필드	검증 규칙	HTTP 코드
	입력: Entitled, Metered, ExternallyMetered 필수	

비동기 오류

다음 오류는 UpdateDimensions 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_INPUT	'%s' 필드가 잘못되었습니다. 상표 기호 앞에 공백을 제거합니다.
INVALID_INPUT	'%s' 필드가 잘못되었습니다. 지원되지 않는 문자 %s를 제거합니다.
INVALID_DIMENSION	각 차원에 대해 비어 있지 않은 필드(Key, Types Name 및/또는 Description)를 제공합니다.
INVALID_DIMENSION	차원을 업데이트할 수 없습니다. 필드에 다른 차원의 중복 값 '%s'Name가 있습니다.
INVALID_DIMENSION	동일한 요청에서 키 '%s' 및 유형 '%s'로 동일한 차원을 여러 번 업데이트할 수 없습니다.
INVALID_DIMENSION	차원을 제한할 수 없습니다. 유형이 '%s'인 차원 키 '%s'가 존재하지 않습니다.
INVALID_DIMENSION	차원을 업데이트할 수 없습니다. 차원 키 '%s'는 입니다Metered.

오류 코드	오류 메시지
INVALID_DIMENSION	이미 제한된 차원에 대해서는 차원을 업데이트 할 수 없습니다.

요금 차원 제한

카탈로그 API를 사용하여 AMI 또는 SaaS 제품의 기존 요금 차원을 제한할 수 있습니다AWS Marketplace.

각 차원은 업데이트를 수행할 차원 키 및 차원 유형으로 고유하게 식별됩니다. 차원을 제한해도 원래 차원이 생성한 활성 제안이나 고객에게는 영향을 주지 않습니다.

요금 차원을 제한하려면 RestrictDimensions 변경 유형으로 StartChangeSet API를 호출합니다.

Note

요금 차원 제한은 제품이 초안 상태인 경우에만 사용할 수 있습니다. 제품이 제한 또는 공개 상태로 전환되면 변경이 허용되지 않습니다.

다음 예제에서는 SaaS 제품의 Entitled 차원을 제한하는 방법을 보여줍니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictDimensions",
      "DetailsDocument":
      [
        {
          "Key": "UniqueApiKey",
          "Types": ["Entitled"]
        }
      ]
    }
  ]
}
```



```

    }
  ],
  "Entity":
  {
    "Identifier": "prod-example12345",
    "Type": "SaaSProduct@1.0"
  }
]
}

```

다음 필드를 RestrictDimensions 변경 유형과 함께 사용합니다.

- DetailsDocument (객체 배열)(필수) - 요청의 세부 정보입니다.
- Key (문자열)(필수) - 설명과 이름을 변경할 제품의 기존 차원의 키를 제공합니다. 의 경우 RestrictDimensions이 필드는 변경할 차원을 식별하기 위한 것입니다.
- Types (문자열 배열)(필수)(태그라고도 함) - 차원이 측정, 권한 부여 또는 외부 측정 지원을 포함하는지 여부를 나타냅니다. 차원이 생성된 후에는 변경할 수 없습니다.
- ["ExternallyMetered", "Entitled"] - SaaS 계약에 대한 이러한 유형만 차원을 선불하거나 측정할 수 있는 소비 요금과 결합할 수 있습니다.
- ["Metered"] - AMI 제품의 시간당 요금 차원입니다. 이 차원에 대해 측정이 이루어질 수 있도록 상거래 플랫폼 사용 유형을 생성해야 함을 나타냅니다.
- ["ExternallyMetered"] - AMI, 컨테이너 및 SaaS 제품의 유연한 소비 요금 차원(사용자 지정 측정이라고도 함)입니다. 판매자가 AWS SDK를 통해 AWS 측정할 수 있도록 게시 중에 Marketplace 측정 서비스(MMS) 차원을 생성해야 함을 나타냅니다.
- ["Entitled"] - SaaS 계약 및 전문 서비스 제품의 계약 요금 차원. 이 태그는 소프트웨어 또는 서비스를 사용할 수 있는 권한을 부여하고, 사용 시작 및 종료 날짜를 설정하고, AMI 연간 제품에 대한 사용 할인 권한을 부여합니다. 각 권한은 권한을 생성하고 업데이트하기 AWS Marketplace Entitlement Service위해의 차원 키로 식별됩니다. 키는 제품 및 제안 게시 중에 차원에 대한 권한을 부여할 수 있음을 나타냅니다.
- Entity (객체)(필수) - 생성 중인 명명된 엔터티 유형입니다.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품의 제공 방법(제품 유형): AmiProduct@1.0 또는를 기반으로 합니다SaaSProduct@1.0.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 검증 정보가 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 [DescribeChangeSet](#) API를 호출하여 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 RestrictDimensions 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
설명	최대 길이: 1000 필수	400
Key(키)	최대 길이: 100 패턴: [A-Za-z0-9_-.]+ 필수	400
이름	최대 길이: 5 필수	400
유형(태그)	필수: 최소 1~최대 3 입력: Entitled, Metered, ExternallyMetered	400

입력 필드	검증 규칙	HTTP 코드
	필수	

비동기 오류

다음 오류는 RestrictDimensions 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리되는 DescribeChangeSet 동안 호출될 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_INPUT	'%s' 필드가 잘못되었습니다. 상표 기호 앞에 공백을 제거합니다.
INVALID_INPUT	'%s' 필드가 잘못되었습니다. 지원되지 않는 문자 %s를 제거합니다.
INVALID_DIMENSION	유형이 '%s'인 차원 키 '%s'가 이미 제한되었습니다.
INVALID_DIMENSION	차원을 제한할 수 없습니다. 유형이 '%s'인 차원 키 '%s'가 존재하지 않습니다.
INVALID_DIMENSION	중복 차원을 제한할 수 없습니다.
INVALID_DIMENSION	모든 권한 있는 차원은 제한할 수 없습니다. 하나 이상의 활성 권한 있는 차원이 있어야 합니다.
INVALID_DIMENSION	유형이 '%s'인 차원 키 '%s'는 유형이 '%s'인 다른 차원과 연결됩니다. 동일한 키의 두 차원을 동시에 제한해야 유효합니다.

대상 지정 구성 업데이트

Catalog API를 사용하여 UpdateTargeting 변경 유형을 호출하여 Public 상태로 이동하기 AWS Marketplace전에에서 AMI, 컨테이너, ML 또는 SaaS 제품을 볼 수 있는 AWS 계정IDs를 추가할 수 있습니다.

새 제품이 생성되면 관리형 카탈로그 작업(MCO) 계정이 허용된 계정 목록에 자동으로 추가됩니다. 이러한 MCO 계정은 허용된 계정을 볼 때 AWS Marketplace관리 포털(AMMP)과 DescribeEntity API 응답의 Targeting 섹션에서 판매자에게 표시됩니다.

Note

를 사용하여 AWS 계정IDs 추가하는 방법에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 다음 주제를 AWS Marketplace Management Portal참조하세요.

- AMI 기반 제품: [허용 목록 업데이트\(미리 보기 계정\)](#)
- 컨테이너 기반 제품: [AWS 계정 IDs의 허용 목록 업데이트](#)
- SaaS 기반 제품: [AWS 계정 IDs의 허용 목록 업데이트](#)
- 기계 학습 제품: [허용 목록 업데이트](#)

AMI, 컨테이너, ML 또는 SaaS 제품을 볼 수 있는 AWS 계정IDs를 추가하려면 다음 예제와 같이 UpdateTargeting 변경 유형을 사용하여 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateTargeting",
      "Entity":
      {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-example12345"
      }
    },
  ],
}
```

```

    "DetailsDocument":
    {
        "PositiveTargeting":
        {
            "BuyerAccounts":
            [
                "1112223334444"
            ]
        }
    }
}

```

다음 필드를 UpdateTargeting 변경 유형과 함께 사용합니다.

- Entity (객체)(필수) - 생성 중인 명명된 엔터티 유형입니다.
- Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다. 다SaaSProduct@1.0, AmiProduct@1.0, MachineLearningProduct@1.0 또는 .
- DetailsDocument (객체)(필수) - ChangeSet를 실행하는 데 필요한 세부 정보입니다.
 - PositiveTargeting (객체) (선택 사항) - 긍정적인 타겟팅은 구매자의 프로필이 제안에 액세스할 수 있도록 허용하기 위해 충족해야 하는 기준을 정의합니다. 이 필드는 선택 사항이지만 이 필드가 있는 경우 하나 이상의 대상 지정 옵션을 제공해야 합니다.
 - BuyerAccounts (문자열 배열)(선택 사항) - AWS 계정을 기반으로 대상 지정을 허용하는 옵션으로 나열합니다(프라이빗 제안이라고도 함). AWS 계정에 대한 제안을 대상으로 지정하지 않으려는 경우 이 필드를 생략해야 합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
    "ChangeSetId": "example123456789012abcdef",
    "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 검증 정보가 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

요청 상태를 확인하려면 [DescribeChangeSet](#) API를 AWS Marketplace Management Portal에 호출합니다.

요청이 완료되면(가 Status인 경우 SUCCEEDED) 새 ProductId가 생성됩니다.

동기 검증

다음 스키마 검증은 UpdateTargeting 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

대상 지정 유형	유효한 현재 가시성 상태	BuyerAccounts(입력)	Check]를 선택합니다
긍정적:	퍼블릭, Limited 또는 Draft	12자리 AWS 계정ID 문자열 배열입니다. 최소 크기: 0. 최대 크기: 5000.	입력은 현재 문서 대상 계정과 달라야 합니다. 입력이 유효한 상태여야 합니다. AWS 계정.

비동기 오류

다음 오류는 DescribeChangeSet 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요 변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_PRODUCT_VISIBILITY	기존 퍼블릭 Limited 또는 Draft 제품을 사용합니다.
INVALID_AWS_ACCOUNT_IDS	valid AWS 계정IDs. AWS 계정 not found를 제공합니다. [x, y, z].

오류 코드	오류 메시지
ValidationException	전문 서비스 제품에는 허용 목록이 없습니다. 다른 제품 유형과 달리 제한된 상태의 전문 서비스 제품은 허용 목록 없이 모든 구매자에게 확장할 수 있습니다.

제품 표시 여부 업데이트

카탈로그 API를 사용하여 AMI, 컨테이너, ML 또는 SaaS 제품의 가시성(수명 주기 상태라고도 함)을 업데이트할 수 있습니다AWS Marketplace.

Note

를 사용하여 제품 가시성을 업데이트하는 자세한 내용은 AWS Marketplace판매자 안내서의 다음 주제를 AWS Marketplace Management Portal참조하세요.

- AMI 기반 제품: [제품 가시성 업데이트](#)
- 컨테이너 기반 제품: [제품 가시성 업데이트](#)
- SaaS 기반 제품: [제품 가시성 업데이트](#)
- 기계 학습 제품: [제품 가시성 업데이트](#)

허용되는 대상 수명 주기 상태는 Limited, Public또는 입니다Restricted.

Limited

제품이 완료되고 ReleaseProduct가 성공적으로 완료되었습니다ChangeType. 판매자는이 상태의 제품 세부 정보를 볼 수 있습니다. 제품은 공개되지 않습니다. 그러나 판매자는가 제품을 미리 볼 수 있도록 특정 구매자를 대상으로 지정할 수 있습니다.

Public

제품이에 표시됩니다AWS Marketplace. 구매자는 제품을 보고 구독할 수 있습니다.

Restricted

제품은 더 이상 일반에 공개되지 않으며 새 구독자를 수락하지 않습니다. 기존 구독자는 구독이 만료될 때까지이 제품을 계속 사용할 수 있습니다.

Note

UpdateVisibility 변경 유형을 사용하려면 AWS Marketplace 판매자 운영 팀의 수동 검토가 필요하므로 실행 시간이 길어집니다. 자체 변경 세트에서 UpdateVisibility 별도로 사용합니다.

제품의 가시성을 업데이트하려면 다음 예제와 같이 UpdateVisibility 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

TargetVisibility가 Public 또는 인 경우 Limited.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateVisibility",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-example12345"
      },
      "DetailsDocument": {
        "TargetVisibility": "Public"
      }
    }
  ]
}
```

TargetVisibility가 인 경우 입니다Restricted.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
```



```

"Catalog": "AWSMarketplace",
"ChangeSet":
[
  {
    "ChangeType": "UpdateVisibility",
    "Entity":
    {
      "Type": "SaaSProduct@1.0",
      "Identifier": "prod-example12345"
    },
    "DetailsDocument":
    {
      "TargetVisibility": "Restricted",
      "ReplacementProductId": "prod-example54321"
    }
  }
]
}

```

UpdateVisibility 변경 유형을 추가할 필드에 대한 정보를 제공합니다.

- Entity (객체)(필수) - 생성 중인 명명된 엔터티 유형입니다.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다 MachineLearningProduct@1.0, AmiProduct@1.0, SaaSProduct@1.0, ContainerProduct@1.0, . 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument (객체)(필수) -를 실행하는 데 필요한 세부 정보입니다ChangeSet.
 - TargetVisibility - 제품의 의도한 새로운 가시성입니다.

가능한 값: Public, Limited 및 Restricted

- ReplacementProductId (문자열) (선택 사항) - 제품의 대체 제품 ID가 입니다Restricted. 현재 구독자에게 제품 제한에 대해 알리는 데 사용됩니다.

에 Restricted 대해서만 허용합니다TargetVisibility.

동기 검증

다음 스키마 검증은의 UpdateVisibility 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

대상 지정 유형	유효한 현재 상태	ReplacementProduct Id(입력)	검증 검사
퍼블릭	Limited 및 Restricted	허용되지 않음	유효한 현재 상태
Limited	퍼블릭 및 Restricted	허용되지 않음	유효한 현재 상태
Restricted	퍼블릭 및 Limited	문자열(선택 사항)	ReplacementProduct Id는 기존 Limited 또는 퍼블릭 제품에 속해야 합니다.

이 변경 유형을 트리거한 후 완료하는 데 최대 37일이 걸릴 수 있습니다. 여기에는 AWS Marketplace 판매자 운영 팀이 검토, 감사 및 승인해야 하는 시간이 포함됩니다. 제품을 제한할 때 AWS Marketplace 판매자 운영 팀이 감사를 시작하기 CancelChangeSet전예를 호출하여 24시간 동안 마음이 바뀌어야 합니다. 자세한 내용은 [CancelChangeSet](#) 단원을 참조하십시오.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 검증 정보가 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

요청이 완료되면(가 Status인 경우SUCCEEDED) 새 ProductId가 생성됩니다.

비동기 오류

다음 오류는 UpdateVisibility 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_PRODUCT_STATE	기존 Public, Limited 또는 Restricted 제품을 사용합니다.
INVALID_TARGET_VISIBILITY	유효한 대상 가시성 상태 Public, Limited 또는 Restricted를 제공합니다.
EMPTY_TARGET_VISIBILITY	유효한 대상 가시성 상태 Public, 또는 Limited를 제공합니다. Restricted .
INVALID_REPLACEMENT_PRODUCT_ID	기존 Public 또는 Limited 제품을 대체품으로 사용합니다.
INVALID_REPLACEMENT_PRODUCT_ID	대체 제품 ID는 제품을 제한할 때만 유효합니다.
감사_오류	MCO 수동 검토에 따라 다릅니다.
MISSING_SELLER_PROFILE_INFORMATION	제품을 퍼블릭으로 업데이트하려면 먼저 판매자 계정에 퍼블릭 프로필을 추가해야 합니다.

제품 게시

카탈로그 API를 사용하여 Draft AMI, 컨테이너, ML 또는 SaaS 제품을 Limited 상태로 게시할 수 있습니다. AWS Marketplace.

Note

AmiProduct@1.0 및 SaaSProduct@1.0의 경우 ReleaseProduct 변경 유형에는 이 제품에 대해 생성된 해당 초안 퍼블릭 Offer@1.0 엔터티의 ReleaseOffer 변경 유형이 포함되어야 합니다.

제품을 게시하려면 다음 예제와 같이 ReleaseProduct 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-example12345"
      },
      "DetailsDocument": {}
    }
  ]
}
```

ReleaseProduct 변경 유형에 추가할 필드에 대한 정보를 제공합니다. 이 변경 유형은 파라미터 페이로드를 사용하지 않습니다.

- Entity (객체)(필수) - 생성 중인 명명된 엔터티 유형입니다.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다 MachineLearningProduct@1.0, AmiProduct@1.0, SaaSProduct@1.0, ContainerProduct@1.0, . 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument (객체) (필수) - 빈 객체여야 합니다. 변경 유형은 세부 정보를 허용하지 않습니다 ReleaseProduct.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하기 위한 정보 검증이 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

요청이 완료되면(가 Status인 경우 SUCCEEDED) 새 ProductId가 생성됩니다.

비동기 오류

다음 오류는 ReleaseProduct 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
VALIDATION_FAILED	설명 정보를 제공합니다.
VALIDATION_FAILED	버전 정보를 제공합니다.
VALIDATION_FAILED	차원 정보를 제공합니다.
VALIDATION_FAILED	설명 PromotionalResources SupportInformation 정보를 제공합니다.

제품 ID 찾기

카탈로그 API를 사용하여 제품을 수정하려면 먼저 제품의 제품 ID를 AWS Marketplace 가져와야 합니다. 서버 제품의 제품 ID를 찾는 방법에는 두 가지가 있습니다.

- 를 열고 판매자 계정으로 AWS Marketplace Management Portal 로그인합니다. 제품 메뉴에서 서버 제품을 선택한 다음 관심 있는 제품을 선택합니다. 제품 ID는 제품 요약 섹션에 나열되어 있습니다.

- EntityType **AmiProduct** 또는 **ContainerProduct**, **MachineLearningProduct**, 또는 **SaaSProduct**와 함께 [ListEntities](#) 작업을 사용하여 카탈로그 API를 통해 제품 IDs를 포함한 제품 목록을 **DataProduct** 가져옵니다.는 엔터티 유형의 버전(예: AmiProduct@1.0)을 포함하지 않도록 ListEntities 요구합니다.

Note

제품 ID는 제품이 게시된 후에만 사용할 수 있으며 최소한 사용자가 볼 수 있습니다AWS Marketplace. 제품을 처음 생성할 때 제품을 검토하고 완전히 생성하는 데 며칠이 걸릴 수 있습니다. 이 기간 동안에는 제품 ID를 사용할 수 없습니다.

다음 주제에서는 엔터티 ID, 제품 제목, 마지막 수정 날짜 또는 가시성을 기준으로 필터링하여 제품을 찾는 방법을 설명합니다.

주제

- [제품 제목을 기반으로 제품 찾기](#)
- [마지막 수정 날짜를 기준으로 제품 찾기](#)
- [제품 가시성을 기반으로 제품 찾기](#)
- [제품 제목, 마지막 수정 날짜 및 제품 가시성을 기반으로 제품 찾기](#)
- [제품에 대한 추가 세부 정보 가져오기](#)

제품 제목을 기반으로 제품 찾기

요청

```
POST /ListEntities HTTP/1.1
Content-Type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "MaxResults": 10,
  "EntityTypeFilters": {
    "AmiProductFilters": {
      "ProductTitle": {
        "WildcardValue": "XYZ"
```

```

    }
  }
}

```

응답

```

HTTP/1.1 200
Content-type: application/json

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example-abcd-1234",
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "AmiProductSummary": {
        "ProductTitle": "ABC-XYZ-123",
        "Visibility": "Public"
      }
    }
  ],
  "NextToken": ""
}

```

마지막 수정 날짜를 기준으로 제품 찾기

요청

```

POST /ListEntities HTTP/1.1
Content-Type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "MaxResults": 10,
  "EntityTypeFilters": {
    "AmiProductFilters": {
      "LastModifiedDate": {
        "DateRange": {

```

```

        "BeforeValue": "2018-03-27T13:45:22Z",
        "AfterValue": "2018-01-27T13:45:22Z"
      }
    }
  }
}

```

응답

```

HTTP/1.1 200
Content-type: application/json

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
      AmiProduct/example-abcd-1234",
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "AmiProductSummary": {
        "ProductTitle": "ABC-XYZ-123",
        "Visibility": "Public"
      }
    }
  ],
  "NextToken": ""
}

```

제품 가시성을 기반으로 제품 찾기

요청

```

POST /ListEntities HTTP/1.1
Content-Type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "MaxResults": 10,
  "EntityTypeFilters": {

```



```

        "AmiProductFilters": {
            "Visibility": {
                "ValueList": [
                    "Public"
                ]
            }
        }
    }
}

```

응답

```

HTTP/1.1 200
Content-type: application/json

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example-abcd-1234",
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "AmiProductSummary": {
        "ProductTitle": "ABC-XYZ-123",
        "Visibility": "Public"
      }
    }
  ],
  "NextToken": ""
}

```

제품 제목, 마지막 수정 날짜 및 제품 가시성을 기반으로 제품 찾기

요청

```

POST /ListEntities HTTP/1.1
Content-Type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",

```

```

    "MaxResults": 10,
    "EntityTypeFilters": {
      "AmiProductFilters": {
        "LastModifiedDate": {
          "DateRange": {
            "BeforeValue": "2018-03-27T13:45:22Z",
            "AfterValue": "2018-01-27T13:45:22Z"
          }
        },
        "Visibility": {
          "ValueList": [
            "Public"
          ]
        },
        "ProductTitle": {
          "ValueList": [
            "ABC-XYZ-123"
          ]
        }
      }
    }
  }
}

```

응답

```

HTTP/1.1 200
Content-type: application/json

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example-abcd-1234",
      "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "AmiProductSummary": {
        "ProductTitle": "ABC-XYZ-123",
        "Visibility": "Public"
      }
    }
  ],

```

```
"NextToken": ""
}
```

제품에 대한 추가 세부 정보 가져오기

DescribeEntity 작업과 함께 개체 ID를 사용하여 제품에 대한 추가 세부 정보를 얻을 수 있습니다.

요청

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=example-abcd-1234 HTTP/1.1
```

응답

```
HTTP/1.1 200
Content-type: application/json

{
  "DetailsDocument": {
    "ProductTitle": "ABC-XYZ-123",
    "ShortDescription": "My product short description.",
    "LongDescription": "My product longer description.",
    "Sku": "123example456",
    "SupportDescription": "Need help? Contact our experts at support@example.com \n\nYour purchase includes 24x7 support.",
    "Categories": [
      "Operating Systems",
      "Network Infrastructure",
      "Application Development"
    ]
  }
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
  AmiProduct/example-abcd-1234",
  "EntityId": "example1-abcd-1234-5ef6-7890abcdef12@1",
  "EntityType": "AmiProduct",
  "LastModifiedDate": "2018-02-27T13:45:22Z",
}
```

설정 상태 및 오류 변경

AWS Marketplace 카탈로그 API에서 판매자 제품을 변경하려면 변경하려는 변경 사항을 설명하는 변경 세트를 생성한 다음 StartChangeSet 작업을 사용하여 변경 사항을 시작합니다. 요청에 따라 요

청 변경 사항을 완료하는 데 몇 분에서 몇 시간 또는 그 이상 걸릴 수 있습니다. 이 요청에 대한 응답은 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청은 대기열에 추가되고 처리되며, 여기에는 제품 AWS Marketplace 지침을 충족하는지 확인하기 위한 파일 및 정보 스캔이 포함됩니다. 변경 요청에 따라이 프로세스는 몇 분에서 며칠 정도 걸릴 수 있습니다. 를 통해 AWS Marketplace Management Portal 또는 DescribeChangeSet 작업을 사용하여 카탈로그 API에서 요청 상태를 확인할 수 있습니다. 변경 세트에 대한 자세한 내용은 [변경 세트 작업](#)의 내용을 참조하세요.

요청 상태를 확인하려면 DescribeChangeSet 작업을 사용합니다.

```
POST /DescribeChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSetID": "example123456789012abcdef"
}
```

이 호출의 결과는 다음과 같습니다(이 경우 컨테이너 제품에 새 버전을 추가하는 경우).

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2020-10-27T22:21:26Z",
  "EndTime": "2020-10-27T22:32:19Z",
  "Status": "SUCCEEDED",
  "ChangeSet":
  [
    {
      "ChangeType": "AddDeliveryOptions",
      "Entity":
      {
        "Type": "ContainerProduct@1.0",
```

```

    "Identifier": "example-1234-abcd-56ef-abcdef12345678@4"
  },
  "Details": "{\\\"Version\\\": {\\\"VersionTitle\\\": \\\"1.1\\\", \\\"ReleaseNotes\\\": \\\"Minor bug fix\\\"}, \\\"DeliveryOptions\\\": [{\\\"DeliveryOptionTitle\\\": \\\"EKSDelivery\\\", \\\"Details\\\": {\\\"EcrDeliveryOptionDetails\\\" : {\\\"ContainerImages\\\": [\\\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\\\"], \\\"DeploymentResources\\\": [{\\\"Name\\\": \\\"HelmDeploymentTemplate\\\", \\\"Url\\\": \\\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame2:mychart1.1\\\"}], \\\"CompatibleServices\\\": [\\\"EKS\\\"], \\\"Description\\\": \\\"Sample Description\\\", \\\"UsageInstructions\\\": \\\"helm pull 111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame2:mychart1.1\\\"}}}, {\\\"DeliveryOptionTitle\\\": \\\"HelmChartDeliveryOption\\\", \\\"Details\\\": {\\\"HelmDeliveryOptionDetails\\\": {\\\"CompatibleServices\\\": [\\\"EKS\\\", \\\"EKS-Anywhere\\\"], \\\"ContainerImages\\\": [\\\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\\\"], \\\"HelmChartUri\\\": \\\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:helmchart1.1\\\", \\\"Description\\\": \\\"Helm chart description\\\", \\\"UsageInstructions\\\": \\\"Usage instructions\\\", \\\"MarketplaceServiceAccountName\\\": \\\"Service account name\\\", \\\"ReleaseName\\\": \\\"Optional release name\\\", \\\"Namespace\\\": \\\"Optional Kubernetes namespace\\\", \\\"OverrideParameters\\\": [{\\\"Key\\\": \\\"HelmKeyName1\\\", \\\"DefaultValue\\\": \\\"${AWSMP_LICENSE_SECRET}\\\"}, {\\\"Key\\\": \\\"HelmKeyName2\\\", \\\"DefaultValue\\\": \\\"${AWSMP_SERVICE_ACCOUNT}\\\"}]]}}}}\",
  \"DetailsDocument\":
  {
    \"Version\":
    {
      \"VersionTitle\": \"1.1\",
      \"ReleaseNotes\": \"Minor bug fix\"
    },
    \"DeliveryOptions\":
    [
      {
        \"DeliveryOptionTitle\": \"EKSDelivery\",
        \"Details\":
        {
          \"EcrDeliveryOptionDetails\":
          {
            \"ContainerImages\":
            [
              \"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\"
            ],
            \"DeploymentResources\":
            [
              {
                \"Name\": \"HelmDeploymentTemplate\",

```

```

        "Url": "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame2:mychart1.1"
    }
  ],
  "CompatibleServices":
  [
    "EKS"
  ],
  "Description": "Sample Description",
  "UsageInstructions": "helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1"
}
}
},
{
  "DeliveryOptionTitle": "HelmChartDeliveryOption",
  "Details":
  {
    "HelmDeliveryOptionDetails":
    {
      "CompatibleServices":
      [
        "EKS",
        "EKS-Anywhere"
      ],
      "ContainerImages":
      [
        "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
      ],
      "HelmChartUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/reponame1:helmchart1.1",
      "Description": "Helm chart description",
      "UsageInstructions": "Usage instructions",
      "MarketplaceServiceAccountName": "Service account name",
      "ReleaseName": "Optional release name",
      "Namespace": "Optional Kubernetes namespace",
      "OverrideParameters":
      [
        {
          "Key": "HelmKeyName1",
          "DefaultValue": "${AWSMP_LICENSE_SECRET}"
        },
        {

```

```

        "Key": "HelmKeyName2",
        "DefaultValue": "${AWSMP_SERVICE_ACCOUNT}"
    }
  ]
}
}
},
"ErrorDetailList":
[]
}
]
}

```

Status 필드에는 요청의 현재 상태가 표시됩니다. 이 경우입니다 SUCCEEDED.

실패하는 경우 결과에 두 가지 유형의 오류가 포함될 수 있습니다. 대부분의 오류의 경우 오류 메시지가 직접 포함됩니다. 그러나 제품에서 보안 취약성을 스캔하는 동안 발견된 오류에는 ErrorMessage 필드에 발견된 모든 오류를 나열하는 파일의 URL이 포함됩니다. 스캔 중에 발견된 오류는 ErrorCode "SCAN_ERROR"입니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2020-10-27T22:21:26Z",
  "EndTime": "2020-10-27T22:32:19Z",
  "Status": "FAILED",
  "FailureDescription": "Change set preparation has failed. For details see 'ErrorDetailList'.",
  "ChangeSet":
  [
    {
      "ChangeType": "AddDeliveryOptions",
      "Entity":
      {
        "Type": "ContainerProduct@1.0",
        "Identifier": "example-1234-abcd-56ef-abcdef12345678@4"
      }
    },
  ]
}

```

```

    "Details": "{\\"Version\\": {\\"VersionTitle\\": \\"1.1\\",\\"ReleaseNotes\\": \\"Minor
    bug fix\\"},\\"DeliveryOptions\\": [{\\"DeliveryOptionTitle\\": \\"EKSDelivery\\",\\"Details
    \": {\\"EcrDeliveryOptionDetails\\": {\\"ContainerImages\\": [\\"111122223333.dkr.ecr.us-
    east-1.amazonaws.com/sellername/reponame1:1.1\\"],\\"DeploymentResources\\": [{\\"Name\\":
    \\"HelmDeploymentTemplate\\",\\"Url\\": \\"111122223333.dkr.ecr.us-east-1.amazonaws.com/
    sellername/reponame2:mychart1.1\\"}],\\"CompatibleServices\\": [\\"EKS\\"],\\"Description
    \": \\"Sample Description\\",\\"UsageInstructions\\":\\"helm pull 111122223333.dkr.ecr.us-
    east-1.amazonaws.com/sellername/reponame2:mychart1.1\\"}}},{\\"DeliveryOptionTitle
    \": \\"HelmChartDeliveryOption\\",\\"Details\\": {\\"HelmDeliveryOptionDetails\\":
    {\\"CompatibleServices\\": [\\"EKS\\", \\"EKS-Anywhere\\"],\\"ContainerImages\\":
    [\\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\\"],
    \\"HelmChartUri\\": \\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
    reponame1:helmchart1.1\\",\\"Description\\": \\"Helm chart description\\",
    \\"UsageInstructions\\": \\"Usage instructions\\",\\"MarketplaceServiceAccountName\\":
    \\"Service account name\\",\\"ReleaseName\\": \\"Optional release name\\",\\"Namespace\\":
    \\"Optional Kubernetes namespace\\",\\"OverrideParameters\\": [{\\"Key\\": \\"HelmKeyName1\\",
    \\"DefaultValue\\": \\"${AWSMP_LICENSE_SECRET}\\"},{\\"Key\\": \\"HelmKeyName2\\",
    \\"DefaultValue\\": \\"${AWSMP_SERVICE_ACCOUNT}\\"}]}]}]}"}",
    "DetailsDocument":
    {
      "Version":
      {
        "VersionTitle": "1.1",
        "ReleaseNotes": "Minor bug fix"
      },
      "DeliveryOptions":
      [
        {
          "DeliveryOptionTitle": "EKSDelivery",
          "Details":
          {
            "EcrDeliveryOptionDetails":
            {
              "ContainerImages":
              [
                "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
              ],
              "DeploymentResources":
              [
                {
                  "Name": "HelmDeploymentTemplate",
                  "Url": "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame2:mychart1.1"
                }
              ]
            }
          }
        }
      ]
    }
  }

```



```

        }
    ],
    "CompatibleServices":
    [
        "EKS"
    ],
    "Description": "Sample Description",
    "UsageInstructions": "helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1"
    }
}
},
{
    "DeliveryOptionTitle": "HelmChartDeliveryOption",
    "Details":
    {
        "HelmDeliveryOptionDetails":
        {
            "CompatibleServices":
            [
                "EKS",
                "EKS-Anywhere"
            ],
            "ContainerImages":
            [
                "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
            ],
            "HelmChartUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/reponame1:helmchart1.1",
            "Description": "Helm chart description",
            "UsageInstructions": "Usage instructions",
            "MarketplaceServiceAccountName": "Service account name",
            "ReleaseName": "Optional release name",
            "Namespace": "Optional Kubernetes namespace",
            "OverrideParameters":
            [
                {
                    "Key": "HelmKeyName1",
                    "DefaultValue": "${AWSMP_LICENSE_SECRET}"
                },
                {
                    "Key": "HelmKeyName2",
                    "DefaultValue": "${AWSMP_SERVICE_ACCOUNT}"
                }
            ]
        }
    }
}

```

```

        }
      ]
    }
  }
]
},
"ErrorDetailList":
[
  {
    "ErrorCode": "DUPLICATE_VERSION_TITLE",
    "ErrorMessage": "The version title must be different from any other version
titles of this product."
  },
  {
    "ErrorCode": "SCAN_ERROR",
    "ErrorMessage": "https://123sample456.cloudfront.net/example-1234-abcd-5678-
abcdef12345678/1234abcdef567890"
  }
]
}
]
}

```

이 예제에서는 직접 보고된 오류(DUPLICATE_VERSION_TITLE)가 하나 있습니다. 다른 오류에는 오류 메시지가 있는 파일이 있습니다(단일 예는 연결된 파일에서 발견된 오류가 여러 개 있을 SCAN_ERROR 수 있음).

Note

에 반환된 링크ErrorMessage는 60일 동안 유효합니다.

AMI 기반 제품 작업

를 사용하여 Amazon Machine Image(AMI) 기반 제품 작업을 위한 작업을 자동화AWS Marketplace Catalog API할 수 있습니다.

카탈로그 API를 사용하여 AMI 기반 제품을 생성하는 방법에 대한 자세한 내용은 섹션을 참조하세요[제품 생성](#).

다음 주제에서는 카탈로그 API를 사용하여 AMI 기반 제품에 대한 작업을 수행하는 방법을 설명합니다.

주제

- [새 버전 추가](#)
- [버전 정보 업데이트](#)
- [버전 제한](#)
- [향후 AWS 리전지원 업데이트](#)
- [지원되는 추가AWS 리전](#)
- [제한AWS 리전](#)
- [새 인스턴스 유형 추가](#)
- [인스턴스 유형 제한](#)

새 버전 추가

카탈로그 API를 사용하여 기존 AMI 기반 제품에 새 버전을 추가할 수 있습니다AWS Marketplace. 를 사용하여 제품에 새 AMI 버전을 추가하는 방법에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 [새 버전 추가](#)를 AWS Marketplace Management Portal참조하세요.

AWS Marketplace에 새 AMI를 추가하는 요청이 포함된 요청을 생성할 때는 AMI를 AWS Marketplace 시스템에 복사한 다음 보안 문제를 스캔해야 합니다. AMI에서 작업을 수행할 권한이 있는 AWS Identity and Access Management(IAM) 역할을 생성하여 AWS Marketplace에 AMI에 대한 액세스 권한을 부여해야 합니다. 필요한 권한에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 [AWS Marketplace에 AMI에 대한 액세스 권한 부여를 참조하세요](#).

새 버전을 추가하려면 다음 예제와 같이 AMI 기반 제품의 AddDeliveryOptions 변경 유형으로 StartChangeSet API 작업을 호출합니다. 실제로 새 버전을 생성하지 않고 API 호출을 테스트하려면 Intent 파라미터를 로 설정합니다VALIDATE. 자세한 내용은 [의도](#)를 참조하세요.

Note

AMI 제품의 경우 버전은 하나 이상의 전송 옵션으로 구성됩니다. 동일한 버전의 모든 전송 옵션은 동일한 세부 정보를 가진 동일한 AmiSource 객체를 가져야 합니다. 버전을 생성할 때 모든 전송 옵션을 버전에 추가해야 합니다. 기존 버전에는 전송 옵션을 추가할 수 없습니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "AddDeliveryOptions",
      "Entity":
      {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-example12345"
      },
      "DetailsDocument":
      {
        "Version":
        {
          "VersionTitle": "My new title",
          "ReleaseNotes": "My new Release notes"
        },
        "DeliveryOptions":
        [
          {
            "Details":
            {
              "AmiDeliveryOptionDetails":
              {
                "AmiSource":
                {
                  "AmiId": "ami-1234567890abcdef",
                  "AccessRoleArn": "arn:aws:iam::12345678901:role/
AWSMarketplaceAmiIngestion",
                  "UserName": "ec2-user",
                  "OperatingSystemName": "AMAZONLINUX",
                  "OperatingSystemVersion": "Amazon Linux 2 AMI 2.0.20210126.0 x86_64
HVM gp2",
                  "AfiIds": ["afi-1234567890abcdefg", "afi-abcdefg1234567890"] //
Optional (Required only for FPGA products)
                },
                "UsageInstructions": "Easy to use AMI",
                "RecommendedInstanceType": "m4.xlarge",
                "SecurityGroups":
                [
                  {
                    "IpProtocol": "tcp",

```

```

        "FromPort": 443,
        "ToPort": 443,
        "IpRanges":
        [
            "0.0.0.0/0"
        ]
    }
}
},
{
    "DeliveryOptionTitle": "My new AMI with CFTdelivery option",
    "Details":
    {
        "DeploymentTemplateDeliveryOptionDetails":
        {
            "ShortDescription": "My new short description",
            "LongDescription": "My new long description",
            "UsageInstructions": "My new usage instructions",
            "RecommendedInstanceType": "m4.xlarge",
            "ArchitectureDiagram": "https://my-bucket.s3.amazonaws.com/my-folder/
diagram.png",
            "Template": "https://my-bucket.s3.amazonaws.com/my-folder/
cft.template",
            "TemplateSources":
            [
                {
                    "ParameterName": "MyAmiParam",
                    "AmiSource":
                    {
                        "AmiId": "ami-1234567890abcdef",
                        "AccessRoleArn": "arn:aws:iam::12345678901:role/
AwsMarketplaceAmiIngestion",
                        "UserName": "ec2-user",
                        "OperatingSystemName": "AMAZONLINUX",
                        "OperatingSystemVersion": "Amazon Linux 2 AMI 2.0.20210126.0
x86_64 HVM gp2"
                    }
                }
            ]
        }
    }
}
}

```

```

    ]
  }
}
],
"Intent": "APPLY"
}

```

다음 목록은 AddDeliveryOptions 변경 유형과 함께 사용하는 입력 필드에 대한 정보를 제공합니다. 달리 명시되지 않는 한 모든 필드는 필수입니다. 이러한 필드에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [새 버전 추가](#)를 참조하세요.

- DeliveryOptions(객체 배열) - 각 객체의 세부 정보를 포함한 전송 옵션 객체 목록입니다. 하나 이상을 포함해야 합니다.
- AMI(독립 실행형) 전송 옵션에서 다음 세부 정보가 포함된 객체를 지정합니다. 이 유형의 전송 옵션은 하나만 가질 수 있습니다.
 - 세부 정보(객체) - 이 전송 옵션의 세부 정보를 보유합니다. 이 중첩된 세부 정보 객체는 이중 이 스케이프할 필요가 없습니다.
 - AmiDeliveryOptionDetails(객체) -를 사용하여 각 AMI 전송 옵션의 세부 정보를 제공합니다.
 - AmiSource(객체) - 추가된 버전에 사용할 AMI에 대한 세부 정보입니다.
 - AmiId(문자열) - API가 호출되는 AWS 리전에 있는 소스 AMI의 ID입니다. 카탈로그 API 를 사용할 수 있는 유일한 리전이므로 항상 미국 동부(버지니아 북부)여야 합니다. 호출자 계정에 속해야 합니다.
 - AccessRoleArn(문자열) -가 제공된 AMI에 액세스하는 AWS Marketplace에 사용하는 IAM 역할 Amazon 리소스 이름(ARN)입니다. 이 ARN 생성 및 사용에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [AMI에 대한 AWS Marketplace 액세스 권한 부여를 참조하세요](#).
 - UserName(문자열) - 사용자 이름으로 로그인하여 AMI의 운영 체제(OS)에 액세스합니다. 일반적으로 Linux AMIs용 ec2-user 또는 Windows용 관리자.
 - ScanningPort(정수) - OS에 액세스하는 데 사용되는 SSH 또는 RDP 포트입니다. 제공된 AMI에서 보안 취약성을 스캔하는 데 사용됩니다. 기본값은 22입니다.
 - OperatingSystemName(문자열) - 구매자에게 표시되는 운영 체제의 이름입니다.
 - OperatingSystemVersion(문자열) - 구매자에게 표시되는 운영 체제 버전 문자열입니다.
 - AfiIds(문자열 배열)(선택 사항) -이 버전에 대한 FPGA 지원을 활성화하는 Amazon FPGA 이미지(AFI) IDs. 제공되면 FPGA 지원 F2 인스턴스 유형에서 제품을 실행할 수 있습니다. 버전당 최대 15개의 AFI IDs를 제공할 수 있습니다. 제공하는 모든 AFI ID는 미국 동부(버지니아 북부) 리전에서 시작하여 AWS Marketplace 판매자 계정 내에 있어야 하며,

제공된 IAM 액세스 역할에는 이 AFI를 AWS Marketplace와 공유할 수 있는 권한이 있어야 합니다. 필요한 권한에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [FPGA 이미지에 대한 액세스 권한 AWS Marketplace 부어를 참조하세요](#).

- **UsageInstructions(문자열)** - AMI 사용에 대한 지침 또는 AMI에 대한 자세한 정보 링크입니다.
- **AccessEndpointUrl(객체)(선택 사항)** - AMI를 사용한 후 AMI에 액세스할 경로를 생성하는 데 사용됩니다.
 - **포트(문자열)** - AMI에서 실행되는 서비스에 액세스하는 데 사용되는 포트 번호입니다.
 - **프로토콜(문자열)** - AMI에서 실행되는 서비스에 액세스하는 데 사용되는 프로토콜(http 또는 https)입니다.
- **RelativePath(문자열)** - AMI에서 실행되는 서비스에 액세스하기 위한 웹 루트의 경로입니다(예: /index.html).
- **RecommendedInstanceType(문자열)** - AMI로 서비스를 실행하는 데 권장되는 인스턴스 유형으로, 원클릭 서비스 설치의 기본값입니다. 인스턴스 유형 목록은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 [유형을](#) 참조하세요.
- **SecurityGroups(객체 배열)** - 버전에 대해 자동으로 생성된 그룹의 수신 규칙을 나타내는 객체 목록입니다.
 - **IpProtocol(문자열)** - 사용할 프로토콜(tcp 또는 udp).
 - **FromPort(정수)** - 소스 포트입니다.
 - **ToPort(정수)** - 대상 포트입니다.
 - **IpRanges(문자열 배열)** - CIDR 형식(xxx.xxx.xxx.xxx/nn 형식, 예: **192.0.2.0/24**)으로 허용할 IP 범위입니다.
- **CloudFormation 전송 옵션이 있는 AMI의 경우** 다음 세부 정보가 포함된 객체를 지정합니다. 이 유형의 전송 옵션은 최대 3개까지 가능합니다.
 - **DeliveryOptionTitle(문자열)** - 전송 옵션의 제목입니다.
 - **세부 정보(객체)** - AMI 전송 옵션의 세부 정보를 보유합니다. 이 중첩된 세부 정보 객체는 이중 이스케이프할 필요가 없습니다.
 - **DeploymentTemplateDeliveryOptionDetails(객체)** -를 사용하여 각 CFT 전송 옵션의 세부 정보를 제공합니다.
 - **ShortDescription(문자열)** - CloudFormation 템플릿 전송 옵션에 대한 간략한 설명입니다.
 - **LongDescription(문자열)** - CloudFormation 템플릿 전송 옵션에 대한 자세한 설명입니다.
 - **UsageInstructions(문자열)** - AMI 사용에 대한 지침 또는 AMI에 대한 자세한 정보 링크입니다.

- **RecommendedInstanceType**(문자열) - AMI로 서비스를 실행하는 데 권장되는 인스턴스 유형으로, 원클릭 서비스 설치의 기본값입니다. 인스턴스 유형 목록은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 [유형을](#) 참조하세요.
- **ArchitectureDiagram**(문자열) - Amazon S3의 아키텍처 다이어그램 위치에 대한 URL입니다.
- **템플릿**(문자열) - Amazon S3의 CloudFormation 템플릿 위치에 대한 URL입니다.
- **TemplateSources**(객체 배열)
 - **ParameterName**(문자열) - 이 버전의 AMI를 전달해야 하는 CloudFormation 템플릿의 파라미터 이름입니다. 자세한 내용은 [AMI 세부 정보에 대한 요구 사항](#)을 참조하세요.
- **AmiSource**(객체)
 - **AmiId**(문자열) - API가 호출되는에 있는 소스 AMI의 ID입니다(현재는 카탈로그 API를 사용할 수 AWS 리전있는 유일한 리전이므로 항상 미국 동부(버지니아 북부)여야 함). 호출자 계정에 속해야 합니다.
 - **AccessRoleArn**(문자열) -에서 제공된 AMI에 액세스하는 AWS 리전데 사용하는 IAM 역할 Amazon 리소스 이름(ARN)입니다. 이 ARN 생성 및 사용에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 [AWS Marketplace에 AMI에 대한 액세스 권한 부여를 참조하세요](#).
 - **UserName**(문자열) - 사용자 이름으로 로그인하여 AMI의 운영 체제(OS)에 액세스합니다. 일반적으로 ec2-user Linux AMIs. Administrator
 - **OperatingSystemName**(문자열) - 구매자에게 표시되는 운영 체제의 이름입니다.
 - **OperatingSystemVersion**(문자열) - 구매자에게 표시되는 운영 체제 버전 문자열입니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```


변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 검증 정보가 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

판매자 제품 변경 세트의 오류에 대한 자세한 내용은 섹션을 참조하세요 [설정 상태 및 오류 변경](#).

요청이 완료되면 버전이 추가되고 기존 구독자는 새 버전에 대해 알리는 이메일 메시지를 받게 됩니다. 새 버전을 추가하는 프로세스에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [새 버전 추가](#)를 참조하세요.

비동기 오류

다음 오류는 AddDeliveryOptions 작업과 관련이 있습니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_AMI_NAME	AMI 이름의 접두사(x)가 잘못되었습니다. 회사 또는 브랜드 이름을 접두사로 사용하여 새 이름을 입력합니다.
INVALID_PRODUCT	기존의 제한 또는 공개 제품을 사용합니다.
DUPLICATE_VERSION_TITLE	버전 제목은이 제품의 다른 버전 제목과 달라야 합니다.
INVALID_VERSION_TITLE	상표 기호 앞에 공백을 제거합니다.
INVALID_VERSION_TITLE	지원되지 않는 문자 제거: [x, y, z]
INVALID_VERSION_TITLE	버전 제목의 시작 부분에서 공백을 제거합니다.
INVALID_VERSION_TITLE	[x]자 미만의 버전 제목을 제공합니다.
INVALID_RELEASE_NOTES	상표 기호 앞에 공백을 제거합니다.
INVALID_RELEASE_NOTES	지원되지 않는 문자 제거: [x, y, z]

오류 코드	오류 메시지
INVALID_RELEASE_NOTES	릴리스 정보 시작 부분에서 공백을 제거합니다.
INVALID_RELEASE_NOTES	(x)자 미만의 릴리스 정보를 제공합니다.
INVALID_USAGE_INSTRUCTIONS	상표 기호 앞에 공백을 제거합니다.
INVALID_USAGE_INSTRUCTIONS	지원되지 않는 문자 제거: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	릴리스 정보 시작 부분에서 공백을 제거합니다.
INVALID_USAGE_INSTRUCTIONS	(x)자 미만의 사용 지침을 제공합니다.
RECOMMENDED_INSTANCE_TYPE_NOT_AVAILABLE	사용 가능한 기존 인스턴스 유형을 제공합니다.
INVALID_RECOMMENDED_INSTANCE_TYPE	유효한 인스턴스 유형을 입력합니다.
INVALID_SECURITY_GROUP	보안 그룹 포트는 1에서 [최대] 사이여야 합니다.
INVALID_SECURITY_GROUP	CIDR IP 범위의 값을 제공합니다.
INVALID_SECURITY_GROUP	종료 포트보다 크지 않은 보안 그룹 시작 포트를 제공합니다.
INVALID_SECURITY_GROUP_PROTOCOL	보안 그룹 프로토콜은 'tcp' 또는 'udp'여야 합니다.
INVALID_CIDR_IP	표준 CIDR IP 범위를 '0.0.0.0/0' 형식으로 제공합니다.
INVALID_ACCESS_ENDPOINT_PORT	[x] 미만의 엔드포인트 포트를 제공합니다.
INVALID_ACCESS_ENDPOINT_PORT	1에서 [최대] 사이의 엔드포인트 포트를 제공합니다.
INVALID_ACCESS_ENDPOINT_PORT	엔드포인트 포트를 제공합니다.
INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	상대 경로에서 공백을 제거합니다.

오류 코드	오류 메시지
INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	상대 경로에서 이전 '/'를 제거합니다.
INCOMPATIBLE_OPERATING_SYSTEM	인스턴스 유형과 호환되는 운영 체제 이름 및 버전 제공: [x]
INCOMPATIBLE_OPERATING_SYSTEM_NAME	(x)자 미만의 이름을 입력합니다.
INCOMPATIBLE_OPERATING_SYSTEM_NAME	지원되는 운영 체제 이름을 입력합니다.
INCOMPATIBLE_OPERATING_SYSTEM_VERSION	(x)자 미만의 버전을 제공합니다.
INVALID_SCANNING_PORT	1에서 [최대] 사이의 스캔 포트를 제공합니다.
INVALID_AMI_ID	유효한 AMI ID를 입력합니다.
기존_AMI_PRODUCT_CODE	이미지 x에 연결된 제품 코드를 제거합니다.
INVALID_AMI_ARCHITECTURE	새 AMI에 아키텍처 [x]를 제공합니다.
INVALID_AMI_VIRTUALIZATION_TYPE	가상화 유형 [x]로 새 AMI를 제공합니다.
INVALID_AMI_VIRTUALIZATION_TYPE	이미지 [x]에 예상 [z] 볼륨을 제공합니다.
INCOMPATIBLE_AMI	[y]의 아키텍처 [x]가 다음 인스턴스 유형에서 지원되지 않으므로 새 AMI 제공: [z]
INCOMPATIBLE_AMI	[y]의 가상화 유형 [x]는 다음 인스턴스 유형에서 지원되지 않으므로 새 AMI를 제공합니다. [z]
INCOMPATIBLE_AMI	다음 인스턴스 유형에는 ENA 지원이 필요하므로 이미지 x에 대한 ENA 지원을 활성화합니다. [y]

오류 코드	오류 메시지
ASSET_NOT_FOUND	[account-id]의 us-east-1 리전에 [ami-id]가 있고 AWS 계정 AccessARN 제공된 [ARN]에이 AMI를와 공유할 권한이 있는지 확인합니다AWS Marketplace.
ASSET_ACCESS_EXCEPTION	AMI [x]를 계정에 복사할 수AWS Marketplace 없습니다.
SCAN_ERROR	이미지 [x]의 보안 취약성 [y]를 수정합니다.
ASSET_NOT_FOUND	[account-id]의 us-east-1 리전에 [afi-id]가 있고 AccessARN 제공된 [ARN]에이 AFIAWS 계정을와 공유할 권한이 있는지 확인합니다AWS Marketplace.
ASSET_ACCESS_EXCEPTION	AFI에 액세스할 수 없습니다. AccessARN 제공 [ARN]에이 AFI를와 공유할 수 있는 권한이 AWS Marketplace있고 액세스 역할 계정이 AFI 소유자인지 확인합니다.
MISSING_FPGA_인스턴스_유형	FPGA 기반 인스턴스 유형을 찾을 수 없습니다. AddInstanceTypes 변경 유형을 사용하여 FPGA 기반 인스턴스 유형을 추가합니다.

를 사용하여 AMI 기반 제품을 생성할 수도 있습니다AWS Marketplace Management Portal. 자세한 내용은 AWS Marketplace판매자 안내서의 [AMI 제품을](#) 참조하세요.

AMI 기반 제품 업데이트를 자동화하는 방법을 보여주는 연습은 [카탈로그 API\(5:08\)AWS Marketplace를 사용하여의 제품 목록에 대한 업데이트 자동화](#) 동영상 참조할 수도 있습니다.

버전 정보 업데이트

카탈로그 API를 사용하여 AMI 기반 제품의 기존 버전에 대한 세부 정보를 업데이트할 수 있습니다AWS Marketplace.

Note

를 사용하여 버전 정보를 업데이트하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [버전 정보 업데이트](#)를 AWS Marketplace Management Portal 참조하세요.

버전의 AMI는 업데이트할 수 없습니다. AMI를 업데이트해야 하는 경우 대신 새 버전을 생성합니다.

새 버전을 추가하려면 다음 예제와 같이 UpdateDeliveryOptions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateDeliveryOptions",
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "Version":
        {
          "ReleaseNotes": "*My new Release notes*"
        },
        "DeliveryOptions":
        [
          {
            "Id": "example1-2222-cccc-2222-cccccccccccc",
            "Details":
            {
              "AmiDeliveryOptionDetails":
              {
                "UsageInstructions": "Easy to use AMI"
              }
            }
          }
        ]
      }
    }
  ]
}
```

```

    }
  },
  {
    "Id": "example1-2222-dddd-2222-dddddddddddd",
    "Details":
    {
      "DeploymentTemplateDeliveryOptionDetails":
      {work with ami
        "DeliveryOptionTitle": "My updated delivery option title",
        "UsageInstructions": "Updated usage instructions here."
      }
    }
  }
]
}
]
}

```

다음은 UpdateDeliveryOptions 변경 유형을 추가하기 위해 제공하는 입력 필드에 대한 정보입니다. 이러한 필드에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [버전 정보 업데이트를](#) 참조하세요.

- 개체(객체)(필수) - AMI 기반 제품.
 - 식별자(문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - 유형(문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. `AmiProduct@1.0`.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다. 여기에는 업데이트하려는 AMI 기반 제품의 버전에 대한 모든 정보가 포함됩니다. 포함된 필드는 모두 선택 사항이지만 업데이트할 필드를 하나 이상 포함해야 합니다.
 - 버전(객체) - 소프트웨어 버전에 대한 세부 정보입니다.
 - ReleaseNotes(문자열) - 구매자가 한 버전에서 다음 버전으로의 변경 사항을 알리기 위한 참고 사항입니다.
- DeliveryOptions(객체 배열) - 전송 옵션 객체 목록입니다. 업데이트하려는 전송 옵션만 포함합니다.
 - AMI(독립 실행형) 전송 옵션에서 다음 세부 정보가 포함된 객체를 지정합니다. 업데이트하려는 필드만 포함합니다. 달리 명시되지 않는 한 모든 필드는 선택 사항입니다.

- Id(문자열)(필수) - 전송 옵션의 고유 식별자입니다(업데이트하려는 제품에 대한 DescribeEntity 작업을 호출하여 전송 옵션의 고유 식별자를 가져올 수 있음).
- 세부 정보(객체) - AMI 전송 옵션의 세부 정보를 보유합니다. 이 중첩된 세부 정보 객체는 이중 이스케이프할 필요가 없습니다.
- AmiDeliveryOptionDetails(객체) - 하나의 AMI 전송 옵션에 대한 세부 정보입니다.
 - UsageInstructions(문자열) - AMI 사용에 대한 지침 또는 AMI에 대한 자세한 정보 링크입니다.
 - AccessEndpointUrl(객체) - AMI를 사용한 후 AMI에 액세스할 경로를 생성하는 데 사용됩니다.
 - 포트(문자열) - AMI에서 실행되는 서비스에 액세스하는 데 사용되는 포트 번호입니다.
 - 프로토콜(문자열) - AMI에서 실행되는 서비스에 액세스하는 데 사용되는 프로토콜 (http 또는 https)입니다.
 - RelativePath(문자열) - AMI에서 실행되는 서비스에 액세스하기 위한 웹 루트의 경로입니다(예: */index.html*).
 - RecommendedInstanceType(문자열) - AMI로 서비스를 실행하는 데 권장되는 인스턴스 유형으로, 원클릭 서비스 설치의 기본값입니다.
 - SecurityGroups(객체 배열) - 버전에 대해 자동으로 생성된 그룹의 수신 규칙을 나타내는 객체 목록입니다.
 - FromPort(정수) - 소스 포트입니다.
 - IpProtocol(문자열) - 사용할 프로토콜(tcp 또는 udp).
 - IpRanges(문자열 배열) - CIDR 형식(xxx.xxx.xxx.xxx/nn 형식, 예: *192.0.2.0/24*)으로 허용할 IP 범위입니다.
 - ToPort(정수) - 대상 포트입니다.
- CloudFormation 제공 옵션이 있는 AMI의 경우 다음 세부 정보가 포함된 객체를 지정합니다. 업데이트하려는 필드만 포함합니다. 달리 명시되지 않는 한 모든 필드는 선택 사항입니다.
 - Id(문자열)(필수) - 전송 옵션의 고유 식별자입니다(업데이트하려는 제품에 대한 DescribeEntity 작업을 호출하여 전송 옵션의 고유 식별자를 가져올 수 있음).
 - 세부 정보(객체) - AMI 전송 옵션의 세부 정보를 보유합니다. 이 중첩된 세부 정보 객체는 이중 이스케이프할 필요가 없습니다.
 - DeploymentTemplateDeliveryOptionDetails(객체) -를 사용하여 각 CFT 전송 옵션의 세부 정보를 제공합니다.
 - DeliveryOptionTitle(문자열) - 전송 옵션의 제목입니다.

- ShortDescription(문자열) - CloudFormation 템플릿 전송 옵션에 대한 간단한 설명입니다.
- LongDescription(문자열) - CloudFormation 템플릿 전송 옵션에 대한 자세한 설명입니다.
- UsageInstructions(문자열) - AMI 사용에 대한 지침 또는 AMI에 대한 자세한 정보 링크입니다.
- RecommendedInstanceType(문자열) - AMI로 서비스를 실행하는 데 권장되는 인스턴스 유형으로, 원클릭 서비스 설치의 기본값입니다. 인스턴스 유형 목록은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 [유형을](#) 참조하세요.
- ArchitectureDiagram(문자열) - Amazon S3의 아키텍처 다이어그램 위치에 대한 URL입니다.
- 템플릿(문자열) - Amazon S3의 CloudFormation 템플릿 위치에 대한 URL입니다.
- TemplateSources(객체 배열)
 - ParameterName(문자열) -이 버전의 AMI를 전달해야 하는 CloudFormation 템플릿의 파라미터 이름입니다. 자세한 내용은 [AMI 세부 정보에 대한 요구 사항](#)을 참조하세요.
 - AmiSource(객체)
 - Amild(문자열) - API가 호출되는에 있는 소스 AMI의 ID입니다(현재는 카탈로그 API를 사용할 수 AWS 리전있는 유일한 리전이므로 항상 미국 동부(버지니아 북부)여야 함). 호출자 계정에 속해야 합니다.
 - AccessRoleArn(문자열) -에서 제공된 AMI에 액세스하는 AWS 리전데 사용하는 IAM 역할 Amazon 리소스 이름(ARN)입니다. 이 ARN 생성 및 사용에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 [AMI에 대한 AWS Marketplace 액세스 권한 부여를 참조하세요](#).
 - Username(문자열) - 사용자 이름으로 로그인하여 AMI의 운영 체제(OS)에 액세스합니다. 일반적으로 ec2-user Linux AMIs. Administrator
 - OperatingSystemName(문자열) - 구매자에게 표시되는 운영 체제의 이름입니다.
 - OperatingSystemVersion(문자열) - 구매자에게 표시되는 운영 체제 버전 문자열입니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
```



```
"ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하기 위한 정보 검증이 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

변경 세트에 대한 자세한 내용은 [변경 세트 작업](#)의 내용을 참조하세요. 판매자 제품 변경 세트의 오류에 대한 자세한 내용은 섹션을 참조하세요 [설정 상태 및 오류 변경](#).

비동기 오류

다음 오류는 UpdateDeliveryOptions 작업에만 해당됩니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

변경 세트가 처리를 시작한 후 DescribeChangeSet 작업을 사용하여 UpdatePricingTerms 작업을 검색할 수 있습니다. 오류 세부 정보 및 문제 해결에 대한 자세한 내용은 이 가이드 [설정 상태 및 오류 변경](#) 앞부분의 섹션을 참조하세요.

오류 코드	오류 메시지
INVALID_PRODUCT	기존 제한 또는 공개 제품을 사용합니다.
MISSING_DELIVERY_OPTION_IDS	하나 이상의 전송 옵션 ID를 제공합니다.
INVALID_DELIVERY_OPTION_IDS	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
INVALID_DELIVERY_OPTIONS	동일한 버전에 속하는 전송 옵션 IDs를 제공합니다.

버전 제한

카탈로그 API를 사용하여 AMI 기반 제품의 버전을 제한할 수 있습니다. 이렇게 하면 신규 구매자가 해당 버전을 사용하지 못하지만 기존 구매자가 계속 사용할 수 있습니다. 버전에서 모든 전송 옵션을 제

한하면 버전이 제한됩니다. 버전에서 전부는 아니지만 하나 이상의 전송 옵션을 제한하면 해당 버전의 전송 옵션만 제한됩니다. 항상 사용 가능한 제품의 무제한 버전이 하나 이상 있어야 합니다. 따라서 제품의 마지막 공개적으로 사용 가능한 버전을 제한할 수 없습니다.

Note

를 AWS Marketplace통해에서 AMI 버전을 제한하는 방법에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 버전 제한을 AWS Marketplace Management Portal참조하세요. <https://docs.aws.amazon.com/marketplace/latest/userguide/ami-single-ami-products.html#single-ami-restricting-version>

버전을 제한하려면 다음 예제와 같이 RestrictDeliveryOptions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

모든 구독자는 제한 상태에 관계없이 현재 버전을 사용할 수 있습니다. AWS Marketplace지침에 따라 버전을 제한한 후 90일 동안 기존 구매자에게 지원을 계속 제공해야 합니다. 버전이 제한되면 AMI는 더 이상 사용되지 않는 것으로 표시됩니다. 자세한 내용은 Windows 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 [AMI 사용 중지](#)를 참조하세요.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictDeliveryOptions",
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
```

```
{
  "DeliveryOptionIds":
  [
    "example1-2222-cccc-2222-cccccccccccc"
  ]
}
]
```

다음은 RestrictDeliveryOptions 변경 유형을 추가하기 위해 제공하는 입력 필드에 대한 정보입니다.

- 개체(객체)(필수) - AMI 기반 제품.
 - 식별자(문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - 유형(문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다. AmiProduct@1.0.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다. 여기에는 제한하려는 AMI 기반 제품의 버전에 대한 IDs가 포함됩니다.
 - DeliveryOptionIds(객체 배열) - 제한하려는 버전의 DeliveryOption IDs. 제한하려는 버전에서 DescribeEntity 작업을 호출하여 DeliveryOption ID의 고유 식별자를 가져올 수 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하기 위한 정보 검증이 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

다음 오류는 RestrictDeliveryOptions 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_PRODUCT	기존 퍼블릭 제품을 사용합니다.
MISSING_DELIVERY_OPTION_IDS	하나 이상의 전송 옵션 ID를 제공합니다.
INVALID_DELIVERY_OPTION_IDS	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
INVALID_DELIVERY_OPTION	퍼블릭 상태인 전송 옵션 IDs를 제공합니다. 퍼블릭 상태가 아닌 IDs: [x]
ALL_DELIVERY_OPTIONS_RESTRICTED	하나 이상의 퍼블릭 상태로 유지되어야 하므로 제한할 전송 옵션을 더 적게 제공합니다.

향후 AWS 리전지원 업데이트

카탈로그 API를 사용하여 AMI 기반 제품에 대한 향후 AWS 리전지원 기본 설정을 변경할 수 있습니다. AWS Marketplace.

Note

를 사용하여 향후 리전 지원을 변경하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 향후 AWS 리전에 대한 지원 업데이트를 AWS Marketplace Management Portal 참조하세요. <https://docs.aws.amazon.com/marketplace/latest/userguide/ami-single-ami-products.html#single-ami-updating-future-region-support>

Note

UpdateFutureRegionSupport 변경 유형은 에서만 사용할 수 있습니다. AmiProduct@1.0.

향후 AWS 리전지원 기본 설정을 변경하려면 다음 예제와 같이 UpdateFutureRegionSupport 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateFutureRegionSupport",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "FutureRegionSupport":
        {
          "SupportedRegions":
          [
            "All"
          ]
        }
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateFutureRegionSupport 변경 유형을 추가합니다.

- Entity (객체)(필수) - AMI 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. 다AmiProduct@1.0.
- DetailsDocument (객체)(필수) - ChangeSet를 실행하는 데 필요한 세부 정보입니다.
 - FutureRegionSupport - 객체

- SupportedRegions - 문자열의 단일 요소 배열

요소 지원 값: ["All", "US", "None"] 중 하나

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 검증 정보가 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

다음 오류는 UpdateFutureRegionSupport 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [요변경 세트 작업](#).

오류 코드	오류 메시지
TOO_MANY_REGIONS	현재 FutureRegionSupport: All, US 또는 None에는 1개의 값만 지원됩니다.
INVALID_REGIONS	요청된 리전[a, b, c]이 유효하지 않거나 사용할 수 없습니다. 지원되는 값만 [x, y, z]입니다.
INVALID_INPUT	SupportedRegions은 비워둘 수 없습니다.

지원되는 추가AWS 리전

카탈로그 API를 사용하여 AMI 기반 제품에 AWS 리전지원되는 새를 추가할 수 있습니다AWS Marketplace.

Note

를 사용하여 지원되는 새 리전을 추가하는 방법에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 [AWS 리전 추가](#)를 AWS Marketplace Management Portal참조하세요.

Note

AddRegions 변경 유형은 에서만 사용할 수 있습니다AmiProduct@1.0.

지원되는 새 리전을 추가하려면 다음 예제와 같이 AddRegions 변경 유형을 사용하여 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddRegions",
      "DetailsDocument": {
        "Regions": [
          "us-east-1",
          "ap-northeast-2"
        ]
      },
      "Entity": {
        "Identifier": "prod-123456@1",
        "Type": "AmiProduct@1.0"
      }
    }
  ]
}
```

```
}
```

AddRegions 변경 유형을 추가할 필드에 대한 정보를 제공합니다.

- Entity (객체)(필수) - AMI 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. `AmiProduct@1.0`.
- Example

<caption>DetailsDocument (object) (required) – The details required to execute the ChangeSet.</caption>

- Regions: 문자열 배열

요소 지원 값: 유효한 AWS 리전코드 문자열입니다.

예: `["us-east-1"]`.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 검증 정보가 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

- 변경 세트 실행 상태가 인 경우SUCCEEDED: 새 개체Identifier(또는 EntityId)가 생성됩니다. 제품 개체에서 [DescribeEntity](#) API 작업을 사용하여 결과를 확인할 수 있습니다.
- 변경 세트 실행 상태가 인 경우CLIENT_ERROR: DescribeChangeSet 응답은 오류에 대한 세부 정보와 오류를 해결하기 위해 수행할 해당 작업을 제공합니다.

비동기 오류

다음 오류는 AddRegions 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_REGIONS	요청된 리전[a, b, c]이 유효하지 않거나 사용할 수 없습니다. 지원되는 값만 [x, y, z]입니다.
INVALID_INPUT	리전은 비워둘 수 없습니다.

제한 AWS 리전

카탈로그 API를 사용하여 AMI 기반 제품에 AWS 리전 대해 이전에 지원되는 리전을 제한할 수 있습니다 AWS Marketplace.

Note

를 사용하여 이전에 지원되는 리전을 제한하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [AWS 리전 제한을](#) AWS Marketplace Management Portal참조하세요.

Note

RestrictRegions 변경 유형은 에서만 사용할 수 있습니다 AmiProduct@1.0.

이전에 지원되는 리전을 제한하려면 다음 예제와 같이 RestrictRegions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
```

```

"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType": "RestrictRegions",
    "DetailsDocument": {
      "Regions": [
        "us-east-1",
        "ap-northeast-2"
      ]
    },
    "Entity": {
      "Identifier": "prod-123456@1",
      "Type": "AmiProduct@1.0"
    }
  }
]
}

```

AddRegions 변경 유형을 추가할 필드에 대한 정보를 제공합니다.

- Entity (객체)(필수) - AMI 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. daAmiProduct@1.0.
- DetailsDocument(객체)(필수) - ChangeSet를 실행하는 데 필요한 세부 정보입니다.
 - Regions - 문자열 배열

요소 지원 값:와 같은 유효한 AWS 리전코드 문자열입니다"us-east-1".

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 검증 정보가 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

- 변경 세트 실행 상태가 인 경우 SUCCEEDED- 새 엔티티Identifier(또는 EntityId)가 생성됩니다. 제품 개체DescribeEntity에서 사용하여 결과를 확인할 수 있습니다. 자세한 내용은 [DescribeEntity](#) 단원을 참조하십시오.
- 변경 세트 실행 상태가 인 경우CLIENT_ERROR: DescribeChangeSet 응답은 오류에 대한 세부 정보와 오류를 해결하기 위해 수행할 해당 작업을 제공합니다.

비동기 오류

다음 오류는 AddRegions 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하십시오[변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_REGIONS	요청된 리전[a, b, c]이 유효하지 않거나 사용할 수 없습니다. 지원되는 값만 [x, y, z]입니다.
INVALID_INPUT	리전은 비워둘 수 없습니다.

새 인스턴스 유형 추가

카탈로그 API를 사용하여 AMI 기반 제품의 새 인스턴스 유형을 추가할 수 있습니다AWS Marketplace.

Note

를 사용하여 인스턴스 유형을 추가하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [인스턴스 추가](#)를 AWS Marketplace Management Portal 참조하십시오.

AddInstanceTypes는 제품을 생성할 때 기존 제품과 새로 생성된 제품에 새 인스턴스 유형을 추가합니다. 변경 유형은 제품 문서의 모든 버전을 새 인스턴스 유형으로 업데이트합니다.

Note

AddInstanceTypes 변경 유형은 예에서만 사용할 수 있습니다AmiProduct@1.0.

제한된 인스턴스 유형을 추가할 때 인스턴스 유형을 제한 목록에서 제거하고 사용 가능한 인스턴스 유형 목록에 추가할 수 있습니다. 이를 통해 판매자는 제품 제한을 변경할 수 있습니다. 인스턴스 유형 목록은 교체 가능하며 제품의 영구 제한 상태가 아닙니다.

내부적으로 측정된 제품의 경우 판매자는 별도의 변경 유형을 호출AddDimensions하고 인스턴스 유형에 대한 요금을 UpdatePricingTerms 업데이트해야 합니다.

새 인스턴스 유형을 추가하려면 다음 예제와 같이 RestrictRegions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

AddInstanceTypes 변경 유형만 아래에 나와 있습니다. 내부적으로 측정된 AMI 판매자는 AMI에 대한 유형을 호출AddInstanceTypes하고 UpdatePricingTerms 변경해야 합니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "AddInstanceTypes",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "InstanceTypes":
        [
          "m1.medium"
        ]
      }
    }
  ]
}
```

```

    }
  ]
}
```

필드에 대한 정보를 제공하여 AddInstanceTypes 변경 유형을 추가합니다.

- Entity (객체)(필수) - AMI 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. 다AmiProduct@1.0.
- DetailsDocument(객체)(필수) - ChangeSet이 경우를 실행하는 데 필요한 세부 정보입니다. 다InstanceTypes.
 - InstanceTypes (문자열 배열)(필수) - 제품에 InstanceTypes 추가할 목록입니다. 이러한 인스턴스는 기존에 추가됩니다. InstanceTypes.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하기 위한 정보 검증이 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 AddInstanceTypes 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP
InstanceTypes	필수	422
InstanceTypes	비어 있으면 안 됩니다.	422
InstanceTypes	항목은 1~24자여야 합니다. ^[A-Za-z0-9_-]+\$와 일치해야 합니다.	422
InstanceTypes	항목은 고유해야 합니다.	422
InstanceTypes	항목이 1,500개 이하여야 합니다.	422
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 AddInstanceTypes 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_INSTANCE_TYPES	[x] 인스턴스 유형이 유효하지 않습니다.
INVALID_INSTANCE_가상화	제품의 가상화는 [x]입니다. [x] 인스턴스 유형에서는 지원되지 않습니다.
INVALID_AMI_ARCHITECTURE	제품의 CPU 아키텍처는 '%s'입니다.' + "[x] 인스턴스 유형에서는 지원되지 않습니다.
INCOMPATIBLE_OPERATING_SYSTEM	인스턴스 유형은 제품에 정의된 OS와 호환되지 않습니다. 제품에 정의된 OS와 호환되는 인스턴스 유형을 제공합니다.
INVALID_PRODUCT_TYPE	기존 단일 AMI 제품을 사용합니다.

오류 코드	오류 메시지
INVALID_ENA_SETTING	제품은 ENA를 지원하지 않습니다. ENA 지원은 [x] 인스턴스 유형에 필요합니다.
INVALID_DIMENSIONS	인스턴스 유형에 대해 내부적으로 측정된 차원을 찾을 수 없음: [x]
누락_차원	제품에 대한 차원을 찾을 수 없습니다. AddInstanceTypes 앞에 AddDimensionsAddInstanceTypes 필요합니다.
UPDATE_PRICING_REQUIRED	제품에서 내부적으로 측정된 차원을 사용할 수 있는 경우 UpdatePricingTerms 변경 유형이 필요합니다.

인스턴스 유형 제한

카탈로그 API를 사용하여 AMI 기반 제품에 사용할 수 있는 인스턴스 유형을 제한할 수 있습니다AWS Marketplace.

Note

를 사용하여 사용 가능한 인스턴스 유형을 제한하는 방법에 대한 자세한 내용은 AWS Marketplace판매자 안내서의 [인스턴스 제한](#)을 AWS Marketplace Management Portal참조하세요.

기존 구독자는이 변경의 영향을 받지 않으며 제한된 인스턴스 유형을 사용할 수 있습니다. 그러나 새로운 구매자는 제한된 인스턴스 유형을 사용할 수 없습니다. 현재 인스턴스 유형 구독을 중지하려면(인스턴스 유형이 제한되면) AWS Marketplace판매자 운영 팀에 문의해야 합니다.

RestrictInstanceTypes는 인스턴스 유형을 제품 문서의 모든 버전으로 제한합니다. AddInstanceTypes 변경 유형에서 제품의 모든 버전을 업데이트합니다. 권장 인스턴스 유형을 제한할 수 없습니다. 권장 인스턴스 유형은 버전 수준이므로 판매자가 여러 인스턴스 유형을 제한하지 못할 수 있습니다.

내부적으로 측정된 제품의 경우를 호출할 때 별도의 변경 유형을 호출해야 합니다. RestrictDimensions. 이렇게 하면 제한된 인스턴스 유형에 대한 새 제안이 생성되지 않습니다.

Note

RestrictInstanceTypes 변경 유형은 예에서만 사용할 수 있습니다. AmiProduct@1.0.

AMI 기반 제품에 사용할 수 있는 인스턴스 유형을 제한하려면 다음 예제와 같이 RestrictInstanceTypes 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictInstanceTypes",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument":
      {
        "InstanceTypes":
        [
          "m1.medium"
        ]
      }
    }
  ]
}
```

RestrictInstanceTypes 변경 유형을 추가할 필드에 대한 정보를 제공합니다.

- Entity (객체)(필수) - AMI 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.

- Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. `AmiProduct@1.0`.
- DetailsDocument(객체)(필수) - ChangeSet이 경우를 실행하는 데 필요한 세부 정보입니다. `InstanceTypes`.
- InstanceTypes (문자열 배열)(필수) - 제품으로 제한 `InstanceTypes`할 목록입니다. 이러한 인스턴스는 현재 (또는 기존 인스턴스 유형이 없는 경우 추가됨)에 제한된에 추가됩니다. `InstanceTypes`.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 `ChangeSetArn` 대한 `ChangeSetId` 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하기 위한 정보 검증이 포함되었습니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 `RestrictInstanceTypes` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. `StartChangeSet`. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP
<code>InstanceTypes</code>	필수	422
<code>InstanceTypes</code>	비어 있으면 안 됩니다.	422
<code>InstanceTypes</code>	항목은 1~24자여야 합니다. <code>^[A-Za-z0-9_-]+\$</code> 와 일치해야 합니다.	422

입력 필드	검증 규칙	HTTP
InstanceTypes	항목은 고유해야 합니다.	422
InstanceTypes	항목이 1,500개 이하여야 합니다.	422
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 `RestrictInstanceTypes` 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 `DescribeChangeSet` 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_INSTANCE_TYPES	[x] 인스턴스 유형이 유효하지 않습니다.
INVALID_PRODUCT_TYPE	기존 단일 AMI 제품을 사용합니다.
DUPLICATE_INSTANCE_TYPE	중복이 없는 인스턴스 유형을 제공합니다.
UNAVAILABLE_INSTANCE_TYPE	사용 가능한 인스턴스 유형을 제공합니다.
RECOMMENDED_INSTANCE_TYPE_RESTRICTED	다음 인스턴스 유형은 제한할 수 없습니다. 권장 인스턴스 유형은 제한하기 전에 다른 인스턴스 유형으로 변경해야 합니다. 전송 옵션 ID [X] 인스턴스 유형[X]
DIMENSIONS_NOT_RESTRICTED	내부적으로 측정된 인스턴스 유형을 제한하기 전에 차원 제한: [x]
REGION_NO_INSTANCES	제한된 인스턴스 유형으로 인해 리전에서 제품 시작 실패가 발생합니다. X. 인스턴스 수를 제한하는 것이 좋습니다.
INCOMPATIBLE_RESTRICTION	제한된 인스턴스 유형으로 인해 AFIs에 대한 FPGA 기반 인스턴스 유형이 없는 제품이 발생

오류 코드	오류 메시지
	합니다. 제품은 하나 이상의 FPGA 기반 인스턴스 유형을 지원해야 합니다.

EC2 Image Builder 구성 요소 제품 작업

AWS Marketplace 판매자는 EC2 Image Builder 구성 요소를 사용하여 AWS Marketplace 구매자에게 제공되는 AMI 기반 제품을 나열할 수 있습니다. 구성 요소를 생성하고 AWS Marketplace 목록을 게시하려면 다음 섹션을 순차적으로 진행합니다.

주제

- [Image Builder 구성 요소 빌드 및 테스트](#)
- [구성 요소 ARN 복사](#)
- [AWS Marketplace IAM 정책 생성](#)
- [AWS Marketplace IAM 역할 생성](#)
- [Image Builder 구성 요소 목록 준비](#)
- [Image Builder 구성 요소 제품 목록 게시](#)
- [Image Builder 구성 요소 제품 정보 업데이트](#)
- [기존 Image Builder 구성 요소 제품에 새 버전 추가](#)
- [기존 버전에 대한 정보 업데이트](#)
- [Image Builder 구성 요소 제품 버전 제한](#)
- [변경 세트 모니터링](#)
- [소프트웨어 다운로드 보안](#)

Image Builder 구성 요소 빌드 및 테스트

Image Builder에서 구성 요소를 빌드하고 테스트합니다. 지침은 [Image Builder 사용 설명서의 Image Builder 이미지에 대한 사용자 지정 구성 요소 개발을 참조하세요](#). Image Builder를 사용하여 구성 요소를 생성할 때 다음을 수행해야 합니다.

- Amazon Simple Storage Service(Amazon S3) 버킷, 보안 암호 또는 파라미터와 같은 구성 요소 및 모든 기본 종속성은 미국 동부(버지니아 북부)(us-east-1) AWS 리전에서 생성해야 합니다.
- 구성 요소 설명에 지원되는 아키텍처 및 모든 소프트웨어 종속성을 포함합니다.

- [이미지 파이프라인](#)을 생성하고 파이프라인에서 생성한 AMI를 배포하여 AWS 계정에서 구성 요소를 테스트합니다.
- 구성 요소에 S3 버킷에서 바이너리, 패키지 또는 파일을 복사하는 지침이 포함되어 있는 경우 S3Download 작업 모듈을 사용합니다. S3Download 모듈의 S3 버킷에 있는 파일의 정적 위치를 source 입력합니다. 다음 예제에서는 구성 요소 설치의 일부로 S3 버킷에서 바이너리를 복사합니다.

```
- name: DownloadMyFile
  action: S3Download
  inputs:
    - source: s3://amzn-s3-demo-source-bucket/path/to/package.zip
      destination: C:\myfolder\package.zip
```

- 구성 요소는 S3Download 작업을 사용할 때 최대 2GB의 파일을 수집할 수 있습니다.
- 구성 요소에서 [파라미터](#)를 사용하는 경우 모든 파라미터에 기본값이 있는지 확인합니다. 예를 들어 라는 파라미터가 있는 경우 region와 같은 유효한 기본값이 있는지 확인합니다 us-east-1. 이러한 기본값은 AWS Marketplace 처리 및 테스트를 위한 것입니다. 기본값을 포함하지 않으면 테스트가 실패할 수 있습니다.
- 구성 요소에서 파라미터 스토어 또는 AWS Secrets Manager의 기능을 사용하여 파라미터를 저장하는 경우 다음을 수행합니다 AWS Systems Manager .
- 구성 요소의 한 단계로 값을 검색하려면 YAML 구성 파일에 명령을 포함합니다 AWS Command Line Interface .
- AWS 계정의 Secrets Manager 또는 Parameter Store에서 해당 항목을 생성합니다. 기본 키를 사용하고 AWS Marketplace 테스트 프로세스 중에 구성 요소를 빌드하는 데 도움이 되는 유효한 값을 제공합니다. 예를 들어 Parameter Store를 사용하는 기본값이 saas_token 인 라는 파라미터 token가 있다고 가정해 보겠습니다. 이 경우 Parameter Store에서 키-값 페어를 생성합니다. 를 킠티 token로 사용합니다. 값에 애플리케이션에 유효한 SaaS 토큰을 입력합니다.

AWS Marketplace 판매자 계정에 저장된 값은 AWS Marketplace 테스트 목적으로만 사용됩니다. 이러한 값은 구매자와 공유되지 않습니다.

- AWS Marketplace 는 구성 요소 생성 프로세스 중에 선택한 모든 호환 운영 체제 버전에서 구성 요소에 대한 Amazon Machine Image(AMIs)를 자동으로 생성합니다. 구성 요소를 빌드할 때 호환되는 운영 체제 버전을 하나 이상 선택합니다. EC2 Image Builder 파이프라인을 사용하여 AMIs.

구성 요소 ARN 복사

Image Builder에서 구성 요소를 생성하고 테스트한 후 구성 요소 ARN을 복사하고 저장합니다. AWS Marketplace 카탈로그 API를 사용하여 제품 목록을 게시할 때 ARN을 사용합니다.

Image Builder 구성 요소 ARN을 복사하려면

1. AWS Management Console에 로그인하고에서 Image Builder 콘솔을 엽니다<https://console.aws.amazon.com/imagebuilder/>.
2. 왼쪽 탐색 모음의 저장된 리소스에서 구성 요소를 선택합니다.
3. 구성 요소 페이지의 필터 소유자에서 내 소유를 선택합니다.
4. 구성 요소 이름을 선택합니다.
5. 구성 요소 세부 정보 페이지의 요약 섹션에서 ARN을 복사합니다.

AWS Marketplace IAM 정책 생성

다음 IAM 정책을 생성하여 Image Builder 구성 요소 및 Amazon S3 버킷 및 보안 암호와 같은 관련 리소스에 대한 AWS Marketplace 액세스 권한을 부여합니다. 제공된 예제 정책을 사용합니다. 이러한 정책을 [AWS Marketplace IAM 역할에](#) 연결합니다. 정책 생성에 대한 도움말은 IAM 사용 설명서의 [JSON 편집기를 사용하여 정책 생성](#)을 참조하세요.

- Image Builder에서가 구성 요소에 AWS Marketplace 액세스할 수 있도록 허용하는 Image Builder 구성 요소 가져오기 정책입니다. 이 정책은 필수입니다. 정책 이름을 mp_ib_ingest로 지정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "imagebuilder:GetComponent",
      "Resource": "*"
    }
  ]
}
```

- 가 S3 버킷에서 바이너리를 검색할 수 있도록 허용하는 Amazon S3 읽기 액세스 정책입니다. AWS Marketplace 이 정책은 구성 요소가 S3Download 작업 모듈을 사용하고 연결된 바이너리를 S3 버킷에 저장하는 경우에만 필요합니다. 정책 이름을 mp_ib_s3_read_only로 지정합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListObjectsInBucket",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name"
      ]
    },
    {
      "Sid": "ReadObjectsInBucket",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectAttributes"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name/*"
      ]
    }
  ]
}
```

- 가 Secrets Manager에 저장된 보안 암호를 AWS Marketplace 검색할 수 있도록 허용하는 Secrets Manager 읽기 액세스 정책입니다. 이 정책은 구성 요소가 Secrets Manager를 사용하여 암호를 저장하는 경우에만 필요합니다. 정책 이름을 mp_ib_sm_read_only로 지정합니다. 정책을 보안 암호로만 제한하려면 Resource 섹션의 *를 보안 암호로 바꿉니다.

JSON

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

```

    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}

```

- AWS Marketplace 가 파라미터 스토어에 저장된 보안 암호를 검색할 수 있도록 허용하는 파라미터 스토어 읽기 액세스 정책입니다. 이 정책은 구성 요소가 Parameter Store를 사용하여 암호를 저장하는 경우에만 필요합니다. 정책 이름을 mp_ib_ssm_parameter_read_only로 지정합니다. 정책을 보안 암호로만 제한하려면 Resource 섹션의 *를 보안 암호로 바꿉니다.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "ssm:getParameter"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}

```

AWS Marketplace IAM 역할 생성

다음 절차에 따라 구성 요소 및 해당 종속성에 대한 AWS Marketplace 액세스 권한을 부여하는 정책이 포함된 AWS Marketplace IAM 역할을 생성합니다.

AWS Marketplace IAM 역할을 생성하려면

1. AWS Management Console에 로그인하고 <https://console.aws.amazon.com/iam/> IAM 콘솔을 엽니다.
2. 왼쪽 탐색 모음에서 역할을 선택합니다.
3. 역할 생성을 선택합니다.
4. 사용자 지정 신뢰 정책을 선택합니다.
5. 다음 문을 입력합니다.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "Service": "assets.marketplace.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

6. 다음을 선택합니다.
7. 이전에 생성한 Image Builder get-component 정책을 추가합니다. get-component 정책이 필요합니다. 구성 요소에서 이러한 AWS 서비스를 사용하는 경우 S3, Secrets Manager 및 Parameter Store에 대한 관련 정책을 추가합니다.
8. 다음을 선택합니다.
9. 와 같은 역할 이름을 입력합니다MPEC2IBIngestion.
10. 역할 생성을 선택합니다.

AWS Marketplace IAM 역할 ARN 복사

AWS Marketplace IAM 역할을 생성한 후 역할 ARN을 복사하고 저장합니다. AWS Marketplace 카탈로그 API를 사용하여 목록을 게시할 때 ARN을 사용합니다.

AWS Marketplace IAM 역할 ARN을 복사하려면

1. IAM 콘솔의 왼쪽 탐색 모음에서 역할을 선택합니다.
2. 와 같이 이전에 생성한 AWS Marketplace IAM 역할을 선택합니다MPEC2IBIngestion.
3. 역할 세부 정보 페이지의 요약 섹션에서 ARN을 복사합니다.

Image Builder 구성 요소 목록 준비

AWS Marketplace 목록을 게시하기 전에 다음 정보를 준비해야 합니다.

- 제품 메타데이터 - 메타데이터에는 제품 로고, 제품 제목, 최종 사용자 라이선스 계약, 지원되는 인스턴스 유형 및 AWS 리전이 포함됩니다.
- 요금 정보 - 제품을 무료, 시간당 요금 또는 초기 무료 평가판 기간에 시간당 요금으로 제공할 수 있습니다. 기존 보유 라이선스 사용(BYOL)은 지원되지 않습니다.
- 구성 요소 세부 정보 - 세부 정보에는가 구성 요소를 처리하기 위해 수입할 구성 요소 Amazon 리소스 번호(ARN), 사용 세부 정보 및 AWS Identity and Access Management (IAM) 역할 AWS Marketplace 이 포함됩니다.

Image Builder 구성 요소 제품 목록 게시

이 주제에는 AWS Marketplace 카탈로그 API를 AWS Marketplace 사용하여 EC2 Image Builder 구성 요소 목록을 게시하는 지침이 포함되어 있습니다.

사전 조건

Image Builder 구성 요소 제품 목록을 게시하기 전에 다음이 있는지 확인합니다.

- 에서 판매자로 등록 AWS Marketplace. 자세한 내용은 [AWS Marketplace 판매자로 등록을 참조하세요](#).
- AWSMarketplaceSellerFullAccess 권한이 있는 IAM 사용자.
- 구성 요소에 사용되는 경우 회사 로고와 EULA를 호스팅하기 위한 공개적으로 액세스 가능한 Amazon Simple Storage Service(Amazon S3) 버킷입니다. ChangeSet JSON 파일에 S3 버킷의 URL을 입력합니다.
- AWS Command Line Interface (AWS CLI). 자세한 내용은 [AWS 명령줄 인터페이스란 무엇입니까?](#)를 참조하세요. AWS Command Line Interface 사용 설명서의 .

에서 Image Builder 구성 요소 제품 생성 AWS Marketplace

카탈로그 API를 AWS Marketplace 사용하여서 EC2 Image Builder 구성 요소 제품을 생성하려면 섹션을 참조하세요 [제품 생성](#).

Image Builder 구성 요소 제품 정보 업데이트

AWS Marketplace Management Portal에서 Image Builder 구성 요소 제품에 대한 정보를 업데이트할 수 있습니다.

Image Builder 구성 요소 제품 정보를 업데이트하려면

1. [AWS Marketplace Management Portal](#)을 열고 판매자 계정에 로그인합니다.
2. 제품 메뉴에서 서버를 선택합니다.
3. 서버 제품 페이지에서 제품을 선택합니다.
4. 제품 세부 정보 페이지의 변경 요청 메뉴에서 업데이트하려는 정보에 해당하는 항목을 선택합니다.
5. 변경 사항을 제출하면 요청이 요청 탭의 '검토 중' 상태로 나타나고 완료되면 '성공'으로 변경됩니다.

기존 Image Builder 구성 요소 제품에 새 버전 추가

를 AWS Marketplace 사용하여서 Image Builder 구성 요소 제품에 새 버전을 추가할 수 있습니다 AWS Marketplace Catalog API.

새 버전을 추가하는 방법

1. 에서 제품 ID를 AWS Marketplace Management Portal 가져옵니다.
 - a. [AWS Marketplace Management Portal](#)을 열고 판매자 계정에 로그인합니다.
 - b. 제품 메뉴에서 서버를 선택합니다.
 - c. 서버 제품 페이지에서 제품을 선택합니다.
 - d. 제품 요약에서 제품 ID를 복사합니다.
2. 다음 코드 예제를 사용하여 JSON 형식으로 변경 세트 파일을 생성합니다. 이 예제에서는 *your-product-ID*를 1단계에서 얻은 제품 ID로 바꿉니다. *new-version-name*을 버전 제목으로 바꿉니다. *new-delivery-option-title*을 전송 옵션 제목으로 바꿉니다.

```
[
```

```

{
  "ChangeType": "AddDeliveryOptions",
  "Entity": {
    "Identifier": "your-product-ID",
    "Type": "AmiProduct@1.0"
  },
  "DetailsDocument": {
    "Version": {
      "VersionTitle": "new-version-name",
      "ReleaseNotes": "Release notes goes here."
    },
    "DeliveryOptions": [
      {
        "DeliveryOptionTitle": "new title",
        "Details": {
          "Ec2ImageBuilderComponentDeliveryOptionDetails": {
            "UsageInstructions": "Test usage instructions for IB",
            "AccessRoleArn": "arn:aws:iam::123456789:role/sample",
            "ComponentArn": "arn:aws:imagebuilder:us-east-1:123456789:component/sample/2.0.0/1"
          }
        }
      }
    ]
  }
}

```

3. 이름이 인 변경 세트 파일을 저장합니다 `addIBversion.json`.
4. 터미널 또는에서 다음 명령을 AWS CloudShell 실행합니다.

```
aws marketplace-catalog start-change-set --catalog AWSMarketplace --region us-east-1 --change-set file://addIBversion.json
```

`start-change-set` 명령은 `ChangeSetId` 값을 반환합니다. 변경 세트를 모니터링하려면 섹션을 참조하세요 [변경 세트 모니터링](#).

비동기 오류

다음 오류는의 `AddDeliveryOptions` 작업에만 해당됩니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청

상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

 Note

다음 오류 코드는 Image Builder 전송 방법에 따라 다릅니다. Usage Instructions, Recommended Instance type 및와 같은 필드의 기존 오류 메시지는 [섹션을 참조하세요](#) [새 버전 추가](#).

오류 코드	오류 메시지
ASSET_NOT_FOUND	[account-id] AWS 계정에 [component arn]이 있고 제공된 AccessARN[ARN]에이 Image Builder 구성 요소를와 공유할 권한이 있는지 확인합니다 AWS Marketplace.
ASSET_NOT_FOUND	구성 요소[x] 내에 지정된 S3 링크[x]가 존재하지 않습니다. 구성 요소 내에 유효한 S3 링크를 제공합니다.
DUPLICATE_COMPONENT_NAME	구성 요소 이름 [x]가 이미 동일한 버전 내에 있습니다. 다른 구성 요소 이름을 입력합니다.
DUPLICATE_COMPONENT_NAME	제공한 구성 요소 이름 [x]가 이미 다른 제품에서 사용 중입니다. 다른 구성 요소 이름을 입력합니다.
DUPLICATE_COMPONENT_ARN	구성 요소 Arn [x]는이 제품의 동일한 버전의 다른 전송 옵션에 사용되는 구성 요소 Arn과 달라야 합니다.
DUPLICATE_COMPONENT_ARN	구성 요소 ARN [x]는이 제품의 다른 버전에서 사용되는 구성 요소 ARN과 달라야 합니다.
SCAN_ERROR	구성 요소 [x]의 보안 취약성 [y]를 수정합니다.
ASSET_ACCESS_EXCEPTION	지정된 S3 버킷[x]에서 콘텐츠를 읽는 데 필요한 권한이 없습니다. S3 버킷에 적절한 읽기 권한이 있는지 확인합니다.
ASSET_ACCESS_EXCEPTION	지정된 구성 요소 ARN[x]의 내용을 읽는 데 필요한 권한이 없습니다. 구성 요소 ARN에 적절한 읽기 권한이 있는지 확인합니다.

오류 코드	오류 메시지
ASSET_ACCESS_EXCEPTION	지정된 패키지 URI [x]의 내용을 읽는 데 필요한 권한이 없습니다. 패키지 URI에 적절한 읽기 권한이 있는지 확인합니다.
ASSET_ACCESS_EXCEPTION	HTTP 소스 [x]에서 읽지 못했습니다. 제공된 HTTP 소스가 올바른지 확인합니다.
INVALID_IMAGE_BUILDER_COMPONENT_PACKAGE	구성 요소 자산이 수집 크기 제한을 초과합니다. S3/웹 다운로드 크기를 줄이거나 불필요한 다운로드를 제거하여 계속 진행합니다.
INVALID_IMAGE_BUILDER_COMPONENT_PACKAGE	구성 요소 자산을 수집하는 데 너무 오래 걸립니다. 네트워크 연결이 안정적이고 대역폭이 적절한지 확인합니다.
TOO_MANY_IMAGE_BUILDER_COMPONENTS	제품당 최대 5개의 고유한 구성 요소 이름이 지원되므로 추가 전송 옵션을 제한합니다.
INCOMPLETE_SELLER_PUBLIC_PROFILE	판매자 공개 프로필이 완료되지 않았습니다. 제품에 버전을 추가하기 전에 퍼블릭 프로필을 작성합니다.
INVALID_DESCRIPTION	설명이 누락되었습니다. 구성 요소 [x]에 1,024자 미만의 설명을 입력합니다.
INVALID_COMPONENT_NAME	[x]자 미만의 구성 요소 이름을 입력합니다.
INVALID_SUPPORTED_OS_VERSION	OS 버전이 누락되었습니다. 구성 요소 [x]에 지원되는 유효한 OS 버전을 제공합니다.
INVALID_PATH_FORMAT	[PHASE_NAME] 단계의 [STEP_NAME]단계는 InstallMSI/UninstallMSI 작업에 S3 또는 웹 URLs을 사용할 수 없습니다. 지정된 경로가 작업을 실행하는 시스템에서 액세스할 수 있는 유효한 로컬 경로인지 확인합니다.
INCOMPATIBLE_OS_TYPE	지정된 구성 요소 OS 유형[x]이 기본 이미지의 플랫폼과 호환되지 않습니다. 기본 이미지와 호환되는 OS 유형을 제공합니다.

오류 코드	오류 메시지
ASSET_ACCESS_EXCEPTION	지정된 SSM 파라미터 [x]에서 콘텐츠를 읽는 데 필요한 권한이 없습니다. API 입력에 제공된 IAM 액세스 역할에 필요한 읽기 권한이 있는지 확인합니다.
ASSET_ACCESS_EXCEPTION	지정된 SecretsManager 보안 암호 [x]에서 콘텐츠를 읽는 데 필요한 권한이 없습니다. API 입력에 제공된 IAM 액세스 역할에 필요한 읽기 권한이 있는지 확인합니다.
INVALID_IB_COMPONENT_BUILD_VERSION	구성 요소 ARN [x]의 빌드 버전은 [y]입니다.는 빌드 버전 1 AWS Marketplace 만 지원합니다. 빌드 버전 1을 사용하여 새 EC2 Image Builder 버전을 생성하고 다시 시도하세요. AWS Marketplace
INVALID_IB_COMPONENT_PARAMETER	구성 요소 [y]에 대한 입력 [x]에서 SSM 파라미터를 구문 분석할 수 없습니다.
INVALID_IB_COMPONENT_PARAMETER	구성 요소 [y]에 대한 입력 [x]에서 SecretsManager 보안 암호를 구문 분석할 수 없습니다.
MISSING_IB_COMPONENT_PARAMETER	입력 [y]에 대한 구성 요소 [x]에서 파라미터를 찾을 수 없음
INVALID_IB_COMPONENT_PARAMETER	구성 요소 [y]에 파라미터 [x]의 기본값을 입력합니다.
FAILED_LAUNCH_TEST	구성 요소 [x]에 대한 시작 테스트가 실패했습니다. 오류 메시지: [z]
SSHAuthFailedForUserAndKeypair	OS 기본 사용자 이름 [X]로 인스턴스에 로그인할 수 없습니다.
INVALID_IB_COMPONENT	구성 요소 빌드 단계가 필요합니다. 구성 요소에 유효한 빌드 단계 추가 [x]
DUPLICATE_VERSION_TITLE	버전 제목은이 제품의 다른 버전 제목과 달라야 합니다.

오류 코드	오류 메시지
INVALID_VERSION_TITLE	상표 기호 앞에 공백을 제거합니다.
INVALID_VERSION_TITLE	지원되지 않는 문자 제거: [x, y, z]
INVALID_VERSION_TITLE	버전 제목의 시작 부분에서 공백을 제거합니다.
INVALID_VERSION_TITLE	[x]자 미만의 버전 제목을 제공합니다.
INVALID_RELEASE_NOTES	상표 기호 앞에 공백을 제거합니다.
INVALID_RELEASE_NOTES	지원되지 않는 문자 제거: [x, y, z]
INVALID_RELEASE_NOTES	릴리스 정보 시작 부분에서 공백을 제거합니다.
INVALID_RELEASE_NOTES	(x)자 미만의 릴리스 정보를 제공합니다.
INVALID_USAGE_INSTRUCTIONS	상표 기호 앞에 공백을 제거합니다.
INVALID_USAGE_INSTRUCTIONS	지원되지 않는 문자 제거: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	(x)자 미만의 사용 지침을 제공합니다.
DUPLICATE_DELIVERY_OPTION_TITLES	고유한 전송 옵션 제목을 제공합니다.
INVALID_DELIVERY_OPTION_TITLES	전송 옵션 제목이 이미 있습니다. 다른 제목으로 다시 시도하세요.

기존 버전에 대한 정보 업데이트

기존 버전에 대한 정보를 업데이트하려면

1. 에서 제품 ID를 AWS Marketplace Management Portal 가져옵니다.
 - a. [AWS Marketplace Management Portal](#)을 열고 판매자 계정에 로그인합니다.

- b. 제품 메뉴에서 서버를 선택합니다.
 - c. 서버 제품 페이지에서 제품을 선택합니다.
 - d. 제품 요약에서 제품 ID를 복사합니다.
2. 터미널에서 다음 명령을 실행합니다. 명령에서 *your-product-ID*를 1단계에서 얻은 ID로 바꿉니다.

```
aws marketplace-catalog describe-entity --catalog AWSMarketplace --region us-east-1
--entity-id 'your-product-ID'
```

3. 반환된 출력에서 DetailsDocument, Versions 섹션으로 이동합니다. 업데이트하려는 버전의 Id 값DeliveryOptions인를 복사합니다.
4. 다음 코드 예제를 사용하여 JSON 형식으로 변경 세트 파일을 생성합니다. 이 예제에서는 *your-product-ID*를 1단계에서 얻은 제품 ID로 바꿉니다. *your-release-notes* 릴리스 정보로 바꿉니다. *your-delivery-option-ID*를 3단계에서 얻은 전송 옵션의 ID로 바꿉니다. *your-usage-instructions*를 사용 지침으로 바꿉니다.

```
[
  {
    "ChangeType": "UpdateDeliveryOptions",
    "Entity": {
      "Identifier": "your-product-ID",
      "Type": "AmiProduct@1.0"
    },
    "DetailsDocument": {
      "Version": {
        "ReleaseNotes": "your-release-notes"
      },
      "DeliveryOptions": [
        {
          "Id": "your-delivery-option-ID",
          "Details": {
            "Ec2ImageBuilderComponentDeliveryOptionDetails": {
              "UsageInstructions": "your-usage-instructions"
            }
          }
        }
      ]
    }
  }
]
```


5. 이름이 인 변경 세트 파일을 저장합니다 `updateVersionInfo.json`.
6. 터미널 또는에서 다음 명령을 AWS CloudShell 실행합니다.

```
aws marketplace-catalog start-change-set --catalog AWSMarketplace --region us-east-1 --change-set file:///updateVersionInfo.json
```

`start-change-set` 명령은 `ChangeSetId` 값을 반환합니다. 변경 세트를 모니터링하려면 섹션을 참조하세요 [변경 세트 모니터링](#).

Image Builder 구성 요소 제품 버전 제한

버전을 제한하면 구매자가 버전을 사용할 수 없게 됩니다. AWS Marketplace 카탈로그 API를 AWS Marketplace 사용하여서 Image Builder 구성 요소 제품의 버전을 제한할 수 있습니다. 제품 버전을 하나 이상 무제한으로 유지해야 합니다 AWS Marketplace. 유일한 퍼블릭 버전에 대한 액세스를 제한할 수 없습니다.

기존 버전에 대한 정보를 업데이트하려면

1. 에서 제품 ID를 AWS Marketplace Management Portal 가져옵니다.
 - a. [AWS Marketplace Management Portal](#)을 열고 판매자 계정에 로그인합니다.
 - b. 제품 메뉴에서 서버를 선택합니다.
 - c. 서버 제품 페이지에서 제품을 선택합니다.
 - d. 제품 요약에서 제품 ID를 복사합니다.
2. 터미널에서 다음 명령을 실행합니다. 명령에서 *your-product-ID*를 1단계에서 얻은 ID로 바꿉니다.

```
aws marketplace-catalog describe-entity --catalog AWSMarketplace --region us-east-1 --entity-id 'your-product-ID'
```

3. 반환된 출력에서 `DetailsDocument`, `Versions` 섹션으로 이동합니다. 업데이트하려는 버전의 ID 값 `DeliveryOptions` 인를 복사합니다.
4. 다음 코드 예제를 사용하여 JSON 형식으로 변경 세트 파일을 생성합니다. 이 예제에서는 *your-product-ID*를 1단계에서 얻은 제품 ID로 바꿉니다. *your-delivery-option-ID*를 3단계에서 얻은 전송 옵션의 ID로 바꿉니다.

```
[
```

```
{
  "ChangeType": "RestrictDeliveryOptions",
  "Entity": {
    "Identifier": "your-product-ID",
    "Type": "AmiProduct@1.0"
  },
  "DetailsDocument": {
    "DeliveryOptionIds": [
      "your-delivery-option-ID"
    ]
  }
}
```

5. 이름이 인 변경 세트 파일을 저장합니다 `restrictec2ibversion.json`.
6. 터미널 또는에서 다음 명령을 AWS CloudShell 실행합니다.

```
aws marketplace-catalog start-change-set --catalog AWSMarketplace --region us-east-1 --change-set file://restrictec2ibversion.json
```

`start-change-set` 명령은 `ChangeSetId` 값을 반환합니다. 변경 세트를 모니터링하려면 섹션을 참조하세요 [변경 세트 모니터링](#).

변경 세트 모니터링

`start-change-set` 명령은 `ChangeSetId` 값을 반환합니다. 다음과 같은 방법으로 변경 세트 진행 상황을 모니터링할 수 있습니다.

- 터미널에서 다음 명령을 실행합니다. 명령에서 *changeset-ID*를 `start-change-set` 명령에서 반환된 `ChangeSetId` 값으로 바꿉니다.

```
aws marketplace-catalog describe-change-set --catalog AWSMarketplace --change-set-id changesetID
```

- [AWS Marketplace Management Portal](#)의 요청 탭에서 요청 상태를 확인합니다.

소프트웨어 다운로드 보안

ISV 소프트웨어 지적 재산을 보호하고 AWS Marketplace 구매자에게 안정적이고 일관된 소프트웨어 제공을 보장하기 위해 구성 요소의 `S3Download` 및 `WebDownload` 작업 모듈을 AWS Marketplace 자

동으로 구문 분석합니다. 그러면 참조된 파일이에서 관리하는 프라이빗 Amazon S3 버킷에 안전하게 저장됩니다 AWS Marketplace. 이 수집 프로세스를 옵트아웃하고 소프트웨어 다운로드를 독립적으로 관리하려면 `wget` 또는 `curl` 다운로드 명령을 사용하는 `bash` 스크립트를 실행합니다.

AWS Marketplace APIs를 사용하여 컨테이너 기반 제품 작업

AWS Marketplace 카탈로그 API를 사용하여 컨테이너 기반 제품 작업을 위한 작업을 자동화할 수 있습니다.

카탈로그 API를 사용하여 컨테이너 기반 제품을 생성하는 방법에 대한 자세한 내용은 섹션을 참조하세요 [요제품 생성](#).

다음 주제에서는 카탈로그 API를 사용하여 컨테이너 기반 제품에 대한 작업을 수행하는 방법을 설명합니다.

주제

- [새 버전 추가](#)
- [Amazon EKS 추가 기능에 대한 가시성 업데이트](#)
- [리포지토리 및 리소스 생성](#)
- [버전 정보 업데이트](#)
- [버전 제한](#)

새 버전 추가

컨테이너 기반 제품이 이미 있는 경우 AWS Marketplace를 사용하여 새 버전을 추가할 수 있습니다. 이렇게 하려면 제품의 일부인 각 컨테이너 이미지 또는 아티팩트에 AWS Marketplace 대헤에서 리포지토리를 이미 생성하고 로컬 및 Docker Helm 파일에서 리포지토리를 복사할 수 있어야 합니다.

Note

를 사용하여 컨테이너 기반 제품을 생성하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [컨테이너 제품 시작하기](#)를 AWS Marketplace Management Portal 참조하세요. 를 사용하여 리포지토리 생성, 리포지토리에 Docker 및 Helm 파일 빌드 등 새 버전을 추가하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 새 제품 버전 추가를 AWS Marketplace Management Portal 참조하세요. <https://docs.aws.amazon.com/marketplace/latest/userguide/container-product-getting-started.html#container-add-version>

새 리포지토리를 아직 생성하지 않은 경우 카탈로그 API를 사용하여 생성할 수 있습니다. 섹션을 참조하세요 [리포지토리 및 리소스 생성](#).

새 버전을 추가하려면 다음 예제와 같이 AddDeliveryOptions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

컨테이너 기반 제품의 버전은 하나 이상의 전송 옵션으로 구성됩니다. 예를 들어 사용자가 제품을 사용할 방법을 선택할 수 있도록 noSQL 데이터베이스와 함께 작동하는 전송 옵션과 MySQL과 함께 작동하는 전송 옵션이 두 가지 있을 수 있습니다. 제품 버전을 생성하고를 사용하여 단일 요청으로 여러 전송 옵션을 추가합니다 AddDeliveryOptions.

컨테이너 이미지 전송 요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "AddDeliveryOptions",
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument":
      {
        "Version":
        {
          "VersionTitle": "1.1",
          "ReleaseNotes": "Minor bug fix"
        },
        "DeliveryOptions":
        [
          {
```

```

    "DeliveryOptionTitle": "EKS Container image only delivery option",
    "Details":
    {
        "EcrDeliveryOptionDetails":
        {
            "ContainerImages":
            [
                "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
            ],
            "DeploymentResources":
            [
                {
                    "Name": "HelmDeploymentTemplate",
                    "Url": "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame2:mychart1.1"
                }
            ],
            "CompatibleServices":
            [
                "EKS"
            ],
            "Description": "Sample Description",
            "UsageInstructions": "helm pull 111122223333.dkr.ecr.us-
east-1.amazonaws.com/sellername/reponame2:mychart1.1"
        }
    }
}

```

Amazon Bedrock AgentCore 런타임 전송 요청 구문

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [

```

```

{
  "ChangeType": "AddDeliveryOptions",
  "Entity":
  {
    "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
    "Type": "ContainerProduct@1.0"
  },
  "DetailsDocument":
  {
    "Version":
    {
      "VersionTitle": "1.1",
      "ReleaseNotes": "Minor bug fix"
    },
    "DeliveryOptions":
    [
      {
        "DeliveryOptionTitle": "Amazon Bedrock AgentCore Runtime Delivery Option",
        "Details":
        {
          "EcrDeliveryOptionDetails":
          {
            "ContainerImages":
            [
              "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/  

reponame1:1.1"
            ],
            "CompatibleServices":
            [
              "Bedrock-AgentCore"
            ],
            "AgenticType":
            [
              "AGENT"
            ],
            "Description": "Sample Description",
            "UsageInstructions": "To launch and invoke this agent on Amazon Bedrock  

AgentCore Runtime",
            "EnvironmentVariables":
            [
              {
                "Name": "HTTP_PORT",
                "Description": "Port of the server",
                "DefaultValue": "8080"
              }
            ]
          }
        }
      }
    ]
  }
}

```

```

    },
    {
      "Name": "API_KEY",
      "Description": "Provide your unique API key here."
    }
  ]
}
]
}
]
}
]
}
]
}
}

```

Helm 차트 전송 요청 구문

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "AddDeliveryOptions",
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument":
      {
        "Version":
        {
          "VersionTitle": "1.1",
          "ReleaseNotes": "Minor bug fix"
        },
        "DeliveryOptions":
        [
          {
            "DeliveryOptionTitle": "HelmChartDeliveryOption",
            "Details":

```

```

{
  "HelmDeliveryOptionDetails":
  {
    "CompatibleServices":
    [
      "EKS",
      "EKS-Anywhere"
    ],
    "ContainerImages":
    [
      "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
reponame1:1.1"
    ],
    "HelmChartUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/reponame1:helmchart1.1",
    "Description": "Helm chart description",
    "UsageInstructions": "Usage instructions",
    "MarketplaceServiceAccountName": "Service account name",
    "ReleaseName": "Optional release name",
    "Namespace": "Optional Kubernetes namespace",
    "OverrideParameters":
    [
      {
        "Key": "HelmKeyName1",
        "DefaultValue": "${AWSMP_LICENSE_SECRET}"
      },
      {
        "Key": "HelmKeyName2",
        "DefaultValue": "${AWSMP_SERVICE_ACCOUNT}"
      }
    ]
  }
}

```

Amazon EKS 추가 기능 전송 요청 구문

POST /StartChangeSet HTTP/1.1


```

Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddDeliveryOptions",
      "Entity": {
        "Type": "ContainerProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Version": {
          "VersionTitle": "1.1",
          "ReleaseNotes": "New Add-on Release"
        },
        "DeliveryOptions": [
          {
            "DeliveryOptionTitle": "AWS Marketplace Test AddOn from CAPI 1",
            "Visibility": "Limited",
            "Details": {
              "EksAddOnDeliveryOptionDetails": {
                "ContainerImages": [
                  "111122223333.dkr.ecr.us-east-1.amazonaws.com/test-seller/canary-
test-repo-product-6:mongo"
                ],
                "HelmChartUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/rocket/
rocket-product-helm:1.0",
                "Description": "Description for delivery option provided by ISV",
                "UsageInstructions": "Usage instructions with launch instructions",
                "AddOnName": "aws-mp-test",
                "AddOnVersion": "1.2.1",
                "AddOnType": "networking",
                "CompatibleKubernetesVersions": [
                  "1.25",
                  "1.26"
                ],
                "SupportedArchitectures": [
                  "amd64",
                  "arm64"
                ],
                "Namespace": "my-test-namespace",
                "EnvironmentOverrideParameters": [
                  {
                    "Key": "cluster-name",

```

```

        "Value": "${AWS_EKS_CLUSTER_NAME}"
      },
      {
        "Key": "region-name",
        "Value": "${AWS_REGION}"
      }
    ]
  }
}
],
"ChangeName": "PublishAddonNew"
}
]
}

```

필드에 대한 정보를 제공하여 AddDeliveryOptions 변경 유형을 추가합니다.

- Entity (객체)(필수) - 컨테이너 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. 다ContainerProduct@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다. 여기에는 추가하려는 버전에 대한 모든 정보가 포함됩니다. 이 필드는 문자열 필드입니다.
- Version (객체) - 제품에 추가하려는 버전에 대한 세부 정보입니다.
 - VersionTitle (문자열) - 생성 중인 버전의 제목입니다. 일반적으로 이는 **Version 1.1** 또는 단순히와 같은 버전에 대한 설명입니다**1.1**. 구매자는 버전 제목 목록에서 배포할 버전을 선택할 수 있습니다.
 - ReleaseNotes (문자열) -이 버전에 대한 자세한 참고 사항입니다. 30,000자 미만이어야 합니다.
- DeliveryOptions (객체 배열) - 각가 제품 버전의 전송 방법인 전송 옵션 배열입니다. 예를 들어 Amazon Elastic Container Service(Amazon ECS)에 대한 전송 옵션 하나와 Amazon Elastic Kubernetes Service(Amazon EKS)에 대한 전송 옵션 하나가 있는 경우 두 가지 전송 옵션이 있어야 합니다.
 - DeliveryOptionTitle (문자열) - 구매자가 배송 옵션 중에서 선택할 수 있도록 도와주는 간단한 설명입니다.

- **Details (객체)** -이 전송 옵션에 사용되는 리소스입니다. 이는 세부 정보 필드 내의 세부 정보 필드입니다. 이 필드에서 문자를 두 번 이스케이프할 필요는 없습니다.
- **AddOnName** - 구매자가 Amazon EKS 카탈로그에서 볼 수 있는 고유한 추가 기능 이름입니다. 이 이름은 나중에 사용하여 접두사를 추가합니다 `SellerAlias`. 예를 들어, 여기서 `isv-alias_`는 ISV 제공 추가 기능 이름입니다.
- **AddOnType** - Amazon EKS에서 지원되는 값 목록에서 선택한 추가 기능의 유형: `Gitops` | `monitoring` | `logging` | `cert-management` | `policy-management` | `cost-management` | `autoscaling` | `storage` | `kubernetes-management` | `service-mesh` | `etcd-backup` | `ingress-service-type` | `load-balancer` | `local-registry` | `networking` | `Security` | `backup` | `ingress-controller` | `observability`
- **AddOnVersion** - 구매자가 설치 또는 업그레이드해야 하는 특정 버전의 AddOn을 선택할 수 있는 의미 체계 버전입니다.
- **CompatibleKubernetesVersions** -이 소프트웨어가 호환되는 Amazon EKS Kubernetes 버전입니다.
- **CompatibleServices (문자열 배열)** - 릴리스가 호환되는 서비스의 배열입니다. 유효한 옵션: `ECS`, `EKS`, `EKS-Anywhere`, `ECS-Anywhere` 및 `Bedrock-AgentCore`.
- **ContainerImages (문자열 배열)** -이 버전에서 사용하는 컨테이너 이미지 URLs. 경로는이 버전에서 사용하는 이미지의 태그와 함께 이미지를 업로드한 리포지토리입니다. 목록에는 이전 버전에서 변경되지 않은 이미지까지 포함하여 필요한 모든 이미지가 포함되어야 합니다. 카탈로그 API를 사용하여 리포지토리를 생성하는 방법에 대한 자세한 내용은 다음 섹션을 참조하세요.
- **Description (문자열)** - 구매자에게 세부 정보를 제공하는 전송 옵션에 대한 자세한 설명입니다. 다른 곳에 제공된 추가 지침에 대한 링크를 포함할 수도 있습니다.
- **EcrDeliveryOptionDetails - AgenticType** - 런타임 에이전트의 유형입니다. 유효한 옵션: `AGENT`, `MCP_SERVER` 또는 `A2A_SERVER`.
- **EcrDeliveryOptionDetails - EnvironmentVariables - AgentCore** 런타임 컨테이너에 필요하고 배포 시 구매자에게 미리 채워지는 환경 변수 목록입니다. 각 변수에 대해 컨테이너에서 예상한 이름, 설명 및 선택적 `defaultValue`가 포함된 객체를 제공합니다. 고유한 자격 증명 또는 API 키와 같은 변수의 경우 기본값을 제공하지 마십시오. 설명을 사용하여 변수에 대한 세부 정보와 가능한 값을 지정할 수 있습니다. 기본값이 있는 제공된 모든 변수는 구매자가 제품을 시작할 때 미리 채워집니다.
- **EcrDeliveryOptionDetails - DeploymentResources (array of objects)** - Helm 차트와 같이 버전에 필요한 다른 리소스의 배열입니다. 각 리소스에는 이를 설명하는 `Name`와 리소스를 가리키는 URL이 포함되어 있습니다.

- **EnvironmentOverrideParameters** - 추가 기능에서 사용할 시스템 파라미터 목록입니다. 일부 ISV 제공 AddOn(HelmChart)에는 Amazon EKS 실행 환경 상태(/시스템 정보)에서 파생된 정보가 포함된 구성이 필요할 수 있습니다. 예: EksClusterRegion, EKSClusterName 및 기타. Amazon EKS AddOn 시작 시 이러한 값을 동적으로 대체하여 구매자의 추가 작업을 방지할 수 있습니다. Amazon EKS 시스템은 이미 추가 기능에 대한 시스템 파라미터의 자동 대체를 지원합니다. AWS MarketplaceISV 환경을 확장하여 대체가 필요한 파라미터를 수집할 수 있습니다.

대체할 일반 시스템 정보는 Helm 대체와 유사한 규칙에 따라 AWS Marketplace 지정된 상수를 제공하여 표시할 수 있습니다. 지원되는 값은 `${AWS_REGION}` 및 `${AWS_EKS_CLUSTER_NAME}`입니다.

```
"EnvironmentOverrideParameters" : [ {
  "Key" : "my-field.region"
  "Value" : "${AWS_REGION}"
},
{
  "Key" : "my-second-field"
  "Value" : "${AWS_EKS_CLUSTER_NAME}"
},
```

- **HelmDeliveryOptionDetails - HelmChartUri (string)** - 구매자가 소프트웨어를 시작하기 위해 설치할 Amazon ECR에서 호스팅되는 Helm 차트의 URL입니다.
- **HelmDeliveryOptionDetails - MarketplaceServiceAccountName (string)** - 선택 사항 - Kubernetes 서비스 계정의 이름입니다. 서비스 계정을 호출할 수 있는 권한을 위해 AWS Identity and Access Management(IAM)에 연결하는 데 사용됩니다. AWS 서비스.
- **HelmDeliveryOptionDetails - ReleaseName (string)** - 선택 사항 - 구매자가 소프트웨어를 시작하는 데 사용하는 `helm install` 명령에 제공된 Helm 릴리스의 이름입니다. 포함되지 않은 경우 Helm에서 자동으로 생성된 릴리스 이름을 제공합니다.
- **HelmDeliveryOptionDetails - Namespace (string)** - 선택 사항 - Helm 차트를 설치할 Kubernetes 네임스페이스입니다.
- **HelmDeliveryOptionDetails - OverrideParameters (array of objects)** - 애플리케이션을 시작하는 Helm 명령에 사용되는 파라미터입니다. 구매자는 기본값을 재정의할 수 있습니다.

Note

Amazon EKS Anywhere 제품의 경우 라이선스 보안 암호에 대해 하나 이상의 재정의 파라미터를 제공합니다. DefaultValue를 제공합니다. `"${AWSMP_LICENSE_SECRET}"`.

유료 제품의 경우 서비스 계정 구성에 대한 재정의 파라미터를 하나 이상 제공합니다. DefaultValue를 제공합니다. `"${AWSMP_SERVICE_ACCOUNT}"`.

- Key (문자열) - 점 표기법(`override.example.key`)의 파라미터 키입니다.
- DefaultValue (문자열) -이 재정의 파라미터의 기본값입니다.
- Namespace - 추가 기능 설치를 위한 ISV 제공 네임스페이스입니다.
- SupportedArchitectures - amd64 및 arm64와 같은 지원되는 아키텍처 목록입니다.
- UsageInstructions (문자열) -이 전송 옵션의 사용에 대한 지침을 제공합니다. 최대 4,000 자까지 가능합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청은 대기열에 추가되고 처리되며, 여기에는 컨테이너 이미지 및 기타 정보를 스캔하여 [AWS Marketplace 컨테이너 제품에 대한 지침](#)을 충족하는지 확인하는 작업이 포함됩니다. 이 프로세스는 컨테이너의 수와 크기에 따라 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

변경 세트에 대한 자세한 내용은 [변경 세트 작업](#)의 내용을 참조하세요. 판매자 제품 변경 세트의 오류에 대한 자세한 내용은 [섹션을 참조하세요](#) [설정 상태 및 오류 변경](#).

비동기 오류

다음 오류는 AddDeliveryOptions 작업과 관련이 있습니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT_STATUS	기존의 제한 또는 공개 제품을 사용합니다.
호환되지 않는_서비스	서비스 목록에 호환되지 않는 서비스가 포함되어 있습니다. [incompatible_services] 호환되는 서비스의 유효한 목록을 제공합니다.
NO_SERVICE_SPECIFIED	호환되는 서비스를 하나 이상 제공합니다.
DUPLICATE_COMPATIBLE_AWS_SERVICES	서비스 목록에 중복 항목이 포함되어 있습니다. 이를 제거합니다. 각 항목은 고유해야 합니다.
INVALID_VERSION_TITLE	상표 기호 앞에 공백을 제거합니다.
INVALID_VERSION_TITLE	지원되지 않는 [x, y, z] 문자를 제거합니다.
INVALID_VERSION_TITLE	버전 제목의 시작 부분에서 공백을 제거합니다.
INVALID_VERSION_TITLE	[x]자 미만의 버전 제목을 제공합니다.
DUPLICATE_VERSION_TITLE	버전 제목 [duplicate_version_title]은 중복됩니다. 제목을 제거하거나 변경합니다.
INVALID_RELEASE_NOTES	상표 기호 앞에 공백을 제거합니다.
INVALID_RELEASE_NOTES	지원되지 않는 문자 제거: [x, y, z]
INVALID_RELEASE_NOTES	릴리스 정보 시작 부분에서 공백을 제거합니다.
INVALID_RELEASE_NOTES	(x)자 미만의 릴리스 정보를 제공합니다.
INVALID_USAGE_INSTRUCTIONS	상표 기호 앞에 공백을 제거합니다.

오류 코드	오류 메시지
INVALID_USAGE_INSTRUCTIONS	지원되지 않는 문자 제거: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	(x)자 미만의 사용 지침을 제공합니다.
INVALID_USAGE_INSTRUCTIONS	사용 지침을 제공합니다.
누락_컨테이너_IMAGES	하나 이상의 컨테이너 이미지를 제공합니다.
NO_LICENSE_SECRET_KEYS	Amazon EKS Anywhere 제품의 경우 라이선스 보안 암호에 대한 재정의 파라미터 1개를 제공합니다. DefaultValue 의 요구 사항은 섹션의 예제를 "\${AWSMP_LICENSE_SECRET}" 참조하세요.
TOO_MANY_CONTAINER_IMAGES	50개 미만의 컨테이너 이미지를 제공합니다.
DUPLICATE_CONTAINER_IMAGES	컨테이너 이미지 목록에는 [duplicate_images]라는 중복 이미지가 포함되어 있습니다. 고유한 이미지가 포함된 목록을 제공합니다.
INVALID_CONTAINER_IMAGES	컨테이너 이미지에 유효한 URI를 제공합니다.
INVALID_CONTAINER_IMAGE_URI	이미지 [invalid_image_uri]는 이 제품에 액세스할 수 없습니다. 이미지를 해당 제품 리포지토리에 업로드합니다. 업로드에 대한 자세한 내용은 컨테이너 제품 시작하기 를 참조하세요.
INVALID_CONTAINER_IMAGE_TAG	'최신' 태그를 사용하지 마세요.
DUPLICATE_DELIVERY_OPTION_TITLES	중복 전송 옵션 제목: [duplicate_titles]. 중복 항목 제거
INVALID_DELIVERY_OPTION_TITLES	[existing_titles] 전송 옵션 제목이 이미 있습니다. 다른 제목을 입력합니다.
INVALID_FULFILLMENT_OPTION_TITLE	(x)자 미만의 전송 옵션 제목을 제공합니다.

오류 코드	오류 메시지
NO_SERVICE_ACCOUNT_CONFIGURATION	유료 제품의 경우 서비스 계정 구성에 대한 재정의의 파라미터 1개를 제공합니다. DefaultVa lue 의 요구 사항은 섹션의 예제를 "\${AWSMP_ SERVICE_ACCOUNT}" 참조하세요.
INVALID_DETAILS	제공된 세부 정보가 유효하지 않습니다.
EMPTY_RESOURCE_NAME	리소스 이름을 입력합니다.
EMPTY_RESOURCE_URL	리소스 URL을 제공합니다.
INVALID_RESOURCE_NAME	리소스 이름을 256자 미만으로 입력합니다.
INVALID_RESOURCE_URL	256자 미만의 리소스 URL을 제공합니다.
INVALID_SHORT_DESCRIPTION	1,000자 미만의 짧은 설명을 입력합니다.
INVALID_SHORT_DESCRIPTION	간단한 설명을 제공합니다.
SCAN_ERROR	이미지 ""[x]""의 보안 취약성 ""[y]""를 수정합니 다.
IMAGE_NOT_FOUND	퍼블릭 이미지 URI [invalid_image_uri]가 잘못되 었습니다. 유효한 URI를 입력합니다.
INVALID_ARN	이미지 액세스에 유효한 ARN을 제공합니다.
IMAGE_INACCESSIBLE	이미지 액세스에 유효한 ARN을 제공합니다.
DUPLICATE_ADDON_NAME	제공한 AddOn 이름은 이미 다른 제품에서 사용 중입니다. 다른 이름을 입력합니다.
DUPLICATE_ADDON_VERSION	추가 기능 버전 제목 [duplicate_version_title]이 이미 사용 중입니다. 다른 제목을 입력합니다.
INVALID_ADDON_TYPE	[invalid_types] 추가 기능 유형이 잘못되었습 니다. 지원되는 목록에서 유형을 제공합니다. [eks_addon_do_supported_types].

오류 코드	오류 메시지
INVALID_KUBERNETES_VERSION	Kubernetes 버전 [invalid_versions]이 잘못되었습니다. 지원되는 목록에서 [eks_addon_do_supported_kubernetes_versions] 버전을 제공합니다.
DUPLICATE_KUBERNETES_버전	중복 Kubernetes 버전: [duplicate_versions]. 고유한 버전이 포함된 목록을 제공합니다.
INVALID_ARCHITECTURE	[invalid_Architectures] 아키텍처가 잘못되었습니다. Amazon EKS 지원 아키텍처인 [eks_addon_do_supported_Architectures]의 아키텍처를 제공합니다.
DUPLICATE_SUPPORTED_ARCHITECTURES	중복 아키텍처: [duplicate_Architectures]. 지원되는 고유한 아키텍처 목록을 제공합니다.
INVALID_VISIBILITY_STATE	상태 [invalid_states]가 {EKS_DO} 전송 옵션에 유효하지 않습니다. 허용되는 값인 제한에서 유효한 가시성 상태를 제공합니다.
INVALID_ENVIRONMENT_OVERRIDE_PARAMETER_VALUE	재정의의 파라미터 값 [invalid_values]가 잘못되었습니다. [eks_addon_do_environment_override_parameter_values] 목록에서 유효한 값을 입력합니다.
DUPLICATE_ENVIRONMENT_OVERRIDE_PARAMETER_KEY	환경 재정의의 파라미터에는 [duplicate_keys]라는 중복 키가 포함되어 있습니다. 이를 제거합니다.
TOO_MANY_EKS_ADDON_DELIVERY_OPTIONS	버전에 대해 Amazon EKS 추가 기능 제공 옵션을 하나만 제공합니다.
INCOMPATIBLE_ADDON_NAME	[provided_name] 추가 기능 이름이 기존 이름과 일치하지 않습니다. 이 추가 기능의 퍼블릭 버전 또는 이전 버전에서 기존 이름을 재사용합니다. 각 제품에 대해 하나의 추가 기능 이름만 사용할 수 있습니다.

오류 코드	오류 메시지
INCOMPATIBLE_ADDON_TYPE	추가 기능 유형 [provided_type]이 기존 유형과 일치하지 않습니다. 이 추가 기능의 퍼블릭 버전 또는 이전 버전에서 기존 유형을 재사용합니다. 각 제품에 대해 하나의 추가 기능 유형만 사용할 수 있습니다.
INCOMPATIBLE_ADDON_NAMESPACE	추가 기능 네임스페이스 [provided_namespace]가 기존 네임스페이스와 일치하지 않습니다. 이 추가 기능의 퍼블릭 버전 또는 이전 버전에서 기존 네임스페이스를 재사용합니다. 각 제품에 대해 하나의 추가 기능 네임스페이스만 사용할 수 있습니다.
INVALID_HELM_CHART_URI	차트 Helm URI [invalid_uri]가 잘못되었습니다. SemVer 2 형식으로 URI를 제공합니다.
INCOMPATIBLE_HELM_OBJECTS(INVALID_HELM_OBJECTS)	지원되지 않는 Helm 객체: <unsupported-objects>를 사용하지 않고 Helm 차트를 제공합니다.
INVALID_DEPENDENT_HELM_CHARTS	외부 소스가 아닌 상위 차트 디렉터리에 직접 종속 차트 <invalid-subcharts>가 포함된 차트 Helm을 제공합니다.
INVALID_HELM_SENSITIVE_CONFIG	민감한 정보나 보안 암호 없이 고급 구성 스키마를 제공합니다. 키워드: <sensitive-parameters-identified>
INVALID_HELM_UNDECLARED_IMAGES	요청의 전송 옵션 내에 <list-of-images> Helm 차트 이미지를 제공합니다.
INVALID_HELM_CHART_IMAGES	AddRepositories 변경 유형을 통해 생성된 리포지토리 내의 이미지가 포함된 차트 Helm을 제공합니다. 외부 이미지: <images-identified>.
INVALID_HELM_LINT	Helm lint를 성공적으로 전달하는 Helm 차트를 제공합니다.

오류 코드	오류 메시지
INVALID_HELM_TEMPLATE	Helm 템플릿을 성공적으로 전달하는 Helm 차트를 제공합니다.
INVALID_HELM_CHART	AWS Marketplace 판매자 안내서의 차트 Helm 글머리 기호 목록에 나와 있는 AWS Marketplace 지침을 준수하는 차트 Helm을 제공합니다. https://docs.aws.amazon.com/marketplace/latest/userguide/container-product-policies.html
INVALID_ADDON_NAME	다음 정규식 패턴을 따르는 AddOn 이름을 제공합니다. xx
INVALID_ADDON_NAMESPACE	네임스페이스 값 [invalid_namespaces]이 잘못되었습니다. 네임스페이스는 {EKS_ADDON_NAMESPACE_REGEX} 정규식을 따라야 합니다. 네임스페이스, 네임스페이스 테스트 등을 예로 들 수 있습니다.
INVALID_ADDON_NAME_PATTERN	문자 또는 숫자로 시작하는 추가 기능 이름을 입력한 다음 문자, 숫자 및 하이픈의 조합을 입력합니다. 예: test-addon, eksaddon
INVALID_ADDON_VERSION_PATTERN	"<major>.<minor>.<patch>" 패턴을 사용하여 추가 기능 버전을 제공합니다(예: 1.2.3, 0.1.2, 0.1.1).
EMPTY_DELIVERY_OPTION_IDS	전송 옵션 IDs 목록을 제공합니다.
INVALID_DELIVERY_OPTIONS_INPUT	목록에는 하나 이상의 잘못된 전송 옵션이 포함되어 있습니다. 유효한 목록을 제공하고 각 옵션에 단일 전송 방법이 있는지 확인합니다.
OVERRIDE_PARAMETER_KEYS_CONTAINS_SPECIAL_CHARS	재정의의 파라미터 키 [invalid_keys]에 잘못된 문자가 포함되어 있습니다. 키는 문자, 숫자, 큰따옴표(" ") 및 더하기 기호(+)만 사용해야 합니다.

오류 코드	오류 메시지
INVALID_CONTAINER_IMAGE_REPOSITORY	[invalid_repositories] 리포지토리가 잘못되었습니다. AddRepositories 변경 유형을 통해 생성된 리포지토리를 제공합니다.
유효하지 않은_컨테이너_이미지_태그_형식	컨테이너 이미지 태그 [invalid_image_tag]가 잘못되었습니다. {CONTAINER_IMAGE_TAG_REGEX} 정규식을 준수하는 태그를 제공합니다.
DUPLICATE_OVERRIDE_파라미터_키	재정의의 파라미터에는 중복 키 [duplicate_keys]가 포함되어 있습니다. 중복 항목을 제거합니다.
지원되지 않는_컨테이너_이미지_URI	컨테이너 이미지 [unsupported_image]는 지원되지 않습니다. 이미지 매니페스트 V2, 스키마 1 을 따르는 이미지를 제공합니다.
DUPLICATE_REPOSITORY_NAMES	중복 리포지토리 이름: [duplicate_repo_names]. 고유한 이름을 입력합니다.
INVALID_NAMESPACE	네임스페이스 값 [invalid_namespaces]이 잘못되었습니다. {HELM_RELEASE_PARAMETER_REGEX} 정규식을 준수하는 값을 제공합니다.
INVALID_RELEASE_NAME	releaseName 값 [invalid_release_names]이 잘못되었습니다. {HELM_RELEASE_PARAMETER_REGEX} 정규식을 준수하는 값을 제공합니다.
OVERRIDE_PARAMETER_KEYS_CONTAINS_RESERVED_PARAMETER_KEYS	전송 옵션 제목 [invalid_keys]의 재정의의 파라미터 키는 예약되어 있습니다. 예약 키: [reserved_param_keys]. 다른 키를 제공합니다.

Amazon EKS 추가 기능에 대한 가시성 업데이트

카탈로그 API를 사용하여 제품 버전의 Amazon EKS 추가 기능 제공 옵션에 대한 가시성을 업데이트할 수 있습니다. AWS Marketplace. 컨테이너 제품의 컨테이너 및 Helm 전송 옵션은 '공개' 가시성 상태로 자동으로 생성됩니다.

Note

제품 버전의 가시성을 업데이트하는 기능은 나열된 버전의 Amazon EKS 추가 기능 제공 옵션에서만 지원됩니다. 제품이 아직 '퍼블릭'이 아닌 경우를 사용하여 '퍼블릭' 가시성 상태의 제품을 게시하는 요청을 제출합니다. AWS Marketplace Management Portal.

기본적으로 Amazon EKS 추가 기능 전송 옵션을 사용하여 버전을 생성하면 '제한됨' 상태로 게시됩니다. '제한됨' 상태는 구매자가 Amazon EKS 클러스터에서 사용하고 배포할 수 있도록 모든 리전에서 제품을 공개적으로 사용할 수 없음을 의미합니다. UpdateDeliveryOptionsVisibility 변경 유형으로 StartChangeSet API 작업을 호출하여 전송 옵션의 가시성을 '제한됨'에서 '공개'로 업데이트할 수 있습니다. Amazon EKS 추가 기능 제공 옵션에 해당하는 DeliveryOptions Id 제품 버전을 지정합니다.

요청 본문

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDeliveryOptionsVisibility",
      "Entity": {
        "Identifier": "prod-example12345",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {
        "DeliveryOptions": [
          {
            "Id": "do-1234567891234567891234",
            "TargetVisibility": "Public"
          }
        ]
      }
    }
  ]
}
```

```

    }
  ]
}
}
}
}
}

```

UpdateDeliveryOptionsVisibility 변경 유형을 추가하려면 다음 필드에 대한 정보를 제공합니다.

- Entity (객체)(필수) - 컨테이너 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용하는 제공 방법(제품 유형)을 기반으로 합니다ContainerProduct@1.0.
- DetailsDocument (객체) (필수) - 생성하려는 리포지토리에 대한 정보를 포함하여 요청의 세부 정보입니다. 다음 필드는 모두 필수입니다.
 - DeliveryOptions (객체 목록) - 각 객체의 세부 정보를 포함한 DeliveryOption 객체 목록:
 - Id (문자열) -의 고유 식별자입니다DeliveryOption. (의 고유 식별자를 가져오려면 업데이트하려는 제품에 대한 DescribeEntity 작업을 DeliveryOption호출합니다.
 - TargetVisibility - 제품의 의도한 새로운 가시성입니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공합니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청은 대기열에 추가되고 처리되며, 여기에는 컨테이너 이미지 및 기타 정보를 스캔하여 [AWS Marketplace 컨테이너 제품에 대한 지침](#)을 충족하는지 확인하는 작업이 포함됩니다. 이 프로세스는 컨테이너의 수와 크기에 따라 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 [DescribeChangeSet](#) API 작업을 사용하여 AWS Marketplace Catalog API를 통해 요청 상태를 확인할 수 있습니다.

변경 세트에 대한 자세한 내용은 [변경 세트 작업](#)의 내용을 참조하세요. 판매자 제품 변경 세트의 오류에 대한 자세한 내용은 섹션을 참조하세요 [설정 상태 및 오류 변경](#).

비동기 오류

다음 표에는 AddDeliveryOptions 작업과 관련된 오류가 나와 있습니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
EMPTY_DELIVERY_OPTION_IDS	전송 옵션 IDs 목록을 제공합니다.
INVALID_VISIBILITY_STATE	제공한 TargetVisibility 옵션은 지원되지 않습니다. 허용된 옵션으로 다시 시도하세요. 허용되는 옵션(들)은 다음과 같습니다. Public
INVALID_DELIVERY_OPTION_IDS	잘못된 전송 옵션 세부 정보를 제공했습니다. 제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
DUPLICATE_DELIVERY_OPTION_IDS	중복 전송 옵션 IDs: [duplicate_ids]. 고유한 전송 옵션 IDs 제공합니다.
감사_오류	Amazon EKS 추가 기능 버전(들)과 호환되는 모든 Amazon EKS 클러스터 버전에 대해 독립 소프트웨어 공급업체(ISV) 테스트를 완료하지 않았습니다. 전송 옵션(들)을 릴리스하려면 먼저 테스트를 완료해야 합니다.
INVALID_DELIVERY_OPTION_TYPE	제공한 전송 옵션 유형이 유효하지 않습니다. 전송 옵션이 유형인지 확인하고 다시 EksAddOn 시도하세요.
INCOMPATIBLE_HELM_OBJECTS	지원되지 않는 Helm 객체가 없는 Helm 차트 제공: 지원되지 않는 Helm 객체는 모든 릴리스 객

오류 코드	오류 메시지
	체(.Name 및 .Namespace 제외), Helm 후크 및 조회 함수입니다.
INCOMPATIBLE_ADDON_NAME	추가 기능 이름 [provided_name]이 퍼블릭 버전 이름과 일치하지 않습니다. 릴리스하기 전에 퍼블릭 이름을 업데이트합니다.
INCOMPATIBLE_ADDON_TYPE	추가 기능 유형이 일치하지 않습니다. 이 제품의 퍼블릭 추가 기능 버전 또는 이전 추가 기능 버전에서 기존 추가 기능 유형을 재사용합니다. 각 제품에 대해 하나의 추가 기능만 지원됩니다.
INCOMPATIBLE_ADDON_NAMESPACE	제공된 추가 기능 네임스페이스[provided_namespace]가 퍼블릭 버전 네임스페이스와 일치하지 않습니다. 릴리스하기 전에 추가 기능 네임스페이스를 업데이트합니다.

리포지토리 및 리소스 생성

컨테이너 기반 제품의 새 버전을 생성하려면 AWS Marketplace 리포지토리에서 사용할 수 있는 버전의 리소스가 있어야 합니다. 리포지토리를 생성한 다음 Docker (및) 리소스를 리포지토리로 푸시(업로드Helm)합니다. 를 통해 리포지토리를 생성하는 방법을 알아보려면 AWS Marketplace 판매자 안내서의 새 제품 버전 추가를 AWS Marketplace Management Portal 참조하세요. <https://docs.aws.amazon.com/marketplace/latest/userguide/container-product-getting-started.html#container-add-version>

새 리포지토리를 생성하려면 다음 예제와 같이 AddRepositories 변경 유형 StartChangeSet으로 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
```



```
[
  {
    "ChangeType": "AddRepositories",
    "Entity": {
      "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
      "Type": "ContainerProduct@1.0"
    },
    "DetailsDocument": {
      "Repositories": [
        {
          "RepositoryName": "new-repo-1",
          "RepositoryType": "ECR"
        },
        {
          "RepositoryName": "new-repo-2",
          "RepositoryType": "ECR"
        }
      ]
    }
  }
]
```

필드에 대한 정보를 제공하여 AddRepositories 변경 유형을 추가합니다.

리포지토리 생성에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [새 버전 추가](#)를 참조하세요.

- Entity (객체)(필수) - 컨테이너 기반 제품.
 - 식별자(문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다. ContainerProduct@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다. 여기에는 생성하려는 리포지토리에 대한 정보가 포함됩니다. 포함된 필드는 모두 필수입니다.
 - Repositories (구조 배열) - 리포지토리 객체 목록입니다. 각 리포지토리 객체에는 이름과 유형이 포함됩니다.
 - RepositoryName (문자열) - 생성할 리포지토리의 이름입니다.

- RepositoryType (문자열) - 생성할 리포지토리의 유형입니다. 허용되는 유일한 값은 ECR입니다.

Note

제품당 최대 70개의 리포지토리를 보유할 수 있지만, 여러 리소스와 리소스 버전을 푸시할 때 서로 다른 태그를 제공하여 단일 리포지토리에 추가할 수 있습니다.

하나 이상의 리포지토리를 생성한 후 리소스를 리포지토리에 추가합니다. 리소스를 리포지토리로 푸시하는 방법에 대한 자세한 내용은 Amazon Elastic Container Registry 사용 설명서의 [이미지 푸시](#)를 참조하세요. 리포지토리 중 하나에 필요한 특정 푸시 명령에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [새 버전 추가](#)를 참조하세요.

비동기 오류

다음 오류는 AddRepositories 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_ECR_REPOSITORY_NAME	리포지토리 이름을 'nginx-web-app' 형식으로 제공합니다.
DUPLICATE_ECR_리포지토리_이름	[duplicate_repo_names] 리포지토리가 이미 있습니다. 다른 이름을 선택합니다.
MISSING_REPOSITORY_INFORMATION	리포지토리 이름을 하나 이상 입력합니다.
INVALID_ECR_REPOSITORY_NAME	최대 문자 길이 256에 도달했습니다. 문자 길이 수에는 판매자 네임스페이스가 포함됩니다.

버전 정보 업데이트

카탈로그 API를 사용하여 컨테이너 기반 제품의 기존 버전에 대한 세부 정보를 업데이트할 수 있습니다. AWS Marketplace.

Note

제품을 공개적으로 사용할 수 있는 경우 버전의 버전 제목, 컨테이너 이미지, 전송 옵션 제목 또는 배포 리소스를 업데이트할 수 없습니다. 제품의 이러한 측면을 업데이트해야 하는 경우 대신 새 버전을 생성합니다.

컨테이너 기반 제품의 기존 버전을 업데이트하려면 다음 예제와 같이 UpdateDeliveryOptions 변경 유형으로 StartChangeSet API 작업을 호출합니다. 이렇게 하면 지정한 전송 옵션에 대한 세부 정보와 관련 버전이 업데이트됩니다. 하나 이상의 전송 옵션을 포함해야 합니다.

컨테이너 이미지 전송 요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDeliveryOptions",
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {
        "Version": {
          "ReleaseNotes": "New release notes",
          "VersionTitle": "Version 1.2"
        },
        "DeliveryOptions": [
          {
            "Id": "example4-2222-cccc-2222-cccccccccccc",
            "Details": {
              "EcrDeliveryOptionDetails": {
                "DeliveryOptionTitle": "New Delivery Option Title",
                "Description": "New description",
                "UsageInstructions": "New usage instructions",
                "CompatibleServices": [
                  "EKS"
                ]
              }
            }
          }
        ]
      }
    }
  ]
}
```

```

    }
  }
]
}
]
}

```

Helm 차트 전송 요청 구문

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

```

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDeliveryOptions",
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {
        "Version": {
          "ReleaseNotes": "New release notes",
          "VersionTitle": "Version 1.2"
        },
        "DeliveryOptions": [
          {
            "Id": "example5-2222-cccc-2222-cccccccccccc",
            "Details": {
              "HelmDeliveryOptionDetails": {
                "DeliveryOptionTitle": "New Delivery Option Title",
                "ContainerImages": [
                  "111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/
imagename:1.0"
                ],
                "HelmChartUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/
sellername/helmname:1.0",
                "CompatibleServices": [
                  "EKS-Anywhere"
                ],
                "Description": "New description",

```

```

        "UsageInstructions": "New usage instructions",
        "MarketplaceServiceAccountName": "new-service-account-name",
        "ReleaseName": "new-release-name",
        "Namespace": "new-cluster-namespace",
        "OverrideParameters": [
            {
                "Key": "new.parameter.key",
                "DefaultValue": "New parameter default value"
            }
        ]
    }
}

```

Amazon EKS 추가 기능 전송 요청 구문

POST /StartChangeSet HTTP/1.1
 Content-type: application/json

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDeliveryOptions",
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {
        "Version": {
          "ReleaseNotes": "New release notes",
          "VersionTitle": "Version 1.2"
        },
        "DeliveryOptions": [
          {
            "Id": "example4-2222-cccc-2222-cccccccccccc",
            "Details": {
              "EksAddOnDeliveryOptionDetails": {

```

```
"ContainerImages":[
    "709825985650.dkr.ecr.us-east-1.amazonaws.com/test-seller/canary-test-repo-product-6:mongo"
],
    "Description":"Description for delivery option provided by ISV",
    "UsageInstructions":"Usage instructions with launch instructions",
    "HelmChartUri":"709825985650.dkr.ecr.us-east-1.amazonaws.com/rocket/rocket-product-helm:1.0",
    "AddOnName":"aws-mp-test",
    "AddOnVersion":"1.2.1",
    "AddOnType":"networking",
    "CompatibleKubernetesVersions":[
        "1.19",
        "1.20"
    ],
    "SupportedArchitectures":[
        "amd64",
        "arm64"
    ],
    "Namespace":"my-test-namespace",
    "EnvironmentOverrideParameters":[
        {
            "Key":"my-field",
            "Value":"${AWS_EKS_CLUSTER_NAME}"
        }
    ]
}
}
```

필드에 대한 정보를 제공하여 UpdateDeliveryOptions 변경 유형을 추가합니다.

이러한 필드에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [새 버전 추가](#)를 참조하세요.

- Entity (객체)(필수) - 컨테이너 기반 제품.
- Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.

- **Type (문자열)(필수)** - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. `다ContainerProduct@1.0`.
- **DetailsDocument (객체)(필수)** - 요청의 세부 정보입니다. 여기에는 업데이트하려는 컨테이너 기반 제품의 버전에 대한 모든 정보가 포함됩니다. 포함된 필드는 모두 선택 사항이지만 업데이트할 필드를 하나 이상 포함해야 합니다.
- **Version (객체)** - 소프트웨어 버전에 대한 세부 정보입니다.
 - **VersionTitle (문자열)** - 생성 중인 버전의 제목입니다. 일반적으로 이는 **Version 1.1** 또는 단순히와 같은 버전에 대한 설명입니다**1.1**. 구매자는 모든 버전 제목 목록에서 배포할 버전을 선택할 수 있습니다.

제품이 이미 공개적으로 게시된 경우이 속성을 업데이트할 수 없습니다.

- **ReleaseNotes (문자열)** - 구매자가 한 버전에서 다음 버전으로의 변경 사항에 대해 알리기 위한 참고 사항입니다.
- **DeliveryOptions (객체 목록)** - 각 객체의 세부 정보를 포함한 `DeliveryOption` 객체 목록입니다.
 - **Id (문자열)** -의 고유 식별자`DeliveryOption`(업데이트 중인 제품에 대한 `DescribeEntity` 작업을 호출`DeliveryOption`하여의 고유 식별자를 가져올 수 있음).
 - **Details (객체)** - 전송 옵션의 세부 정보를 보유합니다. 이 중첩된 세부 정보 객체는 이중 이스케이프할 필요가 없습니다.
 - **EcrDeliveryOptionDetails (객체)** - 컨테이너 이미지 전송 옵션의 세부 정보입니다.
 - **DeliveryOptionTitle (문자열)** - 구매자가 배송 옵션 중에서 선택할 수 있는 간단한 설명입니다.

제품이 이미 공개적으로 게시된 경우이 속성을 업데이트할 수 없습니다.

- **ContainerImages (문자열 배열)** -이 버전에서 사용하는 컨테이너 이미지 URLs. 경로는 이 버전에서 사용하는 이미지의 태그와 함께 이미지를 업로드한 리포지토리입니다. 이 필드가 포함된 경우 목록에는 변경되지 않는 이미지까지 포함하여 필요한 모든 이미지가 포함되어야 합니다.

제품이 이미 공개적으로 게시된 경우이 속성을 업데이트할 수 없습니다.

- **DeploymentResources (객체 배열)** - Helm 차트 또는 기타 설명서 링크와 같이 버전에 필요한 다른 배포 리소스의 배열입니다. 각 리소스에는 이를 설명하는 이름과 리소스를 가리키는 URL이 포함됩니다. 해당 버전의 시작 페이지에 링크 목록으로 표시됩니다.

제품이 이미 공개적으로 게시된 경우이 속성을 업데이트할 수 없습니다.

- Name (문자열) - 구매자에게 표시되는 하이퍼링크의 텍스트입니다.
- Url (문자열) - 구매자에게 표시되는 하이퍼링크의 URL입니다.
- CompatibleServices (문자열 배열) - 릴리스가 호환되는 서비스의 배열입니다. 유효한 옵션: ECS, EKS, EKS-Anywhere, ECS-Anywhere 및 Bedrock-AgentCore.
- AgenticType 런타임 에이전트의 유형입니다. 유효한 옵션: AGENT, MCP_SERVER 또는 A2A_SERVER.
- Description (문자열) - 구매자에게 세부 정보를 제공하는 제공 옵션에 대한 자세한 설명입니다. 다른 곳에서 호스팅되는 추가 지침에 대한 링크를 포함할 수도 있습니다.
- UsageInstructions (문자열) - 제품을 배포하고 사용하는 방법에 대한 지침을 제공합니다. 다른 곳에서 호스팅되는 사용 지침에 대한 링크를 추가할 수도 있습니다. 최대 4,000자까지 가능합니다.
- EnvironmentVariables - AgentCore 런타임 컨테이너에 필요하고 배포 시 구매자에게 미리 채워질 환경 변수 목록입니다. 각 변수에 대해 컨테이너에서 예상한 이름, 설명 및 선택적 defaultValue가 포함된 객체를 제공합니다. 고유한 자격 증명 또는 API 키와 같은 변수의 경우 기본값을 제공하지 마십시오. 설명을 사용하여 변수에 대한 세부 정보와 가능한 값을 지정할 수 있습니다. 기본값이 있는 제공된 모든 변수는 구매자가 제품을 시작할 때 미리 채워집니다.
- Id (문자열) -의 고유 식별자DeliveryOption(업데이트 중인 제품에 대한 DescribeEntity 작업을 호출DeliveryOption하여의 고유 식별자를 가져올 수 있음).
- Details (객체) - 전송 옵션의 세부 정보를 보유합니다. 이 중첩된 세부 정보 객체는 이중 이스케이프할 필요가 없습니다.
- HelmDeliveryOptionDetails (객체) - Helm 차트 전송 옵션의 세부 정보입니다.
 - DeliveryOptionTitle (문자열) - 구매자가 배송 옵션 중에서 선택할 수 있는 간단한 설명입니다.

제품이 이미 공개적으로 게시된 경우 이 속성을 업데이트할 수 없습니다.

- ContainerImages (문자열 배열) -이 버전에서 사용하는 컨테이너 이미지 URLs. 경로는 이 버전에서 사용하는 이미지의 태그와 함께 이미지를 업로드한 리포지토리입니다. 목록에는 이전 버전에서 변경되지 않은 이미지까지 포함하여 필요한 모든 이미지가 포함되어야 합니다. 카탈로그 API를 사용하여 리포지토리를 생성하는 방법에 대한 자세한 내용은 다음 섹션을 참조하세요.
- HelmChartUri (문자열) - 구매자가 소프트웨어를 시작하기 위해 설치할 Amazon ECR에서 호스팅되는 Helm 차트의 URL입니다.

- **CompatibleServices** (문자열 배열) - 릴리스가 호환되는 서비스의 배열입니다. 유효한 옵션은 ECS 및 EKS입니다.
- **Description** (문자열) - 구매자에게 세부 정보를 제공하는 제공 옵션에 대한 자세한 설명입니다. 다른 곳에 제공된 추가 지침에 대한 링크를 포함할 수도 있습니다.
- **UsageInstructions** (문자열) - 이 전송 옵션의 사용에 대한 지침을 제공합니다. 최대 4,000자까지 가능합니다.
- **MarketplaceServiceAccountName** (문자열) - Kubernetes 서비스 계정의 이름입니다. 서비스 계정은 서비스 호출 AWS 권한을 AWS Identity and Access Management에 연결하는 데 사용됩니다.
- **ReleaseName** (문자열) - 구매자가 소프트웨어를 시작하는 데 사용하는 `helm install` 명령에 제공된 Helm 릴리스의 이름입니다.
- **Namespace** (문자열) - Helm 차트가 설치될 Kubernetes 네임스페이스입니다.
- **OverrideParameters** (객체 배열) - 애플리케이션을 시작하는 Helm 명령에 사용할 파라미터입니다. 구매자는 기본값을 재정의할 수 있습니다.
 - **Key** (문자열) - 점 표기법의 파라미터 키(`override.example.key`).
 - **DefaultValue** (문자열) - 이 재정의 파라미터의 기본값입니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 `ChangeSetArn` 대한 `ChangeSetId` 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청은 대기열에 추가되고 컨테이너 이미지 및 기타 정보 스캔을 포함하여 처리되어 컨테이너 [AWS Marketplace 제품에 대한 지침](#)을 충족하는지 확인합니다. 이 프로세스는 컨테이너의 수와 크기에 따라 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

변경 세트에 대한 자세한 내용은 [변경 세트 작업](#)의 내용을 참조하세요. 판매자 제품 변경 세트의 오류에 대한 자세한 내용은 [섹션을 참조하세요](#) [설정 상태 및 오류 변경](#).

비동기 오류

다음 오류는 UpdateDeliveryOptions 작업과 관련이 있습니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT_STATUS	기존 제한 또는 공개 제품을 사용합니다.
호환되지 않는_서비스	서비스 목록에 호환되지 않는 서비스가 포함되어 있습니다. [incompatible_services] 호환되는 서비스의 유효한 목록을 제공합니다.
NO_SERVICE_SPECIFIED	호환되는 서비스를 하나 이상 제공합니다.
DUPLICATE_COMPATIBLE_AWS_SERVICES	서비스 목록에 중복 항목이 포함되어 있습니다. 이를 제거합니다. 각 항목은 고유해야 합니다.
INVALID_VERSION_TITLE	상표 기호 앞에 공백을 제거합니다.
INVALID_VERSION_TITLE	지원되지 않는 [x, y, z] 문자를 제거합니다.
INVALID_VERSION_TITLE	버전 제목의 시작 부분에서 공백을 제거합니다.
INVALID_VERSION_TITLE	[x]자 미만의 버전 제목을 제공합니다.
DUPLICATE_VERSION_TITLE	버전 제목 [duplicate_version_title]은 중복됩니다. 제목을 제거하거나 변경합니다.
INVALID_RELEASE_NOTES	상표 기호 앞에 공백을 제거합니다.
INVALID_RELEASE_NOTES	지원되지 않는 문자 제거: [x, y, z]
INVALID_RELEASE_NOTES	릴리스 정보 시작 부분에서 공백을 제거합니다.
INVALID_RELEASE_NOTES	(x)자 미만의 릴리스 정보를 제공합니다.
INVALID_USAGE_INSTRUCTIONS	상표 기호 앞에 공백을 제거합니다.

오류 코드	오류 메시지
INVALID_USAGE_INSTRUCTIONS	지원되지 않는 문자 제거: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	(x)자 미만의 사용 지침을 제공합니다.
INVALID_USAGE_INSTRUCTIONS	사용 지침을 제공합니다.
누락_컨테이너_IMAGES	하나 이상의 컨테이너 이미지를 제공합니다.
TOO_MANY_CONTAINER_IMAGES	50개 미만의 컨테이너 이미지를 제공합니다.
DUPLICATE_CONTAINER_IMAGES	컨테이너 이미지 목록에는 [duplicate_images]라는 중복 이미지가 포함되어 있습니다. 고유한 이미지가 포함된 목록을 제공합니다.
INVALID_CONTAINER_IMAGES	컨테이너 이미지에 유효한 URI를 제공합니다.
INVALID_CONTAINER_IMAGE_URI	이미지 [invalid_image_uri]는 이 제품에 액세스할 수 없습니다. 이미지를 해당 제품 리포지토리에 업로드합니다. 업로드에 대한 자세한 내용은 컨테이너 제품 시작하기 를 참조하세요.
INVALID_CONTAINER_IMAGE_TAG	'최신' 태그를 사용하지 마세요.
MISSING_DELIVERY_OPTION_IDS	기존 ID 목록에서 전송 옵션을 제공합니다.
EMPTY_DELIVERY_OPTION_IDS	비어 있지 않은 전송 옵션 IDs 목록을 제공합니다.
DUPLICATE_DELIVERY_OPTION_IDS	중복 전송 옵션 IDs: [duplicate_ids]. 고유한 전송 옵션 IDs를 제공합니다.
DUPLICATE_DELIVERY_OPTION_TITLES	중복 전송 옵션 제목: [duplicate_titles]. 중복 항목 제거
INVALID_DELIVERY_OPTION_TITLES	[existing_titles] 전송 옵션 제목이 이미 있습니다. 다른 제목을 입력합니다.
INVALID_FULFILLMENT_OPTION_TITLE	(x)자 미만의 전송 옵션 제목을 제공합니다.

오류 코드	오류 메시지
EMPTY_RESOURCE_NAME	리소스 이름을 입력합니다.
EMPTY_RESOURCE_URL	리소스 URL을 제공합니다.
INVALID_RESOURCE_NAME	리소스 이름을 256자 미만으로 입력합니다.
INVALID_RESOURCE_URL	256자 미만의 리소스 URL을 제공합니다.
INVALID_SHORT_DESCRIPTION	1,000자 미만의 짧은 설명을 입력합니다.
INVALID_SHORT_DESCRIPTION	간단한 설명을 제공합니다.
NO_LICENSE_SECRET_KEYS	Amazon EKS Anywhere 제품의 경우 라이선스 보안 암호에 대한 재정의의 파라미터 1개를 제공합니다. 의 요구 사항은 섹션의 예제를 <code>"\${AWSMP_LICENSE_SECRET}"</code> 참조하세요 <code>DefaultValue</code> .
NO_SERVICE_ACCOUNT_CONFIGURATION	유료 제품의 경우 서비스 계정 구성에 대한 재정의의 파라미터 1개를 제공합니다. <code>DefaultValue</code> 의 요구 사항은 섹션의 예제를 <code>"\${AWSMP_SERVICE_ACCOUNT}"</code> 참조하세요.
SCAN_ERROR	이미지 <code>"[x]"</code> 의 보안 취약성 <code>"[y]"</code> 를 수정했습니다.
FIELD_NOT_ALLOWED_TO_CHANGE	[x] 필드는 변경할 수 없습니다.
INVALID_DELIVERY_OPTIONS_STATUS	[invalid_ids] 전송 옵션 IDs가 잘못되었습니다. 제한 또는 퍼블릭 상태에서 전송 옵션을 제공합니다.
NO_CHANGE_FOUND	하나 이상의 변경 사항을 제공합니다.
MULTIPLE_VERSION_UPDATE	동일한 버전의 전송 옵션 IDs를 제공합니다.

오류 코드	오류 메시지
OVERRIDE_PARAMETER_KEYS_CONTAINS_SPECIAL_CHARS	재정의의 파라미터 키[invalid_keys]에 잘못된 문자가 포함되어 있습니다. 키는 문자, 숫자, 큰따옴표(" ") 및 더하기 기호(+)만 사용해야 합니다.
INVALID_CONTAINER_IMAGE_REPOSITORY	[invalid_repositories] 리포지토리가 잘못되었습니다. AddRepositories 변경 유형을 통해 생성된 리포지토리를 제공합니다.
INVALID_CONTAINER_IMAGE_TAG_FORMAT	컨테이너 이미지 태그 [invalid_image_tag]가 잘못되었습니다. {CONTAINER_IMAGE_TAG_REGEX} 정규식을 준수하는 태그를 제공합니다.
DUPLICATE_OVERRIDE_파라미터_키	재정의의 파라미터에는 중복 키 [duplicate_keys]가 포함되어 있습니다. 중복 항목을 제거합니다.
지원되지 않는_컨테이너_이미지_URI	컨테이너 이미지 [unsupported_image]는 지원되지 않습니다. 이미지 매니페스트 V2, 스키마 1 을 따르는 이미지를 제공합니다.
INVALID_NAMESPACE	네임스페이스 값[invalid_namespaces]이 잘못되었습니다. {HELM_RELEASE_PARAM_REGEX} 정규식을 준수하는 값을 제공합니다.
INVALID_RELEASE_NAME	releaseName 값 [invalid_release_names]이 잘못되었습니다. {HELM_RELEASE_PARAM_REGEX} 정규식을 준수하는 값을 제공합니다.
OVERRIDE_PARAMETER_KEYS_CONTAINS_RESERVED_PARAMETER_KEYS	전송 옵션 제목 [invalid_keys]에 대한 재정의의 파라미터 키는 예약되어 있습니다. 예약 키: [reserved_param_keys]. 다른 키를 제공합니다.

오류 코드	오류 메시지
INCOMPATIBLE_ADDON_NAME	추가 기능 이름 [provided_name]이 기존 이름과 일치하지 않습니다. 이 추가 기능의 퍼블릭 버전 또는 이전 버전에서 기존 이름을 재사용합니다. 각 제품에 대해 하나의 추가 기능 이름만 사용할 수 있습니다.
INCOMPATIBLE_ADDON_NAMESPACE	추가 기능 네임스페이스 [provided_namespace]가 기존 네임스페이스와 일치하지 않습니다. 이 추가 기능의 퍼블릭 버전 또는 이전 버전에서 기존 네임스페이스를 재사용합니다. 각 제품에 대해 하나의 추가 기능 네임스페이스만 사용할 수 있습니다.

버전 제한

카탈로그 API를 사용하여 컨테이너 기반 제품의 버전을 제한할 수 있습니다AWS Marketplace. 이렇게 하면 새 구매자가 해당 버전을 사용할 수 없습니다. 제품에는 공개적으로 사용 가능한 버전이 하나 이상 있어야 합니다. 제품에 대해 공개적으로 사용 가능한 나머지 버전만 제한할 수 없습니다.

버전을 제한하려면 다음 예제와 같이 `RestrictDeliveryOptions` 변경 유형으로 `StartChangeSet` API 작업을 호출합니다.

Note

버전에서 전부는 아니지만 하나 이상의 전송 옵션을 제한하면 구매자가 해당 옵션을 사용할 수 없게 됩니다. 버전에 대한 모든 전송 옵션을 제한하면 카탈로그에서 해당 버전이 제거됩니다AWS Marketplace.

Amazon EKS 추가 기능 제한은 현재 카탈로그 API를 통해 지원되지 않습니다.

기존 고객은 제한된 버전을 계속 사용할 수 있습니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictDeliveryOptions",
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {
        "DeliveryOptionIds": [
          "example1-2222-cccc-2222-cccccccccccc"
        ]
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 RestrictDeliveryOptions 변경 유형을 추가합니다.

- Entity (객체)(필수) - 컨테이너 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다. 다ContainerProduct@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다. 여기에는 제한하려는 컨테이너 기반 제품의 전송 옵션에 대한 IDs가 포함됩니다.
 - DeliveryOptionIds (문자열 배열) - 제한하려는 버전의 DeliveryOption IDs 목록입니다. 제한하려는 제품에 대한 DescribeEntity 작업을 호출DeliveryOption하여의 고유 식별자를 가져올 수 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
```

```

"ChangeSetId": "example123456789012abcdef",
"ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 이 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

변경 세트에 대한 자세한 내용은 [변경 세트 작업](#)의 내용을 참조하세요. 판매자 제품 변경 세트의 오류에 대한 자세한 내용은 섹션을 참조하세요 [설정 상태 및 오류 변경](#).

비동기 오류

다음 오류는 RestrictDeliveryOptions 작업에만 해당됩니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT_STATUS	퍼블릭 제품을 사용합니다.
MISSING_DELIVERY_OPTION_IDS	기존 IDs 목록에서 전송 옵션을 제공합니다.
INVALID_DELIVERY_OPTIONS_STATUS	[invalid_ids] 전송 옵션 IDs가 잘못되었습니다. 퍼블릭 상태에서 전송 옵션을 제공합니다.
EMPTY_DELIVERY_OPTION_IDS	비어 있지 않은 전송 옵션 IDs 목록을 제공합니다.
INVALID_MINIMUM_PUBLIC_DELIVERY_OPTIONS	모든 전송 옵션 IDs 제한할 수는 없습니다.
DUPLICATE_DELIVERY_OPTION_IDS	중복 전송 옵션 IDs: [duplicate_ids]. 고유한 전송 옵션 IDs 제공합니다.

AWS Marketplace APIs를 사용하여 SaaS 제품 작업

를 사용하여 SaaS 기반 제품 작업을 위한 작업을 자동화 AWS Marketplace Catalog API 할 수 있습니다. SaaS 기반 제품 생성에 대한 자세한 내용은 섹션을 참조하세요 [제품 생성](#). 다음 주제에서는 카탈로그 API 작업을 수행하는 방법을 설명합니다.

주제

- [SaaS 제품의 무료 요금 모델 구성](#)
- [SaaS 빠른 시작 옵션](#)
- [전송 옵션 추가](#)
- [SaaSUrlDeliveryOption](#)

SaaS 제품의 무료 요금 모델 구성

SaaS 제품은 구매자가 무료로 제품을 사용할 수 있는 무료 요금 모델과 함께 제공될 수 있습니다. SaaS 제품에 무료 요금 모델을 사용하는 경우 UsageBasedPricingTerm 또는 ConfigurableUpfrontPricingTerm에서 제로 달러 요금으로 요금 차원을 구성해야 합니다.

Note

SaaS 제품의 무료 요금 모델 요구 사항은 고유합니다. 무료 요금이 적용되는 SaaS 제품의 경우 UsageBasedPricingTerm 또는 ConfigurableUpfrontPricingTerm을 사용하여 차원을 하나 이상 생성해야 하며 모든 차원의 가격은 0.00 USD여야 합니다. 이 요구 사항은 AMI, 컨테이너 또는 기계 학습 제품에 적용되지 않습니다.

SaaS 제품의 무료 요금을 구성하려면:

1. AddDimensions 변경 유형을 사용하여 제품의 요금 차원을 생성합니다. 자세한 내용은 판매자 제품 설명서 [요금 차원 추가](#)의 섹션을 참조하세요.
2. 가 로 PricingModel 설정된 제안을 Free 생성하고 모든 차원 가격이 0.00 USDConfigurableUpfrontPricingTerm로 설정된 UsageBasedPricingTerm 또는를 포함합니다. 제안 생성에 대한 자세한 내용은 섹션을 참조하세요 [AWS Marketplace APIs를 사용하여 제안 작업](#).

제안을 구성할 때 다음을 확인하세요.

- 의 모든 요금 카드 가격은 "0.00"(문자열 형식)으로 UsageBasedPricingTerm 설정됩니다.
- 의 모든 요금 금액은 "0.00"(문자열 형식)으로 ConfigurableUpfrontPricingTerm 설정됩니다.
- 0.00 USD 요금의 차원이 하나 이상 포함됩니다.

SaaS 빠른 시작 옵션

빠른 시작 옵션에 대한 자세한 내용은 다음 리소스를 참조하세요.

- (구매자) SaaS 제품의 빠른 시작 옵션은 AWS Marketplace구매자 안내서의 [빠른 시작을 사용하여 SaaS 제품 구성 및 시작](#)을 참조하세요.
- (판매자) SaaS 제품의 빠른 시작 옵션은 AWS Marketplace판매자 안내서의 [빠른 시작 구성](#)을 참조하세요.
- SaaS 제품에 대한 빠른 시작 워크숍은 AWS Marketplace 판매자 워크숍의 [랩: SaaS 빠른 시작 활성화](#)를 참조하세요.

전송 옵션 추가

카탈로그 API를 사용하여의 SaaS 제품에 대한 전송 옵션을 추가할 수 있습니다AWS Marketplace. API 전송 옵션을 사용하면 판매자가 Amazon Bedrock for AI 에이전트 워크플로와 같은 서비스와 통합되는 API 기반 AWS서비스를 제공할 수 있습니다.

API 전송 옵션을 추가하려면 다음 예제와 같이 AddDeliveryOptions 변경 유형으로 StartChangeSet API 작업을 호출하여 전송 세부 정보를 추가합니다.

다음 주제에서는 ApiDeliveryOptionDetails 및 SaaSUrlDeliveryOptionDetails 옵션을 사용하여를 추가하는 방법을 설명합니다.

주제

- [ApiDeliveryOptionDetails](#)
- [전송 옵션 업데이트](#)
- [전송 옵션 가시성 업데이트](#)

ApiDeliveryOptionDetails

이 섹션의 예제는 SaaS 제품의 API 엔드포인트, 인증 방법 및 통합 프로토콜을 지정할 수 있는 ApiDeliveryOptionDetails 전송 옵션 유형을 지원합니다.

요청 구문

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json
{
  "ChangeType": "AddDeliveryOptions",
  "Entity": {
    "Identifier": "prod-1234567890123",
    "Type": "SaaSProduct@1.0"
  },
  "Details": {
    "DeliveryOptions": [
      {
        "Details": {
          "ApiDeliveryOptionDetails": {
            "ApiType": "AGENT",
            "QuickLaunchEnabled": true,
            "FulfillmentUrl": "https://example.com/fulfillment",
            "UsageInstructions": "# Getting Started\\n\\n\\nTo use this API:\\n1. Register for an API key\\n2. Configure your endpoint\\n3. Start making requests",
            "CompatibleServices": ["Bedrock-AgentCore"],
            "Endpoints": [
              {
                "Name": "GenerateContent",
                "EndpointUrl": "https://api.example.com/v1/generate",
                "Description": "Generate content using AI models",
                "AuthorizationTypes": ["API_KEY"],
                "Schemas": [{
                  "Type": "OPEN_API",
                  "SchemaUrl": "https://example.com/schemas/example-schema.json"
                }],
                "IntegrationProtocols": [
                  {
                    "Type": "MCP",
                    "UsageInstructions": "Connect using Model Context Protocol for seamless integration"
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}

```

```
}
}
```

필드에 대한 정보를 제공하여 AddDeliveryOptions 변경 유형을 추가합니다.

- Entity (객체)(필수) - SaaS 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 식별자를 참조하세요.
 - Type (문자열)(필수) - 유형은 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다SaaSProduct@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - DeliveryOptions (배열) - 추가되는 API 전송 옵션의 세부 정보입니다.
 - Details (객체) - 추가할 전송 옵션ApiDeliveryOptionDetails의를 포함합니다.
 - ApiDeliveryOptionDetails (객체) - SaaS 제품에 대한 API 전송 옵션 세부 정보를 포함합니다.
 - ApiType (문자열)(필수) - API 제공 유형입니다. 유효한 값:MCP_SERVER ,KNOWLEDGE_BASE ,AGENT ,GUARDRAIL .OTHER
 - QuickLaunchEnabled (부울)(필수) - 구매자가 빠른 시작을 사용하여 소프트웨어를 구성하고 시작할 수 있는지 여부를 결정합니다.
 - FulfillmentUrl (문자열)(필수) - 판매자의 소프트웨어 등록 랜딩 페이지의 URL입니다.
 - UsageInstructions (문자열)(필수) -이 API 전송 옵션 사용에 대한 지침입니다. 마크다운 형식을 지원합니다. 최대 30,000자.
 - CompatibleServices (배열)(선택 사항) -이 전송 옵션에 지원되는 AWS 서비스입니다. 는 현재를 지원합니다Bedrock-AgentCore.
 - Endpoints (배열)(필수) -이 제품에 사용할 수 있는 API 엔드포인트입니다. 정확히 하나의 엔드포인트를 포함해야 합니다.
 - Name (문자열)(선택 사항) - API 엔드포인트의 이름입니다. 패턴과 일치해야 합니다^[A-Za-z][a-zA-Z0-9-]+\$. 최대 100자입니다.
 - EndpointUrl (문자열)(필수) - API 엔드포인트의 URL입니다. 유효한 HTTPS URL이어야 합니다.
 - Description (문자열)(선택 사항) - API 엔드포인트 및 기능에 대한 설명입니다. 최대 4,000자.
 - AuthorizationTypes (배열)(필수) - API 엔드포인트에 액세스하는 데 필요한 권한 부여 유형입니다. 유효한 값: API_KEY, OAUTH2. 1~2개의 고유 값을 포함해야 합니다.
 - Schemas (배열)(선택 사항) - API 엔드포인트의 스키마 사양입니다. 최대 1개의 스키마.

- Type (문자열)(필수) - 스키마 유형입니다. 유효한 값: OPEN_API.
- SchemaUrl (문자열)(필수) - Marketplace 소유 S3 버킷에 수집된 스키마의 S3 URL입니다.
- IntegrationProtocols (배열)(선택 사항) - 엔드포인트에서 지원하는 프로토콜 유형입니다. 최대 2개의 프로토콜.
- Type (문자열)(필수) - 프로토콜 식별자입니다. 유효한 값: MCP, A2A.
- UsageInstructions (문자열)(필수) - 엔드포인트에서 프로토콜을 활용하기 위한 추가 지침입니다. 최대 30,000자.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트의 ID를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

동기 검증

오류 조건	메시지	HTTP 코드
잘못된 API 유형	잘못된 API 유형을 제공했습니다. 유효한 값은 MCP_SERVER, KNOWLEDGE_BASE, AGENT, GUARDRAIL, OTHER입니다.	422
잘못된 이행 URL	"https://"로 시작하는 유효한 이행 URL을 제공합니다.	422
필수 필드 누락	필수 파라미터가 누락되었습니다. ApiType, QuickLaunchEnabled, FulfillmentUrl, UsageInstructions 및 엔드포인트를 제공해야 합니다.	422
잘못된 엔드포인트 URL	"https://"로 시작하는 유효한 엔드포인트 URL을 제공합니다.	422

오류 조건	메시지	HTTP 코드
누락된 권한 부여 유형	하나 이상의 권한 부여 유형을 제공해야 합니다. 유효한 값은 API_KEY, OAUTH2입니다.	422
잘못된 권한 부여 유형	잘못된 권한 부여 유형을 제공했습니다. 유효한 값은 API_KEY, OAUTH2입니다.	422
엔드포인트가 너무 많음	API 전송 옵션에는 엔드포인트를 두 개 이상 제공할 수 없습니다.	422
호환 가능한 서비스가 잘못되었습니다.	잘못된 호환 서비스를 제공했습니다. 유효한 값은 Bedrock-AgentCore입니다.	422
잘못된 스키마 유형	잘못된 스키마 유형을 제공했습니다. 유효한 값은 OPEN_API입니다.	422
잘못된 스키마 URL	Marketplace 소유 S3 버킷을 가리키는 유효한 스키마 URL을 제공합니다.	422
잘못된 통합 프로토콜	잘못된 통합 프로토콜 유형을 제공했습니다. 유효한 값은 MCP, A2A입니다.	422
통합 프로토콜이 너무 많음	2개 이상의 통합 프로토콜을 제공할 수 없습니다.	422
잘못된 사용 지침	사용 지침이 최대 길이인 30,000자를 초과합니다.	422

비동기 오류

오류 코드	오류 메시지
DUPLICATE_DELIVERY_OPTIONS	이 제품에 대해 이미 존재하는 하나 이상의 전송 옵션 유형을 제공했습니다. 고유한 전송 옵션 유형을 제공하거나 기존 전송 옵션을 변경하려는 UpdateDeliveryOptions 경우를 사용합니다.

오류 코드	오류 메시지
INVALID_FULFILLMENT_URL	제공한 URL이 HTTP 상태 코드 [x]를 반환했습니다. 가 200으로 렌더링하는 이행 URL을 제공합니다.
INVALID_ENDPOINT_URL	제공한 URL이 HTTP 상태 코드 [x]를 반환했습니다. 가 200으로 렌더링하는 엔드포인트 URL을 제공합니다.
INVALID_SCHEMA_URL	제공한 스키마 URL이 유효하지 않거나 액세스할 수 없습니다. 소유한 Amazon S3 버킷을 가리키는 유효한 스키마 URL을 제공합니다AWS Marketplace.

전송 옵션 업데이트

카탈로그 API를 사용하여에서 SaaS 제품의 전송 옵션을 업데이트할 수 있습니다AWS Marketplace.

전송 옵션을 업데이트하려면 다음 예제와 같이 UpdateDeliveryOptions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

이는 ApiDeliveryOptionDetails 전송 옵션 유형을 지원합니다. 업데이트할 옵션을 식별하려면 전송 옵션 ID를 제공해야 합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDeliveryOptions",
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {
        "DeliveryOptions": [
```

```

{
  "Id": "do-1234567891234567891234",
  "Details": {
    "ApiDeliveryOptionDetails": {
      "ApiType": "AGENT",
      "QuickLaunchEnabled": true,
      "FulfillmentUrl": "https://example.com/fulfillment-
updated",
      "UsageInstructions": "# Updated Getting Started\\n\\nTo
use this updated API:\\n1. Register for an API key\\n2. Configure your endpoint\\n3.
Start making requests",
      "CompatibleServices": ["Bedrock-AgentCore"],
      "Endpoints": [
        {
          "Name": "GenerateContent",
          "EndpointUrl": "https://api.example.com/v2/
generate",
          "Description": "Generate content using updated
AI models",
          "AuthorizationTypes": ["API_KEY", "OAUTH2"],
          "Schemas": [{
            "Type": "OPEN_API",
            "SchemaUrl": "https://example.com/schemas/
updated-schema.json"
          }],
          "IntegrationProtocols": [
            {
              "Type": "MCP",
              "UsageInstructions": "Connect using
Model Context Protocol for seamless integration"
            }
          ]
        }
      ]
    }
  }
}

```


API 전송 옵션으로 UpdateDeliveryOptions 변경 유형을 업데이트하는 필드에 대한 정보를 제공합니다.

- Entity (객체)(필수) - SaaS 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 식별자를 참조하세요.
 - Type (문자열)(필수) - 유형은 제품이 사용할 전송 방법(제품 유형)을 기반으로 합니다 SaaSProduct@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - DeliveryOptions (배열) - 업데이트 중인 API 전송 옵션의 세부 정보를 포함합니다.
 - Id (문자열)(필수) - DeliveryOption의 고유 식별자입니다. (업데이트하려는 제품에 대한 DescribeEntity 작업을 호출하여 DeliveryOption의 고유 식별자를 가져옵니다.)
 - Details (객체) - 업데이트할 전송 옵션ApiDeliveryOptionDetails의를 포함합니다.
 - ApiDeliveryOptionDetails (객체) - SaaS 제품에 대한 API 전송 옵션 세부 정보를 포함합니다.
 - ApiType (문자열)(필수) - API 제공 유형입니다. 유효한 값:MCP_SERVER ,KNOWLEDGE_BASE ,AGENT ,GUARDRAIL .OTHER
 - QuickLaunchEnabled (부울)(필수) - 구매자가 빠른 시작을 사용하여 소프트웨어를 구성하고 시작할 수 있는지 여부를 결정합니다.
 - FulfillmentUrl (문자열)(필수) - SaaS 제품에 대해 업데이트할 URL입니다.
 - UsageInstructions (문자열)(필수) -이 API 전송 옵션 사용에 대한 지침입니다. 마크다운 형식을 지원합니다. 최대 30,000자.
 - CompatibleServices (배열)(선택 사항) -이 전송 옵션에 지원되는 AWS 서비스입니다. 는 현재를 지원합니다Bedrock-AgentCore.
 - Endpoints (배열)(필수) -이 제품에 사용할 수 있는 API 엔드포인트입니다. 정확히 하나의 엔드포인트를 포함해야 합니다.
 - Name (문자열)(선택 사항) - API 엔드포인트의 이름입니다.
 - EndpointUrl (문자열)(필수) - 업데이트할 API 엔드포인트의 URL입니다.
 - Description (문자열)(선택 사항) - API 엔드포인트 및 기능에 대한 설명입니다.
 - AuthorizationTypes (배열)(필수) - API 엔드포인트에 액세스하는 데 필요한 권한 부여 유형입니다. 유효한 값: API_KEY, OAUTH2.
 - Schemas (배열)(선택 사항) - API 엔드포인트의 스키마 사양입니다.
 - Type (문자열)(필수) - 스키마 유형입니다. 유효한 값: OPEN_API.
 - SchemaUrl (문자열)(필수) - 업데이트된 스키마의 S3 URL입니다.

- IntegrationProtocols (배열)(선택 사항) - 엔드포인트에서 지원하는 프로토콜 유형입니다.
- Type (문자열)(필수) - 프로토콜 식별자입니다. 유효한 값: MCP, A2A.
- UsageInstructions (문자열)(필수) - 엔드포인트에서 프로토콜을 활용하기 위한 추가 지침입니다.

응답 구문

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다. 를 통해 AWS Marketplace Management Portal 또는 DescribeChangeSet 작업을 사용하여 카탈로그 API에서 요청 상태를 확인할 수 있습니다.

동기 검증

오류 조건	메시지	HTTP 코드
빈 전송 옵션 ID	제공된 세부 정보가 유효하지 않습니다. /DeliveryOptions/0/Id의 문자열 "이 필수 스키마 정규식 '^do-[a-zA-Z0-9]+\$'와 일치하지 않습니다.	422
전송 옵션 ID 누락	제공된 세부 정보가 유효하지 않습니다. /DeliveryOptions/0의 JSON에 필수 속성 ['Id']가 없습니다.	422
중복 전송 옵션 ID	고유한 전송 옵션 IDs 제공합니다.	422
잘못된 이행 URL	"https://"로 시작하는 유효한 이행 URL을 제공합니다.	422

오류 조건	메시지	HTTP 코드
잘못된 전송 옵션 IDs	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]	422
다중 URL 전송 옵션	둘 이상의 URL 전송 옵션을 제공했습니다. 하나의 URL 전송 옵션을 제공합니다.	422
전송 옵션 ID 누락	전송 옵션 ID가 누락되었습니다. 업데이트하려는 유효한 전송 옵션 IDs를 하나 이상 제공하거나 새 전송 옵션을 추가하려는 AddDeliveryOptions 경우를 사용합니다.	422
잘못된 시작 URL	"https://"로 시작하는 유효한 시작 URL을 제공합니다.	422
시작 URL 누락	필수 파라미터LaunchUrl 가 누락되었습니다. 를 제공해야 합니다LaunchUrl .	422
배포 템플릿 누락	배포 템플릿이 누락되었습니다. 배포 템플릿을 하나 이상 제공합니다.	422
배포 템플릿이 너무 많음	배포 템플릿은 20개를 초과할 수 없습니다.	422
잘못된 템플릿 URL	빠른 시작 URL이 잘못되었습니다. AWS QuickStarts를 통해 Amazon S3에 게시되는 배포 템플릿 URL을 제공합니다. 잘못된 배포 템플릿 URL: [x]	422
잘못된 배포 템플릿 스택 이름	배포 템플릿 스택 이름이 잘못되었습니다. 영숫자와 하이픈만 사용하여 유효한 스택 이름을 입력합니다. 알파벳 문자로 시작해야 하며 128자를 초과할 수 없습니다.	422

오류 조건	메시지	HTTP 코드
배포 템플릿 제목 복제	중복 배포 템플릿 제목을 제공했습니다. 고유한 배포 템플릿 제목을 제공합니다.	422
중복 배포 템플릿 URL	중복 배포 템플릿 URL을 제공했습니다. 고유한 배포 템플릿 URL을 제공합니다.	422
잘못된 배포 템플릿 유형	배포 템플릿 유형이 잘못되었습니다. 유효한 배포 템플릿 유형을 제공합니다. 지원되는 값은 ["CloudFormation@1.0"]입니다.	422
잘못된 배포 템플릿 IAM 정책	배포 템플릿 IAM 정책이 잘못되었습니다. 유효한 IAM 정책을 제공합니다.	422
잘못된 사용 지침	- 이미지는 사용 지침에서 지원되지 않습니다. 이미지 [x]를 제거합니다. - 사용 지침에서 잘못된 URL에 대한 링크를 제공했습니다. [x]. 유효한 URL을 제공합니다. - 사용 지침에서 지원되지 않는 URI 체계가 포함된 링크를 제공했습니다. 지원되는 체계를 사용합니다. ["http", "https", "tel", "mailto"].	422

비동기 오류

오류 코드	오류 메시지
INVALID_DELIVERY_OPTION_IDS	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
감사_오류	AWSMP 카탈로그 감사 목록 - CQ 팀

오류 코드	오류 메시지
INVALID_FULFILLMENT_URL	제공한 URL이 HTTP 상태 코드 [x]를 반환했습니다. 가 200으로 렌더링하는 이행 URL을 제공합니다.
INVALID_LAUNCH_URL	반환된 HTTP 상태 코드 [x]를 제공한 URL입니다. 가 200으로 렌더링하는 시작 URL을 제공합니다.
INVALID_TEMPLATE_URL	빠른 시작 URL이 잘못되었습니다. Amazon S3에 AWSQuickStarts를 통해 게시되는 배포 템플릿 URL을 제공합니다. 잘못된 배포 템플릿 URL: [x]

전송 옵션 가시성 업데이트

일부 사용자만 SaaS 제품의 가시성을 변경할 수 있도록 카탈로그 API를 사용하여 권한을 구성할 수 있습니다AWS Marketplace.

일부 사용자만 SaaS 제품의 가시성을 변경할 수 있도록 권한을 구성하려면 다음 예제와 같이 UpdateDeliveryOptionsVisibility 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

이는 하나의 전송 옵션인 에서만 지원됩니다SaaSUrlDeliveryOptionDetails.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateDeliveryOptionsVisibility",
      "Entity":
```

```

{
  "Identifier": "prod-example12345",
  "Type": "SaaSProduct@1.0"
},
"DetailsDocument":
{
  "DeliveryOptions":
  [
    {
      "Id": "do-1234567891234567891234",
      "TargetVisibility": "Public"
    },
    {
      "Id": "do-43210987654321",
      "TargetVisibility": "Limited",
      "Targeting":
      {
        "PositiveTargeting":
        {
          "BuyerAccounts":
          [
            "123456789012"
          ]
        }
      }
    }
  ]
}
}

```

필드에 대한 정보를 제공하여 UpdateDeliveryOptionsVisibility 변경 유형을 추가합니다.

- Entity (객체)(필수) - SaaS 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다
다SaaSProduct@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - DeliveryOptions (배열) - 업데이트할 목록DeliveryOptions입니다.
 - TargetVisibility (문자열) - 업데이트할 전송 옵션 ID입니다.

- TargetVisibility (문자열) - 전송 옵션의 의도한 가시성입니다.

가능한 값: Limited, Public 및 Unavailable

Note

항상 정확히 하나의 Public 전송 옵션과 최대 하나의 Limited 전송 옵션이 있습니다.

- Targeting (객체) 선택 사항 - Limited 가시성을 로 변경하기 전에 새 전송 옵션을 테스트할 수 있도록 가시성 상태와 함께 사용되는 전송 옵션을 대상으로 지정합니다Public.
- PositiveTargeting (객체) - 포괄적인 대상 지정.
 - BuyerAccounts (문자열 배열) - 새 전송 옵션을 사용할 수 있는 구매자AWS 계정 ID 목록입니다.

최소 크기: 0. 최대 크기: 100.

응답 구문

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다. 를 통해 AWS Marketplace Management Portal또는 DescribeChangeSet 작업을 사용하여 카탈로그 API에서 요청 상태를 확인할 수 있습니다.

동기 검증

오류 조건	메시지	HTTP 코드
전송 옵션 ID 누락	전송 옵션 ID가 누락되었습니다. 업데이트하려는 유효한 전송 옵션 IDs를 하나 이상 제공하거나 새 전송 옵션을 추가하려는 AddDeliveryOptions 경우를 사용합니다.	422

오류 조건	메시지	HTTP 코드
잘못된 가시성	에 잘못된 옵션을 제공했습니다TargetVisibility . 허용되는 옵션은 Limited, Public, 입니다Unavailable .	422
잘못된 대상 지정	에 잘못된 옵션을 제공했습니다PositiveTargeting .에 유효한 파라미터를 제공해야 합니다BuyerAccounts .	422
가시성 및 대상 지정 누락	잘못된 전송 옵션 가시성 세부 정보를 제공했습니다. TargetVisibility 또는 중 하나 이상에 유효한 파라미터를 제공해야 합니다Targeting .	422
ID가 너무 AWS 계정 많음	대상 구매자 계정은 100개를 초과할 수 없습니다.	422

비동기 오류

오류 코드	오류 메시지
INVALID_DELIVERY_OPTION_IDS	잘못된 전송 옵션 세부 정보를 제공했습니다. 제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
INVALID_VISIBILITY	퍼블릭 상태에 대해 둘 이상의 전송 옵션을 제공했습니다. 퍼블릭 전송 옵션을 하나만 제공합니다.
INVALID_VISIBILITY	퍼블릭 전송 옵션을 제공하지 않았습니다. 하나의 퍼블릭 전송 옵션을 제공합니다.
감사_오류	MCO 수동 검토에 따라 다릅니다.

SaaSUrlDeliveryOption

다음 예제에서는를 사용하여 업데이트SaaSUrlDeliveryOptionDetails하는 방법을 보여줍니다FulfillmentUrl.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDeliveryOptions",
      "Entity": {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {
        "DeliveryOptions": [
          {
            "Id": "do-1234567891234567891234",
            "Details": {
              "SaaSUrlDeliveryOptionDetails": {
                "FulfillmentUrl": "https://www.aws.amazon.com/
marketplace/management",
                "LaunchUrl": "URL",
                "UsageInstructions": "Instructions",
                "DeploymentTemplates": [
                  {
                    "Title": "CloudFormation Template 123",
                    "Description": "CloudFormation description",
                    "IamPolicy": "{\n\"Version\":\n\"2012-10-17\",
                    \"Statement\": [{\n\"Effect\":\n\"Allow\",
                    \"Action\": [\n\"s3:Get*\",\n\"s3:List*\"],
                    \"Resource\":\n[\n\"arn:aws:s3:::amzn-s3-demo-bucket\",
                    \"arn:aws:s3:::amzn-s3-demo-bucket/*\"
                    ]}]}"
                  }
                ]
              }
            }
          ]
        }
      }
    ]
  }
}
```

```

    }
  ]
}

```

필드에 대한 정보를 제공하여 AddDeliveryOptions 변경 유형을 추가합니다.

- Entity (객체)(필수) - SaaS 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다SaaSProduct@1.0.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
 - DeliveryOptions (배열) - 업데이트 중인 전송 옵션의 세부 정보를 포함합니다.
 - TargetVisibility (문자열) -의 고유 식별자입니다DeliveryOption. (업데이트하려는 제품에 대한 DescribeEntity 작업을 호출DeliveryOption하여의 고유 식별자를 가져옵니다.)
 - Details (객체) - 업데이트할 전송 옵션SaaSUrlDeliveryOptionDetails의를 포함합니다.
 - SaaSUrlDeliveryOptionDetails (객체) - SaaS 제품에 대한 제공 옵션FulfillmentUrl의를 포함합니다.
 - FulfillmentUrl (문자열) - SaaS 제품에 대해 업데이트할 URL입니다.
 - LaunchUrl (문자열) - SaaS 제품 랜딩 페이지의 URL입니다. QuickLaunchEnabled가 True로 설정된 경우 이 데이터베이스는 필수입니다.
 - UsageInstructions (문자열) -이 전송 옵션 사용에 대한 지침입니다. 를 사용하지 않는 고객을 위한 수동 단계에 대한 설명서를 포함합니다DeploymentTemplates.
 - DeploymentTemplates (배열) - 고객이 SaaS 제품 및 관련 AWS리소스를 설정하고 구성하는 데 사용할 수 있는 배포 템플릿입니다.
 - Title (문자열) - 배포 템플릿의 표시 이름입니다.
 - Description (문자열) - 배포 템플릿에 포함된 항목에 대한 설명입니다.
 - IamPolicy (문자열) - 템플릿을 배포하는 데 필요한 권한을 설명하는 IAM 정책입니다. 구매자는이 IAM 정책을 사용하여 템플릿을 빠르게 배포할 수 있습니다.
 - CloudFormationDetails (객체) - CloudFormation 템플릿의 세부 정보입니다.
 - TemplateUrl (문자열) - 배포 템플릿의 URL입니다.
 - DefaultStackName (문자열) - 고객이 템플릿을 생성할 CloudFormation때에 사용되는 기본 이름입니다.

응답 구문

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다. 를 통해 AWS Marketplace Management Portal 또는 DescribeChangeSet 작업을 사용하여 카탈로그 API에서 요청 상태를 확인할 수 있습니다.

동기 검증

오류 조건	메시지	HTTP 코드
빈 전송 옵션 ID	제공된 세부 정보가 유효하지 않습니다. /DeliveryOptions/0/Id의 문자열 "가 필수 스키마 정규식 '^do-[a-zA-Z0-9]+\$'와 일치하지 않습니다.	422
전송 옵션 ID 누락	제공된 세부 정보가 유효하지 않습니다. /DeliveryOptions/0의 JSON에 필수 속성이 없습니다. ['Id'].	422
중복 전송 옵션 ID	고유한 전송 옵션 IDs 제공합니다.	422
잘못된 이행 URL	"https://"로 시작하는 유효한 이행 URL을 제공합니다.	422
잘못된 전송 옵션 IDs	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]	422
다중 URL 전송 옵션	둘 이상의 URL 전송 옵션을 제공했습니다. 하나의 URL 전송 옵션을 제공합니다.	422
전송 옵션 ID 누락	전송 옵션 ID가 누락되었습니다. 업데이트하려는 유효한 전송 옵션 IDs를	422

오류 조건	메시지	HTTP 코드
	하나 이상 제공하거나 새 전송 옵션을 추가하려는 AddDeliveryOptions 경우를 사용합니다.	
잘못된 시작 URL	"https://"로 시작하는 유효한 시작 URL을 제공합니다.	422
시작 URL 누락	필수 파라미터LaunchUrl 가 누락되었습니다. 를 제공해야 합니다LaunchUrl .	422
배포 템플릿 누락	배포 템플릿이 누락되었습니다. 배포 템플릿을 하나 이상 제공합니다.	422
배포 템플릿이 너무 많음	배포 템플릿은 20개 이상 제공할 수 없습니다.	422
잘못된 템플릿 URL	빠른 시작 URL이 잘못되었습니다. AWS QuickStarts를 통해 Amazon S3에 게시되는 배포 템플릿 URL을 제공합니다. 잘못된 배포 템플릿 URL: [x]	422
잘못된 배포 템플릿 스택 이름	배포 템플릿 스택 이름이 잘못되었습니다. 영숫자와 하이픈만 사용하여 유효한 스택 이름을 입력합니다. 알파벳 문자로 시작해야 하며 128자를 초과할 수 없습니다.	422
배포 템플릿 제목 복제	중복 배포 템플릿 제목을 제공했습니다. 고유한 배포 템플릿 제목을 제공합니다.	422
배포 템플릿 URL 복제	중복 배포 템플릿 URL을 제공했습니다. 고유한 배포 템플릿 URL을 제공합니다.	422

오류 조건	메시지	HTTP 코드
잘못된 배포 템플릿 유형	배포 템플릿 유형이 잘못되었습니다. 유효한 배포 템플릿 유형을 제공합니다. 지원되는 값은 ["CloudFormation@1.0"]입니다.	422
잘못된 배포 템플릿 IAM 정책	배포 템플릿 IAM 정책이 잘못되었습니다. 유효한 IAM 정책을 제공합니다.	422
잘못된 사용 지침	- 이미지는 사용 지침에서 지원되지 않습니다. 이미지 [x]를 제거합니다. - 사용 지침에서 잘못된 URL에 대한 링크를 제공했습니다. [x]. 유효한 URL을 제공합니다. - 사용 지침에서 지원되지 않는 URI 체계가 포함된 링크를 제공했습니다. 지원되는 체계를 사용합니다. ["http", "https", "tel", "mailto"].	422

비동기 오류

오류 코드	오류 메시지
INVALID_DELIVERY_OPTION_IDS	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
감사_오류	AWSMP 카탈로그 감사 목록 - CQ 팀
INVALID_FULFILLMENT_URL	제공한 URL이 반환된 HTTP 상태 코드 [x]입니다. 가 200으로 렌더링하는 이행 URL을 제공합니다.
INVALID_LAUNCH_URL	제공한 URL이 반환된 HTTP 상태 코드 [x]입니다. 가 200으로 렌더링하는 시작 URL을 제공합니다.

오류 코드	오류 메시지
INVALID_TEMPLATE_URL	빠른 시작 URL이 잘못되었습니다. Amazon S3에 AWSQuickStarts를 통해 게시되는 배포 템플릿 URL을 제공합니다. 잘못된 배포 템플릿 URL: [x]

전송 옵션 가시성 업데이트

일부 사용자만 SaaS 제품의 가시성을 변경할 수 있도록 카탈로그 API를 사용하여 권한을 구성할 수 있습니다 AWS Marketplace.

이렇게 하려면 다음 예제와 같이 UpdateDeliveryOptionsVisibility 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

이는 하나의 전송 옵션인 에서만 지원됩니다 SaaSUrlDeliveryOptionDetails.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDeliveryOptionsVisibility",
      "Entity": {
        "Identifier": "prod-example12345",
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {
        "DeliveryOptions": [
          {
```

```

      "Id": "do-1234567891234567891234",
      "TargetVisibility": "Public"
    },
    {
      "Id": "do-43210987654321",
      "TargetVisibility": "Limited",
      "Targeting": {
        "PositiveTargeting": {
          "BuyerAccounts": [
            "123456789012"
          ]
        }
      }
    }
  ]
}

```

필드에 대한 정보를 제공하여 UpdateDeliveryOptionsVisibility 변경 유형을 추가합니다.

- Entity (객체)(필수) - SaaS 기반 제품.
 - Identifier (문자열)(필수) - 제품 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - Type는 제품이 사용할 제공 방법(제품 유형)을 기반으로 합니다SaaSProduct@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - DeliveryOptions (배열) - 업데이트할 목록DeliveryOptions입니다.
 - TargetVisibility (문자열) - 업데이트할 전송 옵션 ID입니다.
 - TargetVisibility (문자열) - 전송 옵션의 의도한 새로운 가시성입니다.

가능한 값: Limited, Public 및 Unavailable

Note

항상 정확히 하나의 Public 전송 옵션과 최대 하나의 Limited 전송 옵션이 있습니다.

- Targeting (객체) 선택 사항 - Limited 가시성을 로 변경하기 전에 새 전송 옵션을 테스트할 수 있도록 가시성 상태와 함께 사용되는 전송 옵션을 대상으로 지정합니다Public.
- PositiveTargeting (객체) - 포괄적인 대상 지정.
 - BuyerAccounts (문자열 배열) - 새 전송 옵션을 사용할 수 있는 구매자AWS 계정 ID 목록입니다.

최소 크기: 0. 최대 크기: 100.

응답 구문

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다. 를 통해 AWS Marketplace Management Portal또는 DescribeChangeSet 작업을 사용하여 카탈로그 API에서 요청 상태를 확인할 수 있습니다.

동기 검증

오류 조건	메시지	HTTP 코드
전송 옵션 ID 누락	전송 옵션 ID가 누락되었습니다. 업데이트하려는 유효한 전송 옵션 IDs를 하나 이상 제공하거나 새 전송 옵션을 추가하려는 AddDeliveryOptions 경우를 사용합니다.	422
잘못된 가시성	에 잘못된 옵션을 제공했습니다TargetVisibility . 허용되는 옵션은 Limited, Public, 입니다Unavailable .	422
잘못된 대상 지정	에 잘못된 옵션을 제공했습니다PositiveTargeting . 에 유효	422

오류 조건	메시지	HTTP 코드
	효한 파라미터를 제공해야 합니다BuyerAccounts .	
가시성 및 대상 지정 누락	잘못된 전송 옵션 가시성 세부 정보를 제공했습니다. TargetVisibility 또는 중 하나 이상에 유효한 파라미터를 제공해야 합니다Targeting .	422
ID가 너무 AWS 계정 많음	대상 구매자 계정은 100개를 초과할 수 없습니다.	422

비동기 오류

오류 코드	오류 메시지
INVALID_DELIVERY_OPTION_IDS	잘못된 전송 옵션 세부 정보를 제공했습니다. 제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
INVALID_VISIBILITY	퍼블릭 상태에 대해 둘 이상의 전송 옵션을 제공했습니다. 퍼블릭 전송 옵션을 하나만 제공합니다.
INVALID_VISIBILITY	퍼블릭 전송 옵션을 제공하지 않았습니다. 하나의 퍼블릭 전송 옵션을 제공합니다.
감사_오류	MCO 수동 검토에 따라 다릅니다.

AWS Marketplace APIs 사용하여 기계 학습 제품 작업

AWS Marketplace 카탈로그 API를 사용하여 기계 학습(ML) 기반 제품 관리를 위한 작업을 자동화할 수 있습니다. 이 섹션에서는 카탈로그 API를 사용하는 ML 제품의 주요 작업을 다룹니다.

카탈로그 API를 사용하여 기계 학습 제품을 생성하는 방법에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [에서 제품 나열을 참조하세요 AWS Marketplace](#).

주제

- [ML 제품 목록 게시를 위한 사전 조건](#)
- [새 버전 추가](#)
- [버전 정보 업데이트](#)
- [요금 조건 업데이트](#)
- [버전 제한](#)

ML 제품 목록 게시를 위한 사전 조건

기계 학습 제품 목록 게시를 시작하기 전에 다음 사항이 있는지 확인합니다.

- 에서 판매자로 등록 AWS Marketplace. 자세한 내용은 [AWS Marketplace 판매자로 등록](#)을 참조하세요.
- AWSMarketplaceSellerFullAccess 권한이 있는 IAM 사용자.
- 공개적으로 액세스할 수 있는 Amazon Simple Storage Service(Amazon S3) 버킷으로, 제공하는 경우 회사 로고와 EULA를 호스팅합니다. ChangeSet JSON 파일에 S3 버킷의 URL을 입력합니다.
- 나열하려는 모델 패키지 또는 알고리즘 리소스의 유효한 SageMaker AI ARN입니다.
- AWS Marketplace 서비스 보안 주체와 신뢰 관계가 있고 패키지에 대한 액세스를 제공하는 유효한 IAM 역할입니다. 역할에 대한 자세한 내용은 기계 학습 제품의 [사전 조건](#) 페이지에서 AWS Marketplace 서비스 보안 주체의 IAM 역할을 참조하세요.

새 버전 추가

카탈로그 API 또는 AWS Marketplace 관리 포털을 사용하여 기계 학습 제품에 새 버전을 추가할 수 있습니다. 포털 사용에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [새 버전 추가](#)를 참조하세요.

Note

ML 제품의 경우 버전은 사용 가능한 제품을 나타내는 단일 제공 옵션으로 구성됩니다. 카탈로그 API에서 ML 제품의 제공 옵션을 사용하면 제품의 버전이 효과적으로 수정됩니다. 새 인스턴스 유형을 추가할 때 UpdatePricingTerms 변경 세트에 변경 유형을 포함하여 새 인스턴스 유형에 대한 요금을 지정합니다. 새 제품의 경우는 지원되는 모든 인스턴스 유형을 포함해야 UpdatePricingTerms 합니다. UpdatePricingDimensions차원은 지원되는 모

든 인스턴스 유형에 대해 자동으로 생성되므로는 ML 제품에 필요하거나 지원되지 않습니다. 자세한 정보는 [the section called “요금 차원 업데이트”](#) 섹션을 참조하세요.

StartChangeSet를 사용하여 버전 추가

새 버전을 추가하려면 AddDeliveryOptions 변경 유형으로 StartChangeSet 작업을 호출합니다.

1. 버전을 생성하지 않고 API 호출을 검증하려면 Intent로 설정합니다VALIDATE.
2. 실제 버전 생성의 경우를 Intent로 설정합니다APPLY.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [{
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
      "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
      "Type": "MachineLearningProduct@1.0"
    },
    "DetailsDocument": {
      "Version": {
        "VersionTitle": "version 1.1",
        "ReleaseNotes": "Patch update for small bugfix in version 1.0"
      },
      "DeliveryOptions": [{
        "Details": {
          "SageMakerAlgorithmDeliveryOptionDetails": {
            "SageMakerAlgorithmArn": "arn:aws:sagemaker:us-east-2:605142612156:algorithm/scikit-decision-trees-1552343220",
            "AccessRoleArn": "arn:aws:iam::12345678901:role/AwsMarketplaceSageMakerIngestion",
            "UsageInstructions": "This is how you use your algorithm",
            "SampleNotebookUrl": "https://www.amazon.com",
            "RepositoryUrl": "https://www.amazon.com",
            "InputProperties": {
              "Description": "Input should have all columns in the train/test file except for 'is_fraud' column."
            }
          }
        }
      ]
    }
  }
}
```

```

        "Limitations": "Can predict on 1 input in the CSV only at a
time",
        "SampleInput": {
            "RealtimeInferenceText": "{\\"prompt\\":\\"Write summary
\\",
                                \\"maxTokens\\": 1 }",
            "BatchTransformUrl": "https://www.sampleData.com",
        },
        "Parameters": [{
            "Name": "prompt",
            "Description": "Represents the instruct-style
prompt for the model. DataType is String",
            "Constraints": "Minimum length should be 1",
            "Required": true
        },
        {
            "Name": "maxTokens",
            "Description": "Denotes the number of tokens to
predict per generation. See BPE Tokens for more details.",
            "Constraints": "Minvalue: 1, MaxValue: 30"
        }
    ],
    "SageMakerCustomAttributes": [{
        "Name": "threshold",
        "Description": "Threshold of the confidence score
of the detected object",
        "Constraints": "Should be an Integer"
    }]
},
"OutputProperties": {
    "Description": "The output is a JSON object that has the
generated text along with likelihoods of tokens, if requested. See example json.",
    "SampleOutput": {
        "RealtimeInferenceUrl": "https://www.sampledata.com",
        "BatchTransformUrl": "https://www.sampleData.com",
    },
    "Parameters": [{
        "Name": "id",
        "Description": "An identifier for response"
        "AlwaysReturned": true
    },
    {
        "Name": "generations",

```

```

        "Description": "The generated text along with the
        likelihoods for tokens requested.",
      },
    ],
  },
  "RecommendedInstanceTypes": {
    "BatchTransform": "ml.m4.large",
    "RealtimeInference": "ml.m4.large",
    "Training": "ml.m4.large"
  }
}
]]
}
}],
"Intent": "APPLY"
}

```

필수 필드

- Entity (객체) - 필수

ML 제품에 대한 정보를 포함합니다.

- Identifier (문자열) - 필수

제품 ID입니다. 자세한 내용은 [식별자](#)를 참조하세요.

- Type (문자열) - 필수

전송 방법(제품 유형)을 지정합니다. 반드시 MachineLearningProduct@1.0이어야 합니다.

- DetailsDocument (객체) - 필수

제품의 새 버전에 대한 모든 세부 정보를 포함합니다.

- Version (객체) - 필수

추가되는 버전에 대한 세부 정보입니다.

- VersionTitle (문자열) - 필수

"버전 1.1" 또는 "1.1"과 같은 버전의 제목입니다. 구매자는 이러한 제목에서 버전을 선택합니다.

- ReleaseNotes (문자열) - 필수

이 버전에 대한 자세한 참고 사항입니다. 30,000자 미만이어야 합니다.

- `DeliveryOptions` (배열) - 필수

제품 버전의 제공 방법 배열입니다. 버전당 하나의 전송 옵션으로 제한됩니다.

- `Details` (객체) - 필수

- `SageMakerModelPackageDeliveryOptionDetails` 또는 `SageMakerAlgorithmDeliveryOptionDetails` (객체)

- `SageMakerModelPackageArn` 또는 `SageMakerAlgorithmArn` (문자열) - 필수

모델 패키지 또는 알고리즘의 Amazon 리소스 이름(ARN)입니다.

- `AccessRoleArn` (문자열) - 필수

AWS Marketplace가 SageMaker 리소스에 액세스하기 위한 IAM 역할 ARN입니다.

- `SampleNotebookUrl` (문자열) - 필수

사용 코드가 포함된 Jupyter 노트북 샘플 링크입니다. 자세한 내용은 GitHub의 [샘플 노트북 템플릿](#)을 참조하세요.

- `RepositoryUrl` (문자열) - 필수

노트북 및 샘플 데이터 액세스를 위한 Git 리포지토리 URL입니다. 자세한 내용은 [GitHub의 샘플 Git 리포지토리](#)를 참조하세요. GitHub

- `UsageInstructions` (문자열) - 필수

알고리즘에 대한 훈련 정보 또는 모델의 사용 세부 정보입니다.

- `InputProperties` (객체) - 필수

- `Description` (문자열) - 필수

모델/알고리즘 입력에 대한 설명

- `Limitations`(문자열)

입력 제한 사항

- `SampleInput` (객체) - 필수

`RealtimeInferenceUrl`(문자열) | `RealtimeInferenceText`(문자열) | `BatchTransformUrl`(문자열) | `BatchTransformText`(문자열)

- `Parameters` (배열<객체>)

- SageMakerCustomAttributes (배열<객체>)

모델에서 지원하는 모든 [CustomAttributes](#)를 설명합니다.

Name (문자열) - 필수 | Description (문자열) - 필수 | Constraints (문자열) | Required (부울)

- OutputProperties (객체) - 필수
- Description (문자열) - 필수
- SampleOutput (Array<Object>) - 필수

RealtimeInferenceUrl(문자열) | RealtimeInferenceText(문자열) | BatchTransformUrl(문자열) | BatchTransformText(문자열)

- Parameters (배열)

이름(문자열) - 필수 | 설명(문자열) - 필수 | AlwaysReturned(부울)

- RecommendedInstanceTypes (객체) - 필수
 - BatchTransform (문자열) - 필수
 - RealtimeInference (문자열) - 필수
 - Training (문자열) - SageMaker 알고리즘에만 필요합니다.

응답 구문

요청을 제출하면 변경 세트가 생성되고 API가 다음을 반환합니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

세트 처리 변경

변경 요청은 여러 단계를 거치는 처리 대기열로 들어갑니다.

1. 검증: 시스템은 모든 정보가 AWS Marketplace 지침을 충족하는지 확인합니다.

- 처리 시간: 몇 분에서 몇 시간
- 검증 오류는 [변경 세트 상태 및 오류](#)를 참조하세요.

2. 상태 모니터링: 요청의 상태를 두 가지 방법으로 확인할 수 있습니다.

- AWS Marketplace 관리 포털을 통해
- DescribeChangeSet 작업 사용

3. 완료: 승인되면 새 버전이 추가됩니다.

오류

비동기 오류

다음 오류는 AWS Marketplace 카탈로그 API의 AddDeliveryOptions 작업과 관련이 있습니다. 이러한 오류는 변경 세트가 처리되는 DescribeChangeSet 동안 호출할 때 나타납니다. 를 사용하여 변경 요청의 상태를 확인하는 DescribeChangeSet 방법에 대한 자세한 내용은 [변경 세트 작업을 참조](#)하세요.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT_STATUS	제공된 제품을 찾을 수 없습니다. 유효한 제품을 제공합니다.
DUPLICATE_VERSION_TITLE	제공된 버전 제목이 이미 사용 중입니다. 고유한 버전 제목을 생성합니다.
INVALID_RECOMMENDED_INSTANCE_TYPE	[x]에 유효한 인스턴스 유형을 제공하지 않았습니다. 유효한 인스턴스 유형을 입력하고 다시 시도하세요. 유효한 유형: [valids]
INCOMPATIBLE_DELIVERY_OPTIONS	제공한 전송 옵션이 이전 선택 사항인 [이전 선택 사항]과 일치하지 않습니다. 전송 옵션을 업데이트하고 다시 시도하세요.
INVALID_ASSET_ARN	SageMakerAlgorithmDeliveryOption에 유효한 ARN을 제공하지 않았습니다. 유효한 ARN을 입력하고 다시 시도하세요.
DUPLICATE_ASSET	이 제품에 고유한 ARN을 제공하지 않았습니다. 고유한 ARN을 입력하고 다시 시도하십시오.

오류 코드	오류 메시지
ASSET_NOT_FOUND	제공한 ARN을 찾을 수 없습니다. ARN이 올바른 고 필요한 권한이 있는지 확인합니다.
ASSET_VALIDATION_EXCEPTION	SagemakerModelArn/SagemakerAlgorithmArn [x]를 AWS Sagemaker 계정으로 수집할 수 없음
INVALID_ACCESS_ROLE	제공한 IAM 역할 ARN을 찾을 수 없습니다. ARN 이 올바른지 확인하고 다시 시도하세요.
UPDATE_PRICING_REQUIRED	UpdatePricingTerms가 필요합니다.

버전 정보 업데이트

카탈로그 API를 사용하여 기계 학습 제품의 기존 버전에 대한 세부 정보를 업데이트할 수 있습니다 AWS Marketplace.

Important

버전의 ARN은 업데이트할 수 없습니다. ARN을 수정해야 하는 경우 대신 새 버전을 생성해야 합니다.

StartChangeSet를 사용하여 버전 추가:

- 버전 정보를 업데이트하려면 UpdateDeliveryOptions 변경 유형으로 StartChangeSet 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
```

```

"ChangeType": "UpdateDeliveryOptions",
"Entity":
{
  "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
  "Type": "MachineLearningProduct@1.0"
},
"DetailsDocument":
{
  "Version":
  {
    "ReleaseNotes": "Adding support for new parameters"
  },
  "DeliveryOptions":
  [
    {
      "Id": "example1-2222-cccc-2222-cccccccccccc",
      "Details":
      {
        "SagemakerModelPackageDeliveryOptionDetails":
        {
          "SampleNotebookUrl": "https://www.amazon.com",
          "RepositoryUrl": "https://www.amazon.com",
          "InputProperties":
          {
            "SampleInput": {
              "RealtimeInferenceUrl": "https://www.sampleData.com",
              "BatchTransformUrl": "https://www.sampleData.com",
            },
          },
          "RecommendedInstanceTypes": {
            "BatchTransform": "ml.m4.large",
            "RealtimeInference": "ml.m4.large"
          }
        }
      }
    }
  ]
}

```

필수 필드

• Entity (객체) - 필수

ML 제품에 대한 정보를 포함합니다.

• Identifier (문자열) - 필수

제품 ID입니다. 자세한 내용은 [식별자](#)를 참조하세요.

• Type (문자열) - 필수

전송 방법(제품 유형)을 지정합니다. 반드시 MachineLearningProduct@1.0이어야 합니다.

• DetailsDocument (객체) - 필수

업데이트된 버전 정보를 포함합니다.

• Version (객체) - 필수

버전별 정보를 정의합니다.

• VersionTitle (문자열) - 필수

"버전 1.1" 또는 "1.1"과 같이 구매자에게 표시되는 버전 식별자입니다. 구매자는 이 제목을 사용하여 배포할 버전을 선택합니다.

• ReleaseNotes (문자열) - 필수

이 버전에 대한 자세한 참고 사항은 30,000자로 제한됩니다.

• DeliveryOptions (배열) - 필수

제품 버전의 전송 방법을 지정합니다. 버전당 하나의 전송 옵션으로 제한됩니다.

• Id (문자열) - 필수

DeliveryOption의 고유 식별자입니다. 제품에서 DescribeEntity 작업을 사용하여 이를 검색합니다.

• Details (객체) - 필수

전송 옵션 사양을 포함합니다.

• SageMakerModelPackageSubscriptionDetails 또는
SageMakerAlgorithmSubscriptionDetails (객체) - 필수

- SampleNotebookUrl (문자열) - 필수

구매자 사용에 대한 코드를 제공하는 샘플 Jupyter 노트북 링크입니다.

- RepositoryUrl (문자열) - 필수

노트북 및 샘플 데이터를 복제하기 위한 Git 리포지토리 URL입니다.

- UsageInstructions (문자열) - 필수

알고리즘의 경우: 훈련 정보. 모델의 경우: 추가 사용 정보.

- InputProperties (객체) - 필수

모델/알고리즘의 입력 요구 사항에 대한 세부 정보입니다.

- Description (문자열) - 필수

필수 입력에 대한 설명입니다.

- Limitations(문자열)

모든 입력 제한.

- SampleInput (객체) - 필수

다양한 작업에 대한 샘플 입력 데이터입니다.

- RealtimeInferenceUrl(문자열)

실시간 추론을 위한 샘플 입력 URL입니다.

- RealtimeInferenceText(문자열)

실시간 추론을 위한 샘플 입력 텍스트입니다.

- BatchTransformUrl(문자열)

배치 변환 작업에 대한 샘플 입력 URL입니다.

- BatchTransformText(문자열)

배치 변환 작업에 대한 샘플 입력 텍스트입니다.

- Parameters (배열<객체>)

각 입력 파라미터에 대한 세부 정보입니다.

입력 파라미터의 이름입니다.

- Description (문자열) - 필수

입력 파라미터에 대한 설명입니다.

- Constraints(문자열)

파라미터 제약 조건(MinValue, MaxValue, AllowedValues, MinLength, MaxLength, Pattern 등).

- Required(부울)

파라미터가 필요한지 여부를 나타냅니다. 기본값은 false입니다.

- SageMakerCustomAttributes (배열<객체>)

지원되는 CustomAttributes에 대한 세부 정보입니다.

- Name (문자열) - 필수

사용자 지정 속성의 이름입니다.

- Description (문자열) - 필수

사용자 지정 속성에 대한 설명입니다.

- Constraints(문자열)

속성 제약 조건(MinValue, MaxValue, AllowedValues, MinLength, MaxLength, Pattern 등).

- Required(부울)

속성이 필요한지 여부를 나타냅니다. 기본값은 false입니다.

- OutputProperties (객체) - 필수

모델/알고리즘 출력의 세부 정보입니다.

- Description (문자열) - 필수

모델/알고리즘 출력에 대한 설명입니다.

- SampleOutput (Array<Object>) - 필수

다양한 작업에 대한 샘플 출력 데이터입니다.

실시간 추론을 위한 샘플 출력 URL입니다.

- `RealtimeInferenceText`(문자열)

실시간 추론을 위한 샘플 출력 텍스트입니다.

- `BatchTransformUrl`(문자열)

배치 변환 작업에 대한 샘플 출력 URL입니다.

- `BatchTransformText`(문자열)

배치 변환 작업에 대한 샘플 출력 텍스트입니다.

- `Parameters` (배열)

각 출력 파라미터에 대한 세부 정보입니다.

- `Name` (문자열) - 필수

출력 파라미터의 이름입니다.

- `Description` (문자열) - 필수

출력 파라미터에 대한 설명입니다.

- `AlwaysReturned`(부울)

파라미터가 항상 반환되는지 여부를 나타냅니다. 기본값은 `false`입니다.

- `RecommendedInstanceTypes` (객체) - 필수

최적의 성능을 위해 권장되는 인스턴스 유형입니다.

- `BatchTransform` (문자열) - 필수

배치 변환 작업에 권장되는 인스턴스 유형입니다.

- `RealtimeInference` (문자열) - 필수

실시간 추론 작업에 권장되는 인스턴스 유형입니다.

- `Training` (문자열) - 필수

알고리즘 훈련 작업에 권장되는 인스턴스 유형입니다. SageMaker 알고리즘에만 필요합니다.

응답 구문

요청이 성공하면 다음을 반환합니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

세트 처리 변경

변경 요청은 여러 단계를 거치는 처리 대기열로 들어갑니다.

1. 검증: 시스템에서 모든 정보가 AWS Marketplace 지침을 충족하는지 확인합니다.
 - 처리 시간: 몇 분에서 몇 시간
 - 검증 오류는 [변경 세트 상태 및 오류](#)를 참조하세요.
2. 상태 모니터링: 요청의 상태를 두 가지 방법으로 확인할 수 있습니다.
 - AWS Marketplace 관리 포털을 통해
 - DescribeChangeSet 작업 사용
3. 완료: 승인되면 새 버전이 업데이트됩니다.

오류

비동기 오류

변경 세트가 처리를 시작한 후 DescribeChangeSet 작업을 사용하여 UpdateDeliveryOptions 작업에 대한 특정 오류를 검색할 수 있습니다. 오류 세부 정보 및 문제 해결은 [변경 세트 상태 및 오류](#)를 참조하세요.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT_STATUS	기존의 제한 또는 공개 제품을 사용합니다.
INVALID_DELIVERY_OPTION_ID	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]

오류 코드	오류 메시지
INCOMPATIBLE_DELIVERY_OPTION_STATUS	전송 옵션은 제한된 상태이므로 업데이트할 수 없습니다. 대신 새 버전을 추가해 보세요.
INCOMPATIBLE_DELIVERY_OPTIONS	이전에 [X]를 전송 옵션으로 사용한 제품이므로 예정된 모든 전송 옵션은 [X] 유형이어야 합니다.
INVALID_RECOMMENDED_INSTANCE_TYPE	[X]에 사용할 수 있는 기존 인스턴스 유형을 제공합니다(X는 배치 변환, 실시간 추론 또는 ALgorithm 훈련일 수 있음).
DUPLICATE_VERSION_TITLE	버전 제목은이 제품의 다른 버전 제목과 달라야 합니다.
FIELD_NOT_ALLOWED_TO_CHANGE	필드 X는 변경할 수 없습니다.

요금 조건 업데이트

카탈로그 API를 사용하여에서 기계 학습 제품의 요금을 업데이트할 수 있습니다 AWS Marketplace.

Note

기계 학습 제품의 지원되는 모든 인스턴스 유형의 가격을 책정해야 합니다. 첫 번째 버전을 생성할 때 UpdatePricingTerms 및는 제품을 게시하기 위해 지원되는 모든 인스턴스 유형에 대한 요금으로 유형을 AddDeliveryOptions 변경합니다. 이전에 지원되지 않은 새 인스턴스 유형이 있는 기존 제품에 새 버전을 추가할 때는 UpdatePricingTerms 및 AddDeliveryOptions 변경 유형에 해당 인스턴스 유형을 포함해야 합니다.

요금 조건을 업데이트하려면 UpdatePricingTerms 변경 유형으로 StartChangeSet 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
```



```

"ChangeSet": [
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Usage",
      "Terms": [
        {
          "Type": "UsageBasedPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "RateCard": [
                {
                  "DimensionKey": "ml.m4.4xlarge.m.i.b",
                  "Price": 0.1
                },
                {
                  "DimensionKey": "ml.m5.4xlarge.m.i.b",
                  "Price": 0.1
                },
                {
                  "DimensionKey": "ml.m4.16xlarge.m.i.b",
                  "Price": 0.1
                },
                {
                  "DimensionKey": "m.i.c",
                  "Price": 0.1
                },
                {
                  "DimensionKey": "ml.m4.4xlarge.a.t",
                  "Price": 0.1
                },
                {
                  "DimensionKey": "ml.m5.4xlarge.a.t",
                  "Price": 0.1
                },
                {
                  "DimensionKey": "ml.m4.16xlarge.a.t",
                  "Price": 0.1
                }
              ]
            }
          ]
        }
      ]
    }
  }
]

```

```

    ]
  }
]
},
{
  "Type": "FreeTrialPricingTerm",
  "Duration": "P30D",
  "Grants": [
    {
      "DimensionKey": "ml.m4.4xlarge.m.i.b"
    },
    {
      "DimensionKey": "ml.m5.4xlarge.m.i.b"
    },
    {
      "DimensionKey": "ml.m4.16xlarge.m.i.b"
    },
    {
      "DimensionKey": "m.i.c"
    },
    {
      "DimensionKey": "ml.m4.4xlarge.a.t"
    },
    {
      "DimensionKey": "ml.m5.4xlarge.a.t"
    },
    {
      "DimensionKey": "ml.m4.16xlarge.a.t"
    }
  ]
}
]
},
]
}

```

필수 필드

- Entity (객체) (필수) - ML 제품의 제안에 대한 정보를 포함합니다.
 - Identifier (문자열)(필수) - 요금 조건을 업데이트하려는 제안 ID입니다. 자세한 내용은 [식별자](#)를 참조하세요.

- Type (문자열)(필수) - 요금을 업데이트offer@1.0하려면가 여야 Type 합니다.
- DetailsDocument (객체)(필수) - 기계 학습 제품의 요금 조건에 대한 세부 정보입니다.
- PricingModel (문자열)(필수) - 제품의 요금 모델 - 사용량, 무료 및 계약(프라이빗 제안의 경우) 중에서 선택할 수 있습니다. 자세한 내용은 [Machine Learning 제품 요금을](#) 참조하세요.
- Terms (객체 배열) - 제품의 전체 요금을 집합적으로 정의하는 요금 조건 배열입니다.
 - Type (문자열)(필수) - 요금 조건의 유형입니다. 유효한 옵션은 UsageBasedPricingTerm 또는 FreeTrialPricingTerm 또는 FixedUpfrontPricingTerm(프라이빗 제안의 경우).
 - CurrencyCode (문자열) (UsageBasedPricingTerm 또는 FixedUpfrontPricingTerm - 요금 기간의 통화에 필요합니다. 유효한 값은 USD입니다.
 - RateCards (객체 배열)(의 경우 필수UsageBasedPricingTerm) - 기계 학습 제품의 요금을 정의하는 비율 카드입니다. 이 배열에는 하나RateCard만 포함해야 합니다.
 - RateCard (객체 배열) - 기계 학습 제품의 차원 및 비율 배열입니다. 제품을 게시하려면 지원되는 모든 인스턴스 유형의 가격이 책정되어야 합니다.
 - DimensionKey - 요금을 책정하는 차원입니다. 유효한 차원 키는 다음과 같습니다.
 - 인스턴스 유형 다음에 시간당 요금에 대한 작업이 옵니다.
 - 인스턴스 유형은 로 시작하고 그 m1 뒤에 인스턴스 이름과 크기가 옵니다. exampole의 경우 m1.m4.xlarge
 - 작업은 m.i.b 배치 변환, m.i.r 시간당 실시간 추론 및 알고리즘 훈련a.t에 사용 됩니다.
 - 예: m1.m4.4xlarge.m.i.b 배치 변환, m1.m4.xlarge.m.i.r 실시간 추론 또는 m1.m4.16xlarge.a.t 알고리즘 훈련
 - m.i.c 추론당 실시간 요금.
 - Price - 차원 속도입니다. 요금은 dimensionKey에 따라 시간당 또는 추론당입니다.
- Duration (문자열) (FreeTrialPricingTerm 또는 FixedUpfrontPricingTerm - 무료 평가판 또는 계약 기간. 무료 평가판의 경우 유효한 값은 X에서 Y 사이입니다. 계약의 경우 유효한 값은 X에서 Y 사이입니다.
- 권한 부여(객체 배열) - 무료 평가판을 사용할 수 있는 dimensionKeys에 대한 세부 정보입니다. 지원되는 모든 dimensionKeys 제공해야 합니다.
 - DimensionKey - 무료 평가판 제안의 일부로 포함할 차원입니다.
 - 인스턴스 유형 다음에 시간당 요금에 대한 작업이 옵니다.

- 인스턴스 유형은 로 시작하고 그 m1 뒤에 인스턴스 이름과 크기가 옵니다. 예:
m1.m4.xlarge
- 작업은 m.i.b 배치 변환, m.i.r 시간당 실시간 추론 또는 알고리즘 훈련을 a.t 위한 것입니다.
- 예: m1.m4.4xlarge.m.i.b 배치 변환, m1.m4.xlarge.m.i.r 실시간 추론 또는 m1.m4.16xlarge.a.t 알고리즘 훈련
- m.i.c 추론당 실시간 요금.

응답 구문

요청이 성공하면 다음을 반환합니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

세트 처리 변경

변경 요청은 여러 단계를 거치는 처리 대기열로 들어갑니다.

1. 검증: 시스템은 모든 정보가 AWS Marketplace 지침을 충족하는지 확인합니다.
 - 처리 시간: 몇 분에서 몇 시간
 - 검증 오류는 [변경 세트 상태 및 오류](#)를 참조하세요.
2. 상태 모니터링: 요청의 상태를 두 가지 방법으로 확인할 수 있습니다.
 - AWS Marketplace 관리 포털을 통해
 - DescribeChangeSet 작업 사용
3. 완료: 승인되면 요금 조건이 업데이트됩니다.

오류

비동기 오류

변경 세트가 처리를 시작한 후 DescribeChangeSet 작업을 사용하여 UpdatePricingTerms 작업에 대한 특정 오류를 검색할 수 있습니다. 오류 세부 정보 및 문제 해결은 [변경 세트 상태 및 오류](#)를 참조하세요.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT_STATUS	기존의 제한 또는 공개 제품을 사용합니다.

버전 제한

카탈로그 API를 사용하여 기계 학습 제품의 버전을 제한할 수 있습니다 AWS Marketplace. 제한되면 새 구매자는 해당 버전에 액세스할 수 없습니다. 기존 구독자는 제한된 버전에 액세스할 수 있습니다. AWS Marketplace 지침에 따라 제한 후 90일 동안 기존 구매자에 대한 지속적인 지원이 필요합니다.

Important

하나 이상의 무제한 버전을 사용할 수 있어야 합니다. 제품의 마지막 공개 버전을 제한할 수 없습니다.

버전을 제한하려면:

- 버전을 제한하려면 RestrictDeliveryOptions 변경 유형으로 StartChangeSet 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictDeliveryOptions",
      "Entity":
      {
        "Identifier": "example1-abcd-1234-5ef6-7890abcdef12@1",
        "Type": "MachineLearningProduct@1.0"
      },
      "DetailsDocument":
      {
        "DeliveryOptionIds":
```

```
[
  "example1-2222-cccc-2222-cccccccccccc"
]
}
}
]
```

필수 필드

필수 필드

- Entity (객체) - 필수

ML 제품에 대한 정보를 포함합니다.

- Identifier (문자열) - 필수

제품 ID입니다. 자세한 내용은 [식별자](#)를 참조하세요.

- Type (문자열) - 필수

전송 방법(제품 유형)을 지정합니다. 반드시 MachineLearningProduct@1.0이어야 합니다.

- DetailsDocument (객체) - 필수

업데이트된 버전 정보를 포함합니다.

- DeliveryOptionIds (객체 배열) - 필수

제한하려는 버전의 DeliveryOption IDs. 제한하려는 버전에서 DescribeEntity 작업을 호출하여 각 DeliveryOption의 고유 식별자를 검색합니다.

응답 구문

요청이 성공하면 다음을 반환합니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

1. 검증: 시스템에서 모든 정보가 AWS Marketplace 지침을 충족하는지 확인합니다.

- 처리 시간: 몇 분에서 몇 시간
 - 검증 오류는 [변경 세트 상태 및 오류를](#) 참조하세요.
2. 상태 모니터링: 요청의 상태를 두 가지 방법으로 확인할 수 있습니다.
- AWS Marketplace 관리 포털을 통해
 - DescribeChangeSet 작업 사용
3. 완료: 승인되면 새 버전이 제한됩니다.

오류

비동기 오류

변경 세트 처리 중에 다음 오류가 발생할 수 있으며 DescribeChangeSet 작업을 사용하여 검색할 수 있습니다.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT_STATUS	기존 퍼블릭 또는 제한된 제품 사용
MISSING_DELIVERY_OPTION_IDS	기존 IDs 목록에서 전송 옵션을 제공합니다.
INVALID_DELIVERY_OPTION_IDS	제품에서 찾을 수 있는 전송 옵션 IDs를 제공합니다. IDs 찾을 수 없음: [x]
INVALID_DELIVERY_OPTION_STATUS	[invalid_ids] 전송 옵션 IDs가 잘못되었습니다. 퍼블릭 상태에서 전송 옵션을 제공합니다.
ALL_DELIVERY_OPTIONS_RESTRICTED	하나 이상의 퍼블릭 상태로 유지되어야 하므로 제한할 전송 옵션을 더 적게 제공합니다.

AWS Marketplace APIs를 사용하여 제안 작업

제안은 제품 구매 및 소비에 적용되는 약관과 규칙을 정의합니다. 여기에는 두 당사자 간의 계약 조건 모음이 포함됩니다. 그러면 수락된 제안 조건이 계약으로 변환되어 거래 증명 역할을 합니다.

다음 섹션에서는 판매자 또는 채널 파트너로서 제안을 관리하는 방법에 대한 정보를 제공합니다.

주제

- [AWS MarketplaceAPIs를 사용하여 비공개 제안 작업](#)
- [AWS MarketplaceAPIs를 사용하여 재판매 권한 부여 작업](#)
- [AWS MarketplaceAPIs를 사용하여 채널 파트너 비공개 제안 작업](#)
- [갱신 작업](#)

AWS MarketplaceAPIs를 사용하여 비공개 제안 작업

를 사용하여 제안 작업을 위한 작업을 자동화AWS Marketplace Catalog API할 수 있습니다.

제품은 판매 대상을 설명하지만 AWS Marketplace제안은 제품 구매 및 소비 방식에 대한 약관을 설명합니다. AWS Marketplace제품은 여러 판매자가 여러 제안을 판매할 수 있습니다. 그러나 각 AWS Marketplace제안은 하나의 제품에 대해서만 생성할 수 있습니다. 제안에는 두 당사자 간의 계약 조건 모음이 포함됩니다. 수락된 제안 조건은 거래 증명으로 계약으로 변환됩니다.

제안에는 두 가지 유형이 있습니다.

- 비공개 제안은 판매자와 구매자가 요금을 협상하기 위한 것입니다. 판매자는 소프트웨어 구매를 위한 최종 사용자 라이선스 계약(EULA)에 서명합니다AWS Marketplace. 제안은 지정된 구매자에게만 표시됩니다. 자세한 내용은 AWS Marketplace판매자 안내서의 [비공개 제안을](#) 참조하세요.
- 공개 제안은 글로벌 구매 프로그램을 위한 것입니다. 판매자는 사용 가능한 프로그램과 지리적 위치를 기반으로 고객을 식별하므로 특정 고객만 제안에 액세스할 수 있습니다.

다음 리소스를 참조하세요.

- 작업 코드 예제는 AWS Marketplace 판매자 워크숍의 [API로 제안 관리를](#) 참조하세요.
- API 요청 코드 예제는 GitHub의 AWS 샘플에서 [Python](#) 및 [Java](#) 예제를 참조하세요.
- 비공개 제안 생성에 대한 동영상은 YouTube의 [AWS Marketplace카탈로그 API를 사용하여 비공개 제안 생성을](#) 참조하세요.
- AMI 요금 업데이트에 대한 동영상은 YouTube의 [AWS Marketplace카탈로그 API를 사용하여 AMI 제품 요금 업데이트를](#) 참조하세요.

다음 주제에서는 카탈로그 API를 사용하여 제안을 생성하고 업데이트하는 방법을 설명합니다.

주제

- [제안 생성](#)
- [대체 제안 생성](#)

- [제안 정보 업데이트](#)
- [대상 지정 구성 업데이트](#)
- [환불 정책 업데이트](#)
- [법률 리소스 업데이트](#)
- [요금 업데이트](#)
- [제안 검색 가능성 업데이트](#)
- [제안을 사용하여 생성된 계약의 만료 날짜 정의](#)
- [결제 일정 세부 정보 업데이트](#)
- [갱신 옵션 수정](#)
- [제안 게시](#)
- [기존 제안 세부 정보 설명](#)

제안 생성

카탈로그 API를 사용하여에서 새 제안을 생성할 수 있습니다AWS Marketplace. 요청이 성공적으로 처리되면는 구매자가 볼 수 Draft없는 불완전한 제안인을 AWS Marketplace Catalog API생성합니다. 제안을 완료하려면 Update 변경 유형을 사용합니다. 제안이 완료되면 [ReleaseOffer](#) 변경 유형을 사용하여 제안을 생성하고 릴리스합니다. 제안을 릴리스하면 제안을 검증하고 구매자가 볼 수 있습니다AWS Marketplace.

새 제안을 생성하려면 다음 예제와 같이 CreateOffer 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
```

```

        "ProductId": "prod-ad8EXAMPLE51",
        "Name": "Test Offer",
        "OfferSetId": "offerset-b3f9EXAMPLE27"
    }
}
]
}

```

필드에 대한 정보를 제공하여 CreateOffer 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - ProductId(문자열)(필수) - 제공되는 제품의 고유 식별자입니다.
 - 이름(문자열)(선택 사항) - 사용자와 고객의 가독성을 높이기 위해 제안과 연결된 이름입니다. 계약 정보의 일부로도 표시됩니다.
 - OfferSetId(문자열)(선택 사항) -이 제안을 연결할 제안 세트의 ID입니다. 제안 세트의 일부가 될 제안을 생성할 때만이 필드를 지정합니다. OfferSetId를 제공하지 않으면 독립 실행형으로 구매할 수 있는 개별 제안이 생성됩니다. 제안 생성 중에 OfferSetId를 지정하는 것은 제안을 해당 제안 세트와 연결하려는 의도만 나타냅니다. 연결을 완료하려면 제안이 생성된 후 [AssociateOffers 변경 유형을 사용해야](#) 합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

Status가 SUCCEEDED이면 새 OfferId가 생성됩니다.

그 응답은 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2021-05-27T22:21:26Z",
  "EndTime": "2021-05-27T22:32:19Z",
  "Status": "SUCCEEDED",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "ProductId": "prod-ad8EXAMPLE51",
        "Name": "Test Offer"
      },
      "ErrorDetailList": []
    }
  ]
}
```

동기 검증

다음 스키마 검증은의 CreateOffer 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
ProductId	필수 길이는 1~50자여야 합니다. 잘못된 문자(\, <, >)를 포함할 수 없습니다.	422
ProductId	필수	403

입력 필드	검증 규칙	HTTP 코드
	사용자는 지정된 제품에 대한 제안을 생성할 권한이 있어야 합니다.	
ProductId	필수 카탈로그의 기존 제품이거나 동일한 변경 세트에서 생성되어야 합니다.	404
이름	선택 사항 길이는 1~150자여야 합니다. 잘못된 문자(\, <, >)를 포함할 수 없습니다.	422

비동기 오류

다음 오류는 `CreateOffer` 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 `DescribeChangeSet` 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	Limited 또는 Public 상태에서 활성 제품을 사용합니다.
INCOMPATIBLE_PRODUCT	선택한 제품 유형에 대한 제안 관리는 현재에서 지원되지 않습니다. AWS Marketplace Catalog API.
INCOMPATIBLE_PRODUCT	제품에 대한 제안 관리는 현재에서 지원되지 않습니다. AWS Marketplace Catalog API.
INCOMPATIBLE_PRODUCT	제품 제안에서는 OfferSetId가 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	CreateOffer 변경 유형을 호출하여 제품에 대한 제안을 생성할 수 없습니다. CreateOff

오류 코드	오류 메시지
	erUsingResaleAuthorization 변경 유형을 사용합니다.

대체 제안 생성

카탈로그 API를 사용하여에서 대체 제안(계약 기반 제안이라고도 함)을 생성할 수 있습니다AWS Marketplace.

요청이 성공적으로 처리된 경우 AWS Marketplace Catalog API는 미완료 제안이며 구매자가 볼 수 없는 Draft 상태의 제안을 생성합니다AWS Marketplace. Update 변경 유형을 사용하여 제안을 완료합니다. 제안이 완료되면 [ReleaseOffer](#) 변경 유형을 사용하여 제안 생성 프로세스를 완료하고 제안을 릴리스하면 전체 제안을 검증하고 구매자가 제안을 볼 수 있습니다AWS Marketplace. 여기에서 구매자는 대체 제안을 수락하거나 원래 계약에 따라 계속 운영할 수 있습니다.

대체 제안을 생성하려면 다음 예제와 같이 CreateReplacementOffer 변경 유형으로 StartChangeSet API 작업을 호출하고 기존 계약 ID를 제공합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateReplacementOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "AgreementId": "agmt-12345",
        "Name": "Offer name"
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 CreateReplacementOffer 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - AgreementId(문자열)(필수) - 대체할 현재 계약의 고유 식별자입니다.
 - 이름(문자열)(선택 사항) - 사용자와 고객의 가독성을 높이기 위해 제안과 연결된 이름입니다. 계약 정보의 일부로도 표시됩니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 CreateReplacementOffer 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
AgreementId	필수 길이는 1~64자여야 합니다.	422
AgreementId	필수 사용자는 지정된 계약에 대한 제안을 생성할 권한이 있어야 합니다.	403

입력 필드	검증 규칙	HTTP 코드
이름	선택 사항 길이는 1~150자여야 합니다. 잘못된 문자(\, <, >)를 포함할 수 없습니다.	422

비동기 오류

다음 오류는 CreateReplacementOffer 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#)[변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	제품에 대한 대체 제안은 지원되지 않습니다.
INCOMPATIBLE_AGREEMENT	CreateReplacementOffer 변경 유형을 호출하여 계약에 대한 대체 제안을 생성할 수 없습니다. CreateReplacementOfferUsingResaleAuthorization 변경 유형을 사용합니다.

제안 정보 업데이트

카탈로그 API를 사용하여에서 제안 정보를 업데이트할 수 있습니다AWS Marketplace.

제안 정보를 업데이트하려면 다음 예제와 같이 UpdateInformation 변경 유형으로 StartChangeSet API 작업을 호출합니다. 다른 모든 정보는 변경되지 않습니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
```

```

"ChangeSet": [
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "offer-123456789"
    },
    "DetailsDocument": {
      "Name": "New offer name",
      "Description": "New offer description",
      "PreExistingAgreement": {
        "AcquisitionChannel": "External",
        "PricingModel": "Contract"
      }
    }
  }
]
}

```

UpdateInformation 변경 유형을 추가할 필드에 대한 정보를 제공합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - 이름(문자열)(선택 사항) - 가독성을 높이기 위해 제안과 연결된 이름입니다. 계약 정보의 일부로 표시됩니다.
 - 설명(문자열)(선택 사항) - 본인만 사용할 수 있으며 구매자가 볼 수 없는 자유 형식 텍스트입니다.
 - PreExistingAgreement 전(객체)(선택 사항) -이 제안이 동일한 기본 제품에 대해 기존 고객과 체결한 기존 계약의 갱신인지 여부를 결정합니다. 기존 계약은 내부 또는 외부에 있을 수 있습니다 AWS Marketplace.는 제안이 갱신되었는지 감사하고 확인할 AWS 수 있습니다. AWS가 제안을 확인할 수 없는 경우 AWS는 고객의 제안 및 권한을 취소할 수 있습니다.
 - AcquisitionChannel(문자열)(필수) - 기존 계약이 외부 AWS Marketplace또는 내부에서 서명되었는지 여부를 나타냅니다 AWS Marketplace.

가능한 값: External, AwsMarketplace

- PricingModel(문자열)(필수) - 기존 계약에서 사용하는 요금 모델을 나타냅니다.

가능한 값: Contract, Usage, Byol, Free

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 UpdateInformation 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
속성	다음 속성 중 하나 이상을 제공해야 합니다.	422
이름	선택 사항 길이는 1~150자여야 합니다. 잘못된 문자(\, <, >)를 포함할 수 없습니다.	422
설명	선택 사항 길이는 1~255자여야 합니다.	422
PreExistingAgreement	선택 사항	422

입력 필드	검증 규칙	HTTP 코드
	제안PreExistingAgreement 에서 제거하려면 null일 수 있습니다.	
PreExistingAgreement.PricingModel	필수 [Byo1, , Free, Contract] 값 중 하나일 수 있습니다Usage.	422
PreExistingAgreement.AcquisitionChannel	필수 [AwsMarketplace , External] 값 중 하나일 수 있습니다.	422

비동기 오류

다음 오류는의 UpdateInformation 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRE_EXISTING_AGREEMENT	제안이 릴리스된 후에는 PreExistingAgreement를 변경할 수 없습니다.

대상 지정 구성 업데이트

카탈로그 API를 사용하여에서 제안의 대상 지정 구성을 업데이트할 수 있습니다AWS Marketplace.

최신 요청에 포함되지 않은 모든 기존 대상 지정 옵션은 제안에서 제거됩니다.

Note

제안에는 선택적으로 PositiveTargeting 또는가 포함될 수 NegativeTargeting있지만 둘 다 포함될 수는 없습니다.
긍정적인 타겟팅 옵션:

- 국가 코드만 해당 - 지정된 국가의 구매자가 사용할 수 있는 공개 제안을 생성합니다.
- 구매자 계정만 해당 - 특정 AWS 계정을 대상으로 비공개 제안을 생성합니다.
- 국가 코드와 구매자 계정 모두 - 대상 계정이 지정된 국가 중 하나에 있는 경우에만 제안을 수락할 수 있는 비공개 제안을 생성합니다.

음의 대상 지정 옵션:

- 국가 코드 - 지정된 국가의 구매자를 제외합니다. 이렇게 하면 나열된 국가를 제외한 모든 국가에서 사용할 수 있는 공개 제안이 생성됩니다.

제안의 대상 지정 구성을 업데이트하려면 다음 예제와 같이 UpdateTargeting 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "CA"
          ],
          "BuyerAccounts": [
            "111122223333"
          ]
        },
        "NegativeTargeting": {
          "CountryCodes": [
```

```

        "XX"
      ]
    }
  }
}
]
}

```

필드에 대한 정보를 제공하여 UpdateTargeting 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - PositiveTargeting(객체)(선택 사항) - 긍정적인 타겟팅은 구매자의 프로필이 제안에 액세스할 수 있도록 허용하기 위해 충족해야 하는 기준을 정의합니다. 이 필드는 선택 사항이지만이 필드가 있는 경우 하나 이상의 대상 지정 옵션을 제공해야 합니다.
 - CountryCodes(문자열 배열)(선택 사항) - 국가에 따라 대상 지정을 허용하는 옵션으로 나열합니다. 국가에 대한 제안을 대상으로 하지 않으려는 경우이 필드를 생략해야 합니다. 있는 경우 목록에 하나 이상의 국가 코드가 포함되어야 합니다. 이 목록의 각 요소는 ISO 3166-1 alpha-2 형식을 사용하는 유효한 2자 국가 코드여야 합니다.
 - BuyerAccounts(문자열 배열)(선택 사항) -를 기반으로 대상 지정을 허용하는 옵션으로 나열합니다AWS 계정(프라이빗 제안이라고도 함). 에 대한 제안을 대상으로 지정하지 않으려는 경우 AWS 계정이 필드를 생략해야 합니다.
 - NegativeTargeting(객체)(선택 사항) - 부정 타겟팅은 제안에 액세스하기 위해 고객의 프로필이 충족해야 하는 기준을 정의합니다. 이 필드는 선택 사항이지만이 필드가 있는 경우 하나 이상의 대상 지정 옵션을 제공해야 합니다.
 - CountryCodes(문자열 배열)(필수) - 국가에 따라 대상 지정을 허용하는 옵션으로 나열합니다. 특정 국가를 대상으로 하지 않으려는 경우이 필드를 생략해야 합니다. 있는 경우 목록에 하나 이상의 국가 코드가 포함되어야 합니다. 이 목록의 각 요소는 ISO 3166-1 alpha-2 형식을 사용하는 유효한 2자 국가 코드여야 합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 `UpdateTargeting` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. `StartChangeSet`. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
NegativeTargeting	선택 사항 다음 중 하나를 가질 수 있습니다. [CountryCodes]	422
NegativeTargeting.CountryCodes	선택 사항 목록 크기는 1~244여야 합니다. 국가 코드는 유효해야 합니다(ISO 3166-1 alpha-2).	422
PositiveTargeting	선택 사항 다음 중 하나를 가질 수 있습니다. [CountryCodes , BuyerAccounts]	422
PositiveTargeting.BuyerAccounts	선택 사항 목록 크기는 1~260이어야 합니다.	422

입력 필드	검증 규칙	HTTP 코드
	AWS 계정IDs 유효한 형식(12자리 숫자)이어야 합니다.	
PositiveTargeting.CountryCodes	선택 사항 목록 크기는 1~244여야 합니다. 국가 코드는 유효해야 합니다(ISO 3166-1 alpha-2).	422

비동기 오류

다음 오류는 UpdateTargeting 작업과 관련이 있습니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요 변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_BUYER_ACCOUNTS	유효한 구매자 계정을 제공합니다. 잘못된 계정: [x].
INVALID_COUNTRY_CODES	지원되는 국가 코드를 제공합니다.
INVALID_TARGETING	동일한 속성에서 음수 또는 양수 타겟팅을 사용합니다.
INCOMPATIBLE_PRODUCT	제품에 대해서는 국가 기반 대상 지정이 지원되지 않습니다.
호환되지 않는_재판매_권한 부여	ResaleAuthorization 부여와 호환되는 BuyerAccounts를 제공합니다.
INCOMPATIBLE_TARGETING	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.

오류 코드	오류 메시지
INCOMPATIBLE_TARGETING	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TARGETING	대체 제안에서는 대상 지정을 업데이트할 수 없습니다. 구매자가 제공된 AgreementId와 연결되지 않은 경우 구매자와 연결된 AgreementId를 제공하여 새 비공개 제안을 생성합니다.
TOO_MANY_BUYER_ACCOUNTS	허용된 한도 내에서 BuyerAccounts를 제공합니다.
INCOMPATIBLE_TARGETING	제안이 릴리스된 후에는 BuyerAccounts를 제거할 수 없습니다.
INCOMPATIBLE_TARGETING	제안이 릴리스된 후에는 BuyerAccounts를 추가할 수 없습니다.
누락_국가_코드	CountryCodes에 PositiveTargeting 제공: [x].
INCOMPATIBLE_COUNTRY_CODES	호환되는 CountryCodes를 제공합니다.
INCOMPATIBLE_BUYER_ACCOUNTS	계약과 호환되는 BuyerAccounts를 제공합니다.

환불 정책 업데이트

카탈로그 API를 사용하여에서 제안의 환불 정책을 업데이트할 수 있습니다AWS Marketplace.

이 변경 사항은 기존 계약에 영향을 주지 않습니다. 최신 요청에 포함되지 않은 지원 약관은 제안에서 제거됩니다.

환불 정책을 업데이트하려면 다음 예제와 같이 UpdateSupportTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "SupportTerm",
            "RefundPolicy": "Updated refund policy description"
          }
        ]
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateSupportTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - 용어(구조 배열)(필수) - 업데이트하려는 지원 용어 목록입니다. 허용되는 지원 조건은 다음과 같습니다.
 - SupportTerm(객체)(필수) - 수락자가 소프트웨어를 구매할 때 사용할 수 있는 고객 지원을 정의합니다.
 - 유형(문자열)(필수) - 업데이트 중인 용어의 유형입니다. 객체 값은 입니다 "SupportTerm".
 - RefundPolicy(문자열)(필수) - 웹 사이트 및 콘솔에 있는 그대로 고객에게 표시되는 환불 정책 설명에 대한 자유 텍스트 필드입니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.


```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 `UpdateSupportTerms` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. `StartChangeSet`. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
용어	필수	422
용어[].RefundPolicy	필수 길이는 1~500이어야 합니다. 공백으로 시작하거나 끝날 수 없음	422
용어[].유형	필수 만 가능 SupportTerm	422

비동기 오류

다음 오류는 `UpdateSupportTerms` 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 `DescribeChangeSet` 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	SupportTerm은 제품에 대한 비공개 제안에서 지원되지 않습니다.
INCOMPATIBLE_TERMS	SupportTerm은 무료 평가판 제안에는 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.

법률 리소스 업데이트

카탈로그 API를 사용하여 최종 사용자 라이선스 계약(EULA)과 같은 기존 법률 문서를 교체할 수 있습니다. 최신 요청에 포함되지 않은 법률 약관은 제안에서 제거됩니다.

제안의 법적 리소스를 업데이트하려면 다음 예제와 같이 UpdateLegalTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
```

```

    "Documents": [
      {
        "Type": "CustomEula",
        "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
      }
    ]
  }
}
]
}
}
]
}
}

```

필드에 대한 정보를 제공하여 UpdateLegalTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - LegalTerm(객체)(필수) - 수락자에게 제안할 텍스트 계약 목록을 정의합니다. 이러한 계약의 한 예는 최종 사용자 라이선스 계약(EULA)입니다.
 - 유형(문자열)(필수) - 업데이트 중인 용어의 유형입니다. 객체 값은 입니다"LegalTerm".
 - 문서(구조 배열)(필수) - 구매자에게 제안할 법적 리소스에 대한 참조 목록입니다. 이러한 리소스의 한 가지 예는 최종 사용자 라이선스 계약(EULA)입니다. 각 참조는 Type 및 로 구성됩니다URL.
 - 유형(문자열)(필수) - 문서 유형입니다. 사용 가능한 문서 유형은 다음과 같습니다.
 - CustomEula - 판매자가 제공하는 사용자 지정 EULA입니다. 이 문서 유형에는 퍼블릭 S3 URL 또는 [미리 서명된 URL](#)이 필요합니다.
 - StandardEula – 표준 계약AWS Marketplace(SCMP). SCMP에 대한 자세한 내용은 AWS Marketplace판매자 안내서를 참조하세요. 이 유형은에서 관리하므로 URL을 제공하지 않습니다AWS Marketplace.
- URL(문자열)(조건부 필수) - 구매자가 읽을 수 있는 법률 문서의 URL입니다. Type가 다음 [] 중 하나일 때 필요합니다CustomEula.
- 버전(문자열)(조건부 필수) -에서 제공하는 표준 계약의 버전입니다AWS Marketplace. Type이 []인 경우 필요합니다StandardEula. 사용 가능한 버전:

- 2022-07-14 -에 대한 표준 계약 버전AWS Marketplace은 다음 Amazon S3 버킷에서 사용할 수 있습니다. <https://s3.amazonaws.com/aws-mp-standard-contracts/Standard-Contract-for-AWS-Marketplace-2022-07-14.pdf>

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdateLegalTerms 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
용어	필수 목록에LegalTerm 는 만 허용됩니다. 목록 크기는 10이어야 합니다.	422
용어[].유형	필수 만 가능 LegalTerm	422

입력 필드	검증 규칙	HTTP 코드
용어[].LegalTerm.Documents	필수	422
용어[].LegalTerm.Documents[].유형	필수 허용된 값: • CustomEula • StandardEula	422
용어[].LegalTerm.Documents[].URL	필수이며가 인 경우 유효한 URL이어야 합니다Type. CustomEula	422
용어[].LegalTerm.Documents[].버전	필수이며가 인 경우 유효한 버전이어야 합니다Type. StandardEula 유효한 StandardEula 버전: ["2019-04-24", "2022-07-14"]	422

비동기 오류

다음 오류는의 UpdateLegalTerms 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_LEGAL_DOCUMENTS	액세스 가능한 S3 버킷에 저장된 법적 문서의 URLs을 제공합니다.

오류 코드	오류 메시지
INVALID_LEGAL_DOCUMENTS	새 제안에는 최신 버전의 StandardEula만 지원됩니다.
INVALID_LEGAL_DOCUMENTS	지원되는 파일 형식으로 법률 문서를 제공합니다.
INVALID_LEGAL_DOCUMENTS	지원되는 문서 유형을 사용하여 법률 문서를 제공합니다.
LIMIT_EXCEEDED_LEGAL_DOCUMENT_SIZE	허용되는 크기 제한 내에서 법률 문서를 제공합니다.
INVALID_LEGAL_DOCUMENTS	LegalTerm에는 암호로 보호되는 문서(들)가 포함되어 있습니다. LegalTerm에서 액세스 가능한 문서를 제공합니다.
INVALID_LEGAL_DOCUMENTS	LegalTerm에 잘못된 PDF 문서(들)가 포함되어 있습니다. LegalTerm에서 액세스 가능한 문서를 제공합니다.

요금 업데이트

카탈로그 API를 사용하여 기존 요금 조건을 완전히 바꿀 수 있습니다. 최신 요청에 포함되지 않은 요금 조건은 제안에서 제거됩니다.

제안에 대한 요금 조건을 업데이트하려면 다음 예제와 같이 UpdatePricingTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

다음 요청 구문은 여러 예제를 결합합니다. 이 조합은 유효한 페이로드로 작동하지 않습니다. 예를 들어 Terms 배열에는 용어 유형FixedUpfrontPricingTerm과 용어 유형이 모두 포함될 수 없습니다ConfigurableUpfrontPricingTerm. 다양한 요금 사용 사례에 대해 다양한 기간 유형을 결합하는 방법에 대한 예는 AWS Marketplace판매자 워크숍의 [API로 제안 관리](#)를 참조하세요.

Note

무료 요금 모델을 사용하는 SaaS 제품의 경우 모든 가격이 0.00

USDConfigurableUpfrontPricingTerm로 설정된 RateCard(차원)를 하나 이상 포함UsageBasedPricingTerm하거나 포함해야 합니다. 이 요구 사항은 SaaS 제품에 고유합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "UsageBasedPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
              {
                "RateCard": [
                  {
                    "DimensionKey": "m3.large",
                    "Price": "0.10"
                  },
                  {
                    "DimensionKey": "m4.xlarge",
                    "Price": "0.20"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```
    },
    {
      "Type": "ConfigurableUpfrontPricingTerm",
      "CurrencyCode": "USD",
      "RateCards": [
        {
          "Selector": {
            "Type": "Duration",
            "Value": "P365D"
          },
          "RateCard": [
            {
              "DimensionKey": "m3.large",
              "Price": "300"
            },
            {
              "DimensionKey": "m4.xlarge",
              "Price": "400"
            }
          ],
          "Constraints": {
            "MultipleDimensionSelection": "Allowed",
            "QuantityConfiguration": "Allowed"
          }
        }
      ],
    },
    {
      "Type": "ByolPricingTerm"
    },
    {
      "Type": "RecurringPaymentTerm",
      "CurrencyCode": "USD",
      "BillingPeriod": "Monthly",
      "Price": "100.0"
    },
    {
      "Type": "FixedUpfrontPricingTerm",
      "CurrencyCode": "USD",
      "Price": "200.00",
      "Grants": [
        {
          "DimensionKey": "Users",
          "MaxQuantity": 10
        }
      ]
    }
  ]
}
```



```

    }
  ]
},
{
  "Type": "FreeTrialPricingTerm",
  "Duration": "P30D",
  "Grants": [
    {
      "DimensionKey": "m3.xlarge",
      "MaxQuantity": 10
    },
    {
      "DimensionKey": "m4.xlarge",
      "MaxQuantity": 10
    }
  ]
}
]
}
}
]
}
}
}

```

필드에 대한 정보를 제공하여 UpdatePricingTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
- PricingModel(문자열)(필수) - 제안에 대한 요금 모델입니다. 요금 모델의 가능한 값은 다음과 같습니다.
 - 사용량 - 구매자에게 제품 사용량에 대한 요금이 청구되는 사용량 기반 요금 모델입니다.
 - 계약 - 구매자에게 제품 사용에 대한 요금이 미리 청구되거나 유연한 결제 일정을 제공하는 계약 기반 요금 모델입니다. 구매자는 계약 이상의 추가 사용량에 대해서도 비용을 지불할 수 있습니다.
 - 무료 - 구매자에게 제품 사용에 대한 요금이 부과되지 않는 무료 요금 모델입니다. 이 요금 모델을 사용할 때 요금 조건 또는 결제 일정 기간은 0이 아닌 요금을 가질 수 없습니다.
 - 바이올 - 구매자가 제품 사용에 대한 자체 라이선스를 제공하는 바이올 요금 모델입니다.

- 용어(구조 배열)(필수) - 업데이트하려는 요금 조건 목록입니다. 지원되는 요금 조건은 다음과 같습니다.
- FreeTrialPricingTerm(객체) - 구매자에게 지정된 한도 내에서 요금이 부과되지 않는 단기 무료 요금 모델을 정의합니다.
- 유형(문자열) - 업데이트되는 용어의 유형입니다. 객체 값입니다 "FreeTrialPricingTerm".
- 기간(문자열) - 무료 평가판 기간의 기간입니다.
- 권한 부여(구조 배열) - 계약 실행의 일부로 무료 평가판 수락자에게 부여되는 권한입니다.
- DimensionKey(문자열) - 제품 문서에 정의된 고유한 차원 키입니다. 차원은 제품의 용량 범주를 나타내며 제품이에 나열될 때 지정됩니다AWS Marketplace.
- MaxQuantity(정수)(선택 사항) - 구매자에게 제품의 지정된 차원에 대한 권한을 부여할 수 있는 최대 용량입니다. MaxQuantity이 제공되지 않은 경우 구매자는 지정된 차원을 무제한으로 사용할 수 있습니다.
- UsageBasedPricingTerm(객체) - 제품 사용량에 따라 고객에게 요금이 부과되는 종pay-as-you-go(PAYG) 요금 모델을 정의합니다.
- 유형(문자열)(필수) - 업데이트 중인 용어의 범주입니다. 객체 값은입니다 UsageBasedPricingTerm.
- CurrencyCode(문자열) -이 기간에 언급된 가격의 통화를 정의합니다. 현재는 USD만 지원됩니다.
- RateCards(구조 배열) - 요금 카드 목록입니다.
- RateCard(구조 배열) - Rate Card는 제품 차원의 단위당 요금을 정의합니다.
- DimensionKey(문자열) - 지정된 권한이 적용되는 차원입니다. 차원은 제품의 용량 범주를 나타내며 제품이에 나열될 때 지정됩니다AWS Marketplace.
- 가격(문자열) - 구매자에게 청구되는 금액을 계산하는 데 사용할 제품 차원의 단위당 가격입니다.
- ConfigurableUpfrontPricingTerm(객체) - 구매자가 구매하려는 권한과 권한 기간을 구성할 수 있는 선불 결제 모델을 정의합니다. 각 차원에 대한 각 계약 기간 및 권한에 대한 요금 목록을 업데이트할 수 있습니다.
- 유형(문자열)(필수) - 업데이트 중인 용어의 유형입니다. 객체 값은입니다 ConfigurableUpfrontPricingTerm.
- CurrencyCode(문자열)(필수) -이 기간에 언급된 가격의 통화를 정의합니다. 공개 제안의 경우 USD만 지원됩니다. 비공개 제안의 경우 USD, AUD, EUR, GBP 및 JPY가 지원됩니다.

- 선택기(객체)(필수) - 선택기는 구매자가 선택할 수 있도록 동일한 요금 조건에서 상호 배타적인 요금 카드를 구분하는 데 사용됩니다.
- 유형(문자열)(필수) - 선택기 범주입니다. 현재 만 지원Duration됩니다.
- 값(문자열)(필수) - 계약 기간입니다. 이 필드는 ISO 8601 형식을 지원합니다.
- RateCard(구조 배열)(필수) - Rate Card는 제품 차원의 단위당 요금을 정의합니다.
- DimensionKey(문자열)(필수) - 제품 문서에 정의된 고유한 차원 키입니다. 차원은 제품의 용량 범주를 나타내며 제품이에 나열될 때 지정됩니다AWS Marketplace.
- 가격(문자열)(필수) - 구매자에게 청구되는 금액을 계산하는 데 사용되는 제품 차원의 단위당 가격입니다.
- 제약 조건(객체)(필수) - 수락자가 용어를 구성하는 방법에 대한 제약 조건을 정의합니다.

 Note

현재 MultipleDimensionSelection 및 QuantityConfiguration 값은 동일해야 합니다.

- MultipleDimensionSelection(문자열)(필수) - 구매자가 요금 카드에서 여러 차원을 선택할 수 있는지 여부를 결정합니다. 가능한 값은 Allowed 및 Disallowed입니다.
- QuantityConfiguration(문자열)(필수) - 수락자가 요금 카드의 각 차원에 대한 수량을 구성할 수 있는지 여부를 결정합니다. 가능한 값은 Allowed 및 Disallowed입니다.
- ByolPricingTerm(객체) - 사용자와 고객이 기존 계약을 로 이전할 수 있습니다AWS Marketplace. 고객은 외부의 제품에 대해 이미 요금을 지불했기 AWS Marketplace때문에에서 제품 사용에 대한 요금이 부과되지 않습니다AWS Marketplace.
- 유형(문자열)(필수) - 업데이트 중인 용어의 유형입니다. 객체 값 입니다ByolPricingTerm.
- RecurringPaymentTerm(객체) - 각 결제 기간이 끝날 때 고객에게 고정 반복 요금이 청구되는 요금 모델을 정의합니다.
- 유형(문자열)(필수) - 업데이트 중인 용어의 유형입니다. 객체 값은 입니다RecurringPaymentTerm.
- BillingPeriod(문자열)(필수) - 구매자에게 요금이 청구되는 반복을 정의합니다. 현재Monthly는 만 지원됩니다.
- 가격(문자열)(필수) - 결제 기간마다 구매자에게 청구되는 금액입니다.
- CurrencyCode(문자열)(필수) -이 기간에 언급된 가격의 통화를 정의합니다. 현재 USD만 지원됩니다.

- **FixedUpfrontPricingTerm(객체)** - 고객에게 고정 선결제 금액이 청구되는 선결제 요금 모델을 정의합니다.
- **유형(문자열)(필수)** - 업데이트 중인 용어의 유형입니다. 객체 값은 `FixedUpfrontPricingTerm`.
- **CurrencyCode(문자열)(필수)** - 이 기간에 언급된 가격의 통화를 정의합니다. 공개 제안의 경우 USD만 지원됩니다. 비공개 제안의 경우 USD, AUD, EUR, GBP 및 JPY가 지원됩니다.
- **가격(문자열)(필수)** - 이 기간이 수락되면 고객에게 청구되는 고정 금액입니다.
- **권한 부여(구조 배열)(필수)** - 계약 실행의 일부로 고정 선결제 수락자에게 부여되는 권한입니다.
- **DimensionKey(문자열)(필수)** - 제품 문서에 정의된 고유한 차원 키입니다. 차원은 제품의 용량 범주를 나타내며 제품이에 나열될 때 지정됩니다AWS Marketplace.
- **MaxQuantity(정수)(필수)** - 구매자에게 제품의 지정된 차원에 대한 권한을 부여할 수 있는 최대 용량입니다. `MaxQuantity`이 제공되지 않은 경우 구매자는 지정된 차원을 무제한으로 사용할 수 있습니다.
- **기간(문자열)(선택 사항)** - 용어가 활성 상태로 유지되는 기간을 정의합니다. 이 필드는 ISO 8601 형식을 지원합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 `ChangeSetArn` 대한 `ChangeSetId` 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdatePricingTerms 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP
PricingModel	필수 허용된 요금 모델: ["Byol", "Free", "Usage", "Contract"]	422
용어	필수 허용된 약관: ["ConfigurableUpfrontPricingTerm", "ByolPricingTerm", "FreeTrialPricingTerm", "UsageBasedPricingTerm", "RecurringPaymentTerm", "FixedUpfrontPricingTerm"]	422
용어[].ByolPricingTerm	필수	422
용어[].ByolPricingTerm.Type	필수 "ByolPricingTerm"만 가능합니다.	422
용어[].ConfigurableUpfrontPricingTerm	필수	422
용어[].ConfigurableUpfrontPricingTerm.Type	필수 "ConfigurableUpfrontPricingTerm"만 가능합니다.	422
용어[].ConfigurableUpfrontPricingTerm.CurrencyCode	필수 지원되는 통화: ["USD", "AUD", "EUR", "GBP", "JPN"]	422
용어[].ConfigurableUpfrontPricingTerm.RateCards	필수	422

입력 필드	검증 규칙	HTTP
	목록 크기는 1~5여야 합니다.	
용어[].ConfigurableUpfrontPricingTerm.RateCards[].계약 조건	필수	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints.MultipleDimensionSelection	필수 허용된 값: ["Allowed", "Disallowed"]	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints.QuantityConfiguration	필수 허용된 값: ["Allowed", "Disallowed"]	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard	필수 목록 크기는 1~800이어야 합니다.	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard[].DimensionKey	필수 길이는 1~100이어야 합니다.	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard[].가격	필수 데이터 형식은 "문자열"입니다. 최대 3개의 소수 자릿수가 지원되는 음수가 아닌 소수 자릿수	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].선택기	필수	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].Selector.Type	필수 허용되는 값: ["기간"]	422
용어[].ConfigurableUpfrontPricingTerm.RateCards[].Selector.Value	필수 선택기 유형당 예상 형식: ISO 8601 기간	422

입력 필드	검증 규칙	HTTP
용어[].FixedUpfrontPricingTerm	필수	422
용어[].FixedUpfrontPricingTerm.Type	필수 "FixedUpfrontPricingTerm"만 가능합니다.	422
용어[].FixedUpfrontPricingTerm.CurrencyCode	필수 지원되는 통화: ["USD", "AUD", "EUR", "GBP", "JPN"]	422
용어[].FixedUpfrontPricingTerm.Duration	필수 선택기 유형당 예상 형식: ISO 8601 기간	422
용어[].FixedUpfrontPricingTerm.Grants	필수 목록 크기는 1~200이어야 합니다.	422
용어[].FixedUpfrontPricingTerm.Grants[].DimensionKey	필수 길이는 1~100이어야 합니다.	422
용어[].FixedUpfrontPricingTerm.Grants[].MaxQuantity	RequiredValue는 0보다 커야 합니다.	422
용어[].FixedUpfrontPricingTerm.Price	필수 데이터 형식은 "문자열"입니다. 최대 3개의 소수 자릿수가 지원되는 음수가 아닌 소수 자릿수	422
용어[].FreeTrialPricingTerm	필수	422

입력 필드	검증 규칙	HTTP
용어[].FreeTrialPricingTerm.Type	필수 "FreeTrialPricingTerm"만 가능합니다.	422
용어[].FreeTrialPricingTerm.Duration	필수 예상 형식: ISO 8601 기간	422
용어[].FreeTrialPricingTerm.Grants	필수 목록 크기는 1~800이어야 합니다.	422
용어[].FreeTrialPricingTerm.Grants[].DimensionKey	필수 길이는 1~100이어야 합니다.	422
용어[].FreeTrialPricingTerm.Grants[].MaxQuantity	선택 사항 값은 0보다 커야 합니다.	422
용어[].RecurringPaymentTerm	필수	422
용어[].RecurringPaymentTerm.Type	필수 "RecurringPaymentTerm"만 가능합니다.	422
용어[].RecurringPaymentTerm.BillingPeriod	필수 허용되는 값: ["월별"]	422
용어[].RecurringPaymentTerm.CurrencyCode	필수 지원되는 통화: ["USD"]	422

입력 필드	검증 규칙	HTTP
용어[].RecurringPaymentTerm.Price	필수 데이터 형식은 "문자열"입니다. 최대 3개의 소수 자릿수가 지원되는 음수가 아닌 소수 자릿수	422
용어[].UsageBasedPricingTerm	필수	422
용어[].UsageBasedPricingTerm.Type	필수 "UsagedBasedPricingTerm"만 가능합 니다.	422
용어[].UsageBasedPricingTerm. CurrencyCode	필수 지원되는 통화: ["USD"]	422
용어[].UsageBasedPricingTerm. RateCards	필수 크기는 10이어야 합니다.	422
용어[].UsageBasedPricingTerm. RateCards[].RateCard	필수 목록 크기는 1~8000이어야 합니다.	422
용어[].UsageBasedPricingTerm. RateCards[].RateCard[].Dime nsionKey	필수 길이는 1~1000이어야 합니다.	422
용어[].UsageBasedPricingTerm. RateCards[].RateCard[].가격	필수 데이터 형식은 "문자열"입니다. 최대 8개의 소수 자릿수가 지원되는 음수가 아닌 소수 자릿수	422

비동기 오류

다음 오류는 UpdatePricingTerms 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
DUPLICATE_DIMENSION_KEYS	Grants에 [x]의 고유한 차원 키 목록을 제공합니다.
DUPLICATE_DIMENSION_KEYS	RateCard에 [x]의 고유한 차원 키 목록을 제공합니다.
DUPLICATE_SELECTORS	ConfigurableUpfrontPricingTerm에 고유한 선택기 목록을 제공합니다.
DUPLICATE_TERM_TYPES	용어 유형의 고유한 목록을 제공합니다.
INCOMPATIBLE_AGREEMENT	대체 제안에서 [x, y, z]라는 용어는 제거할 수 없습니다.
INCOMPATIBLE_AGREEMENT	대체 제안에 [x, y, z]라는 용어를 추가할 수 없습니다.
호환되지 않는_통화_코드	제안이 릴리스된 후에는 CurrencyCode를 변경할 수 없습니다.
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_PRODUCT	제품에는 사용 요금 모델이 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에 대해서는 계약 요금 모델이 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에 대해서는 바이올 요금 모델이 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에는 무료 요금 모델이 지원되지 않습니다.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	[x]는 제품에 대한 제안에서 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제공된 결제 및 요금 조건이 호환되지 않습니다.
INCOMPATIBLE_PRODUCT	[x]의 제품에서 사용 가능한 기존 차원을 사용합니다.
INCOMPATIBLE_PRODUCT	제안의 유일한 요금 기간인 FreeTrialPricingTerm은 제품에 대해 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에는 [x,y,z]라는 용어가 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	대체 제안은 계약 요금 모델에 대해서만 지원됩니다.
INCOMPATIBLE_PRODUCT	제품 차원과 호환되는 요금 조건(들)을 제공합니다. 호환되지 않는 요금 조건: [x,y,z].
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	사용량 요금 모델에 대해 ConfigurableUpfrontPricingTerm에서 MultipleDimensionSelection 및 QuantityConfiguration을 허용으로 설정합니다. QuantityConfiguration ConfigurableUpfrontPricingTerm
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	사용량 요금 모델에 대해 ConfigurableUpfrontPricingTerm에서 MultipleDimensionSelection 및 QuantityConfiguration을 허용되지 않음으로 설정합니다. QuantityConfiguration ConfigurableUpfrontPricingTerm
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	제안이 릴리스된 후에는 ConfigurableUpfrontPricingTerm QuantityConfiguration을 변경할 수 없습니다.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	제안이 릴리스된 후에는 ConfigurableUpfrontPricingTerm의 MultipleDimensionSelection을 변경할 수 없습니다.

오류 코드	오류 메시지
INCOMPATIBLE_RATES	무료 요금 모델을 사용할 때 모든 요금 금액과 가격을 0(0)으로 설정합니다.
INCOMPATIBLE_RATES	제품에 대한 무료 평가판 제안의 경우 UsageBasedPricingTerm에는 영(0) 가격만 허용됩니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization에 지정된 것과 동일한 CurrencyCode를 제공합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	FixedUpfrontPricingTerm의 기간이 ResaleAuthorization에 지정된 기간과 일치하는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization과 호환되는 용어(들)를 제공합니다. 호환되지 않는 용어: [x, y, z].
INCOMPATIBLE_SELECTOR_DURATION	제안이 발표된 후에는 ConfigurableUpfrontPricingTerm의 요율 카드에서 기간을 제거할 수 없습니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.
INCOMPATIBLE_TERMS	[x]는 [y,z]라는 용어와 함께 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 [x,y,z]라는 용어를 추가할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 다음 조건을 제거할 수 없습니다. [x,y,z].
INCOMPATIBLE_TERMS	비공개 제안에는 [x]가 지원되지 않습니다.
INCOMPATIBLE_TERMS	무제한 사용을 허용하는 FreeTrialPricingTerm에서는 다음 용어가 지원되지 않습니다. [x,y,z].

오류 코드	오류 메시지
INCOMPATIBLE_TERMS	제품의 FreeTrialPricingTerm에서는 [x,y,z]라는 용어가 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안에 PaymentScheduleTerm이 포함된 경우 FixedUpfrontPricingTerm에 대해 영(0) 가격을 제공합니다.
INCOMPATIBLE_TERMS	다음 조건은 PricingModel [x,y,z]과 호환되지 않습니다.
INCOMPATIBLE_TERMS	FixedUpfrontPricingTerm은 MarkupPercentage가 0(0)보다 큰 경우 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_AGREEMENT_DURATION	[x]~[y]개월의 기간을 입력합니다.
INVALID_AGREEMENT_DURATION	측정된 차원에 대해 지속 시간 세부 수준이 일 수준인지 확인합니다.
INVALID_CURRENCY_CODE	지원되는 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	지원되는 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
INVALID_DURATION	FreeTrialPricingTerm의 기간이 허용된 범위 내에 있는지 확인합니다.

오류 코드	오류 메시지
INVALID_DURATION	AgreementStartDate AgreementEndDate 사이의 기간과 일치하는 FixedUpfrontPricingTerm의 기간을 제공합니다.
INVALID_DURATION	[x]~[y]개월의 기간을 입력합니다.
INVALID_DURATION	측정된 차원에 대한 지속 시간 세부 수준이 일 수준인지 확인합니다.
INVALID_GRANTS	FreeTrialPricingTerm의 모든 권한 부여에 대해 동일한 MaxQuantity를 제공합니다.
INVALID_GRANTS	FreeTrialPricingTerm에서 사용 가능한 모든 측정 차원에 대한 권한 부여를 제공합니다.
INVALID_GRANTS	The combination of Dimensions in grants is invalid in FixedUpfrontPricingTerm for the product.
INVALID_GRANTS	The combination of Dimensions in grants is invalid in FreeTrialPricingTerm for the product.
INVALID_GRANTS	FixedUpfrontPricingTerm with MaxQuantity is not supported for this product.
INVALID_PRICE_CHANGE	[x]는 지난 120일 동안 가격 인상을 요청했기 때문에 [y]까지 업데이트할 수 없습니다. 이전 가격 인상 요청을 취소하거나 자세한 내용은 AWS Marketplace관리형 카탈로그 운영 팀에 문의하십시오.
INVALID_PRICE_CHANGE	[x]의 가격 인상 및 차원 추가는 동일한 요청에서 지원되지 않습니다. 먼저 차원을 추가합니다.

오류 코드	오류 메시지
INVALID_PRICE_CHANGE	UsageBasedPricingTerm의 가격 상승 및 하락은 동일한 요청에서 지원되지 않습니다. 먼저 가격을 낮춥니다.
INVALID_PRICE_CHANGE	동일한 요청에서는 RecurringPaymentTerm의 가격 인상과 UsageBasedPricingTerm의 가격 하락이 지원되지 않습니다. 먼저 가격을 낮춥니다.
INVALID_PRICE_CHANGE	동일한 요청에서는 RecurringPaymentTerm의 가격 감소와 UsageBasedPricingTerm의 가격 증가가 지원되지 않습니다. 먼저 가격을 낮춥니다.
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm에 기간 [x] 동안 하나 이상의 차원 키가 누락되었습니다. 모든 기간 동안 동일한 차원 키 세트에 대한 가격을 제공합니다.
INVALID_RATE_CARD	UsageBasedPricingTerm에서 측정된 차원에 대해서만 요금 카드를 제공합니다.
INVALID_RATE_CARD	[x]에서는 속도를 제거할 수 없습니다. 기존 요금 카드의 모든 차원에 대한 가격을 제공합니다.
INVALID_RATE_CARD	[x]에서 단위가 동일한 차원을 제공합니다.
INVALID_RATE_CARD	모든 측정 차원 또는 모든 권한이 있는 차원을 [x]로 입력합니다.
INVALID_RATE_CARD	[x]에는 권한이 있는 차원만 제공합니다.
INVALID_RATE_CARD	UsageBasedPricingTerm에서 사용 가능한 모든 측정 차원에 대한 사용량 기반 요금을 제공합니다.

오류 코드	오류 메시지
INVALID_RATE_CARD	모든 무료 평가판 차원에 대한 사용량 기반 요금을 제공합니다.
INVALID_RATE_CARD	UsageBasedPricingTerm에 최대 8개의 소수 자릿수가 있는 가격을 제공합니다.
INVALID_RATE_CARD	The combination of Dimensions in rate card is invalid in UsageBasedPricingTerm for the product.
INVALID_SELECTOR_DURATION_VALUE	[x]~[y]개월의 기간을 입력합니다.
INVALID_SELECTOR_DURATION_VALUE	측정된 차원에 대해 지속 시간 세부 수준이 일 수준인지 확인합니다.
INVALID_SELECTOR_DURATION_VALUE	ConfigurableUpfrontPricingTerm의 기간이 허용된 범위 내에 있는지 확인합니다.
INVALID_SELECTOR_DURATION_VALUE	지원되는 계약 기간을 하나 이상 제공합니다.
INVALID_SELECTOR_DURATION_VALUE	지원되는 계약 기간 또는 단일 사용자 지정 기간을 하나 이상 제공합니다.
INVALID_SELECTOR_DURATION_VALUE	AgreementStartDate AgreementEndDate 간의 기간과 일치하는 ConfigurableUpfrontPricingTerm의 기간을 제공합니다.
누락_지속 시간	FixedUpfrontPricingTerm에 기간을 입력합니다.
MISSING_MANDATORY_TERMS	FixedUpfrontPricingTerm은 ByolPricingTerm 또는 PaymentScheduleTerm과 페어링된 경우에만 지원됩니다.
MISSING_MANDATORY_TERMS	[x,y,z] 중 하나 이상을 제공합니다.
MISSING_MANDATORY_TERMS	Byol 요금 모델을 사용할 때 ByolPricingTerm을 제공합니다.

오류 코드	오류 메시지
TOO_MANY_GRANTS	[y]에 최대 [x]개의 권한 부여를 제공합니다.
TOO_MANY_RATE_CARDS	ConfigurableUpfrontPricingTerm의 요금 카드 하나만 제품에 허용됩니다.
TOO_MANY_RATE_CARDS	제품의 ConfigurableUpfrontPricingTerm에는 최대 [x]개의 요금 카드가 허용됩니다.
TOO_MANY_RATES	ConfigurableUpfrontPricingTerm의 허용된 한도 내에서 RateCards를 제공합니다.
TOO_MANY_RATES	UsageBasedPricingTerm의 허용된 한도 내에서 RateCards를 제공합니다.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	ConfigurableUpfrontPricingTerm에서 MultipleDimensionSelection과 QuantityConfiguration을 모두 동일한 값(허용됨 또는 허용되지 않음)으로 설정합니다. QuantityConfiguration
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	ConfigurableUpfrontPricingTerm에서 모든 요금 카드에 대해 동일한 제약 조건을 제공합니다.
INVALID_UPDATE_REQUEST	[x]는 업데이트할 수 없습니다. 요금 변경을 요청하거나 자세한 내용은 AWS Marketplace관리형 카탈로그 운영 팀에 문의하세요.
INCOMPATIBLE_PRICING_MODEL	PricingModel은 [x]에서 [y]로 변경할 수 없습니다.
INVALID_GRANTS	FreeTrialPricingTerm의 MaxQuantity는 제품에 대해 제한됩니다. [x] 이하의 MaxQuantity를 제공합니다. 자세한 내용은 AWS Marketplace관리형 카탈로그 운영 팀에 문의하십시오.
INVALID_GRANTS	FixedUpfrontPricingTerm의 모든 권한 부여에 MaxQuantity를 제공합니다.

오류 코드	오류 메시지
INVALID_GRANTS	MaxQuantity는 제품에 대한 FixedUpfrontPricingTerm에서 지원되지 않습니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	FixedUpfrontPricingTerm의 권한 부여가 ResaleAuthorization에 지정된 RateCards 일치하는지 확인합니다.

제안 검색 가능성 업데이트

카탈로그 API를 사용하여에서 제안의 검색 가능성을 제어할 수 있습니다AWS Marketplace.

향후 특정 날짜를 설정하여 제안 검색 가능성을 제한하거나 과거에 제안을 만료하도록 선택할 수 있습니다. UpdateAvailability 변경 유형은 기존 계약에 영향을 주지 않습니다.

Note

- 이미 [게시](#)된(릴리스된) 비공개 제안에서 UpdateAvailability 변경 유형을 사용할 수 있습니다. 구매자가 이미 비공개 제안을 수락한 경우 기존 계약은 영향을 받지 않습니다.
- 기존 비공개 제안AvailabilityEndDate의를 수정할 때는 [계약 기간의 제약 조건](#)을 준수해야 합니다. 그렇지 않은 경우이 UpdateValidityTerms 변경 세트에 추가 변경 유형을 포함하여 새 만료를 준수하도록 계약 기간을 수정합니다. UpdateValidityTerms 변경 유형은 릴리스되거나 아직 릴리스되지 않은 비공개 제안에 사용할 수 있습니다.
- 기존 비공개 제안AvailabilityEndDate의를 수정할 때는 [결제 일정의 제약 조건](#)을 준수해야 합니다. 그렇지 않고 비공개 제안이 아직 릴리스되지 않은 경우이 UpdatePaymentScheduleTerms 변경 세트에 추가 변경 유형을 포함하여 새 만료를 준수하도록 결제 일정을 수정합니다. 비공개 제안이 이미 릴리스된 경우 새 날짜가 결제 일정의 제약 조건을 준수하는 경우에만 AvailabilityEndDate를 변경할 수 있습니다.

제안의 검색 가능성을 제어하려면 다음 예제와 같이 UpdateAvailability 변경 유형을 사용하여 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
```

```
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2024-05-31"
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateAvailability 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - AvailabilityEndDate(문자열)(필수) - 제안을 검색하고 구매할 수 있는 날짜입니다 AWS Marketplace. 향후 특정 날짜를 설정하여 가용성을 제한하거나 과거에 제안을 만료하도록 선택할 수 있습니다. 날짜는 YYYY-MM-DD 형식으로 표시됩니다.

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트의 ID와 ARN을 제공하며 다음과 같습니다.

응답 구문

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 검증 정보가 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다. 를 통해 AWS Marketplace Management Portal 또는 DescribeChangeSet 작업을 사용하여 카탈로그 API에서 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 UpdateAvailability 작업에 따라 다릅니다 AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다 StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
AvailabilityEndDate	필수 형식: "YYYY-MM-DD"	422

비동기 오류

다음 오류는 UpdateAvailability 작업에만 해당됩니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_AVAILABILITY_END_DATE	AvailabilityEndDate는 공개 제안에 대해 지원되지 않습니다.
INVALID_AVAILABILITY_END_DATE	향후 AvailabilityEndDate를 제공합니다.
INVALID_AVAILABILITY_END_DATE	AgreementEndDate 이전의 AvailabilityEndDate를 제공합니다. AgreementEndDate
MISSING_AVAILABILITY_END_DATE	계약 종료 날짜 이전의 AvailabilityEndDate를 제공합니다.

제안을 사용하여 생성된 계약의 만료 날짜 정의

카탈로그 API를 사용하여 제안을 사용하여 생성된 계약의 만료 날짜 세부 정보를 정의할 수 있습니다 AWS Marketplace.

이 변경 유형은 기존 계약에 영향을 주지 않습니다.

Note

이미 [게시](#)된(릴리스된) 비공개 제안에서 UpdateValidityTerms 변경 유형을 사용할 수 있습니다. 구매자가 이미 비공개 제안을 수락한 경우 해당 기존 계약은 영향을 받지 않습니다. AMI 기반 및 컨테이너 기반 제품의 경우 비공개 제안 [요금 조건에](#)가 있는 기간 유형Duration(예: 기간 유형 FixedUpfrontPricingTerm 또는 ConfigurableUpfrontPricingTerm)이 포함된 경우이 변경 유형의 AgreementDuration 집합은 오늘부터 [비공개 제안 만료](#)까지의 일수에 Duration 해당 기간 유형에 설정된 일수를 더한 값보다 커야 합니다. 이는 구매자가 비공개 제안을 수락하고 계약이 생성된 후 비공개 제안이 만료될 때까지 선택적으로 해당 기간 유형에 지정된 추가 권한을 구매할 수 있기 때문입니다. 또한 모든 추가 권한은 계약 전에 종료되어야 합니다. 예를 들어 구매자가 사용 가능한 첫날에 비공개 제안을 수락한 다음 사용 가능한 마지막 날에 권한을 구매하는 경우 계약 종료 날짜 이후에 해당 권한이 종료되지 않아야 합니다.

제안을 사용하여 생성된 계약의 만료 날짜 세부 정보를 정의하려면 다음 예제와 같이 UpdateValidityTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateValidityTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
```

```

    {
      "Type": "ValidityTerm",
      "AgreementDuration": "P12M",
      "AgreementStartDate": "2021-08-01",
      "AgreementEndDate": "2022-08-01"
    }
  ]
}
]
}

```

필드에 대한 정보를 제공하여 UpdateValidityTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - 용어(구조 배열) - 업데이트하려는 유효 용어 목록입니다. 지원되는 유효 기간은 다음과 같습니다.
 - ValidityTerm(객체) -이 제안에서 생성된 계약을 유효하게 유지할 조건을 정의합니다.
 - 유형(문자열) - 업데이트 중인 용어의 범주입니다. ValidityTerm
 - AgreementDuration(문자열) - 계약이 활성 상태로 유지되는 기간을 정의합니다. AgreementStartDate이 제공되지 않은 경우 계약 기간은 계약 서명 시간을 기준으로 합니다. 기간은 ISO_8601 형식으로 표시됩니다.
 - AgreementStartDate(문자열) - 계약이 시작되는 날짜를 정의합니다. AgreementStartDate는 YYYY-MM-DD 형식으로 표시됩니다. 계약은 제공된 날짜의 00:00:00.000 UTC에 시작됩니다. AgreementStartDate이 제공되지 않은 경우 계약 시작 날짜는 계약 서명 시간을 기준으로 결정됩니다.
 - AgreementEndDate(문자열) - 계약이 종료되는 날짜를 정의합니다. AgreementEndDate는 YYYY-MM-DD 형식으로 표시됩니다. 계약은 제공된 날짜의 23:59:59.999 UTC에 종료됩니다. AgreementEndDate이 제공되지 않은 경우 계약 종료일은 개별 조건의 유효성에 따라 결정됩니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 `UpdateValidityTerms` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. `StartChangeSet`. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
용어	필수	422
용어[].유형	필수 만 가능 "ValidityTerm"	422
용어[].AgreementDuration	선택 사항 선택기 유형당 예상 형식: ISO 8601 기간 단독으로 사용하거나와 페어링할 수 있습니다. AgreementStartDate	422
용어[].AgreementEndDate	선택 사항 날짜는 다음과 같은 형식이어야 합니 다. "YYYY-MM-DD"	422
용어[].AgreementStartDate	선택 사항	422

입력 필드	검증 규칙	HTTP 코드
	날짜는 다음과 같은 형식이어야 합니다. "YYYY-MM-DD"	
	AgreementEndDate 및 와만 페어링할 수 있습니다. Agreement Duration	

비동기 오류

다음 오류는 UpdateValidityTerms 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_AGREEMENT	대체할 현재 계약이 미래 날짜가 아닌 경우 AgreementStartDate는 미래 날짜가 될 수 없습니다.
INCOMPATIBLE_AGREEMENT_END_DATE	제안이 릴리스된 후에는 AgreementEndDate를 업데이트할 수 없습니다.
INCOMPATIBLE_AGREEMENT_START_DATE	제안이 릴리스된 후에는 AgreementStartDate를 업데이트할 수 없습니다.
INCOMPATIBLE_PRODUCT	향후 AgreementStartDate는 지원되지 않습니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementStartDate와 AgreementEndDate 사이의 기간이 ResaleAuthorization 부여와 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementStartDate가 ResaleAuthorization 부여와 호환되는지 확인합니다.

오류 코드	오류 메시지
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementEndDate가 ResaleAuthorization 부여와 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementStartDate와 AgreementEndDate 사이의 기간이 ResaleAuthorization 부여와 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementDuration이 ResaleAuthorization에 지정된 기간과 일치하는지 확인합니다.
INCOMPATIBLE_TERMS	공개 제안에는 ValidityTerm이 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_AGREEMENT_DURATION	[x]일 이상의 AgreementDuration 제공합니다.
INVALID_AGREEMENT_END_DATE	향후 AgreementEndDate를 제공합니다.
INVALID_AGREEMENT_END_DATE	[x] 이후이거나 같은 AgreementEndDate를 제공합니다.
INVALID_AGREEMENT_START_DATE	AvailabilityEndDate 이후인 Agreement StartDate를 제공합니다. AvailabilityEndDate
INVALID_AGREEMENT_START_DATE	AgreementStartDate 이전의 Agreement EndDate를 제공합니다.
INVALID_AGREEMENT_START_DATE	오늘부터 [x]년 이내에 있는 Agreement StartDate를 제공합니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementDuration과 AgreementEndDate가 모두 포함된 ValidityTerm은 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementStartDate와 AgreementDuration이 모두 포함된 ValidityTerm은 제품에 대한 제안에서 지원되지 않습니다.

오류 코드	오류 메시지
INVALID_AGREEMENT_TIME_INTERVAL	AgreementStartDate가 포함된 ValidityTerm은 제품에 대한 제안에서 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementStartDate만 있는 ValidityTerm은 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementEndDate는 향후 AgreementStartDate와 함께 사용하거나 대체 제안에 사용하지 않는 한 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	차이가 [x]년 이하인 경우 AgreementStartDate 및 AgreementEndDate를 제공합니다.
MISSING_AGREEMENT_START_DATE	ConfigurableUpfrontPricingTerm과 함께 사용할 때 AgreementStartDate가 ValidityTerm에 있는지 확인합니다. ValidityTerm
INVALID_AGREEMENT_END_DATE	오늘부터 [x]년 이내에 있는 AgreementEndDate를 제공합니다.
INCOMPATIBLE_AGREEMENT_START_DATE	계약의 미래 시작일이 있을 때 계약에 정의된 것과 동일한 AgreementStartDate를 제공합니다.
INCOMPATIBLE_AGREEMENT	AgreementStartDate는 계약이 미래 날짜가 아닌 미래 날짜가 될 수 없습니다.

결제 일정 세부 정보 업데이트

카탈로그 API를 사용하여 유연한 결제 일정과 같은 제안의 결제 일정 세부 정보를 업데이트할 수 있습니다AWS Marketplace.

Note

이미 [게시](#)된(릴리스된) 제안에는 UpdatePaymentScheduleTerms 변경 유형을 사용할 수 없습니다.

비공개 제안은 비공개 제안 생성부터 [만료](#)(에 설정)까지 언제든지 수락할 수 있습니다AvailabilityEndDate. 결제 일정의 한 가지 ChargeDate 값만 구매자가 비공개 제안을

수락할 수 있는 마지막 날짜(프라이빗 제안 만료 날짜) 또는 그 이전 날짜일 수 있습니다. 의 나머지 값은 비공개 제안이 만료된 이후여야 하지만 비공개 제안이 즉시 수락된 경우 계약 종료까지 ChargeDate여야 합니다. 계약 종료는 비공개 제안이 수락된 시점(계약 생성)과 [계약 기간을 기준으로 합니다](#).

제안에 대한 결제 일정 세부 정보를 업데이트하려면 다음 예제와 같이 UpdatePaymentScheduleTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePaymentScheduleTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "PaymentScheduleTerm",
            "Schedule": [
              {
                "ChargeDate": "2021-12-01",
                "ChargeAmount": "200.00"
              },
              {
                "ChargeDate": "2022-03-01",
                "ChargeAmount": "250.00"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

}

필드에 대한 정보를 제공하여 UpdatePaymentScheduleTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - 용어(구조 배열) - 업데이트하려는 결제 조건 목록입니다. 지원되는 결제 조건은 다음과 같습니다.
 - PaymentScheduleTerm(객체) - 계약 유효 기간 동안 고객에게 다른 날짜에 고정 요금이 청구되는 할부 기반 요금 모델을 정의합니다.
 - 유형(문자열) - 업데이트되는 용어의 유형입니다. 객체 값입니다 "PaymentScheduleTerm".
 - 일정(구조 배열) - 각 요소가 하나의 결제 분할을 정의하는 결제 일정 목록입니다. 여기에는 지불할 가격과 고객에게 요금이 청구되는 날짜를 계산하는 데 필요한 정보가 포함되어 있습니다.
 - ChargeDate(문자열) - 고객이이 결제 일정 기간에 정의된 요금을 지불하는 날짜입니다. ChargeDate는 YYYY-MM-DD 형식으로 표시됩니다. 인보이스는 제공된 날짜에 생성됩니다.
 - ChargeAmount(문자열) - 고객이 예약된 날짜()에 지불하는 가격입니다ChargeDate.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 UpdatePaymentScheduleTerms 작업에 따라 다릅니다 AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다 StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP
용어	필수 만 허용됩니다 PaymentScheduleTerm . 목록 크기는 2보다 작아야 합니다.	422
용어[].유형	필수 만 가능 PaymentScheduleTerm	422
용어[].PaymentScheduleTerm.CurrencyCode	필수 지원되는 통화: ["USD", "AUD", "EUR", "GBP", "JPY"]	422
용어[].PaymentScheduleTerm.Schedule[]	필수	422
용어[].PaymentScheduleTerm.Schedule[].ChargeAmount	필수 날짜 유형은 "문자열"입니다. 최대 2개의 소수 자릿수가 지원되는 음수가 아닌 소수 자릿수	422
용어[].PaymentScheduleTerm.Schedule[].ChargeDate	필수 날짜는 "YYYY-MM-DD" 형식이어야 합니다.	422

비동기 오류

다음 오류는 UpdatePaymentScheduleTerms 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
DUPLICATE_™_DATES	PaymentScheduleTerm에 고유한 청구 날짜를 입력합니다.
호환되지 않는_통화_코드	제안이 릴리스된 후에는 CurrencyCode를 변경할 수 없습니다.
호환되지 않는_마크업_백분율	PaymentScheduleTerm은 MarkupPercentage가 0(0)보다 큰 경우 지원되지 않습니다.
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization과 호환되는 용어(들)를 제공합니다. 호환되지 않는 용어: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	PaymentScheduleTerm의 총 ChargeAmounts가 ResaleAuthorization과 호환되는지 확인합니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_™_DATES	AgreementEndDate 이전의 청구 날짜를 입력합니다.

오류 코드	오류 메시지
INVALID_CURRENCY_CODE	지원되는 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
TOO_MANY_BACKDATED_CHARGES	AvailabilityEndDate 전에 최대 1개의 예약된 결제를 제공합니다.
INVALID_TM_DATES	AgreementEndDate 이전의 마지막 청구 날짜를 입력합니다.
INVALID_TM_DATES	과거가 아닌 첫 번째 청구 날짜를 입력합니다.
TOO_MANY_TMS	PaymentScheduleTerm에서 최대 [x]개의 예약된 결제를 제공합니다.

갱신 옵션 수정

카탈로그 API를 사용하여이 제안을 사용하여 생성된 계약의 갱신 옵션을 제어할 수 있습니다AWS Marketplace.

카탈로그 API를 통해 생성된 제안의 경우 자동 갱신을 허용하도록 UpdateRenewalTerms 변경 유형을 호출할 때까지 자동 갱신은 기본적으로 비활성화된 상태로 유지됩니다. 이 변경 사항은 기존 계약에 영향을 주지 않습니다.

이 제안을 사용하여 생성된 계약의 갱신 옵션을 제어하려면 다음 예제와 같이 UpdateRenewalTerms 변경 유형을 사용하여 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateRenewalTerms",
```

```

    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "offer-123456789"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "RenewalTerm"
        }
      ]
    }
  ]
}

```

필드에 대한 정보를 제공하여 UpdateRenewalTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - 제안입니다.
- 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
- 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
- 용어(구조 배열) - 업데이트하려는 갱신 용어 목록입니다. 지원되는 갱신 기간은 다음과 같습니다.
- RenewalTerm(객체) - 계약의 정상적인 해지(구매자 또는 AWS Marketplace 취소가 ValidityTerm 아닌의 만료) 시 기존 계약의 수락된 조건을 사용하여 새 계약이 생성되도록 정의합니다. 즉, 계약이 갱신됩니다. 제안 RenewalTerm에가 있으면 자동 갱신이 허용됩니다. 구매자는 제안 수락/계약 생성 시 자동 갱신을 수락하거나 거부할 수 있습니다.
- 유형(문자열) - 업데이트되는 용어의 유형입니다. RenewalTerm

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```


변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 `UpdateRenewalTerms` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. `StartChangeSet`. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
용어	필수	422
용어[].유형	필수 "RenewalTerm"만 가능합니다.	422

비동기 오류

다음 오류는 `UpdateRenewalTerms` 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 `DescribeChangeSet` 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	제품에 대한 비공개 제안에서는 <code>RenewalTerm</code> 이 지원되지 않습니다.
INCOMPATIBLE_TERMS	<code>RenewalTerm</code> 은 <code>PaymentScheduleTerm</code> 과 함께 지원되지 않습니다.
INCOMPATIBLE_TERMS	<code>RenewalTerm</code> 은 <code>PricingModel</code> 에서 지원되지 않습니다.

오류 코드	오류 메시지
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.

제안 게시

카탈로그 API를 사용하여 모든 업데이트 변경 유형에서 수집된 정보를 병합한 다음 제안을 게시할 수 있습니다.

제안 `ReleaseOffer`은 가 호출될 때까지 Draft 상태로 유지됩니다. 제안이 릴리스되면에서 검색할 수 있습니다 AWS Marketplace.

제안을 게시하려면 다음 예제와 같이 `ReleaseOffer` 변경 유형으로 `StartChangeSet` API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

필드에 대한 정보를 제공하여 `ReleaseOffer` 변경 유형을 추가합니다.

- 엔터티(객체) - 생성 중인 엔터티의 명명된 유형입니다. Identifier는 제안 ID이고는 항상 Type입니다 Offer@1.0. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체) - 요청 세부 정보의 JSON 값입니다. 의 경우 비어 있어야 합니다 ReleaseOffer.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 ReleaseOffer 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
DetailsDocument	비어 있어야 합니다. ({})

비동기 오류

다음 오류는의 ReleaseOffer 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_PRODUCT	먼저 제품에 대한 공개 제안을 생성합니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.
INCOMPATIBLE_TARGETING	PreExistingAgreement는 구매자 대상 제안에만 지원됩니다.
INCOMPATIBLE_TARGETING	OfferSetId는 구매자 대상 제안에만 지원됩니다.
INVALID_TAX_INFORMATION	세금 정보가 불완전합니다. 전문 서비스를 판매하려면 DAC7 세금 설문지를 작성해야 AWS Marketplace합니다. 결제 정보 섹션으로 이동하여 DAC7 세금 양식을 선택합니다. 세금 정보를 업데이트하는 데 최대 2시간이 걸릴 수 있습니다.
INVALID_UPDATE_REQUEST	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
MISSING_AGREEMENT_END_DATE	대체 제안에 대한 AgreementEndDate를 제공합니다.
MISSING_AVAILABILITY_END_DATE	비공개 제안에 대한 AvailabilityEndDate를 제공합니다.
누락_구매자_계정	ResaleAuthorization을 사용하여 생성된 제안에 대해 PositiveTargeting에 BuyersAccounts를 제공합니다.
누락_구매자_계정	제품에 대한 모든 제안은 비공개여야 합니다. BuyersAccounts에 PositiveTargeting을 제공합니다.

오류 코드	오류 메시지
누락_설명	제안을 릴리스하기 전에 설명을 설정합니다.
MISSING_MANDATORY_TERMS	제안에 [x]를 추가합니다.
MISSING_MANDATORY_TERMS	제안에 PaymentScheduleTerm이 포함된 경우 FixedUpfrontPricingTerm을 제공합니다. PaymentScheduleTerm
누락_이름	제안을 릴리스하기 전에 이름을 설정합니다.
TOO_MANY_OFFERS	제품당 하나의 공개 무료 평가판 제안만 생성할 수 있습니다.
TOO_MANY_OFFERS	제품당 하나의 공개 제안만 생성할 수 있습니다.
MISSING_MANDATORY_TERMS	공개 제안에 대한 RenewalTerm에 제품의 계약 요금을 제공합니다.
MISSING_AGREEMENT_END_DATE	대체 제안에 대한 AgreementEndDate를 제공합니다.

기존 제안 세부 정보 설명

카탈로그 API를 사용하여에서 기존 제안 세부 정보를 설명할 수 있습니다AWS Marketplace.

기존 제안 세부 정보를 설명하려면 다음 예제와 같이 Offer@1.0 엔터티 유형으로 DescribeEntity API 작업을 호출합니다.

요청 구문

```
GET /DescribeEntity?catalog=<Catalog>&entityId=<EntityId> HTTP/1.1
```

필드에 대한 정보를 제공하여 DescribeEntity 변경 유형을 추가합니다.

- 카탈로그(문자열) - 요청과 관련된 카탈로그입니다. 고정 값: AWSMarketplace.
- entityId(문자열) - 설명할 제안의 고유 ID입니다.

응답 구문

이 요청에 대한 응답은 제안 세부 정보를 제공하며 다음과 같습니다.

```
{
  "EntityType": "Offer@1.0",
  "EntityIdentifier": "offer-ad8EXAMPLE51@1",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:111122223333:AWSMarketplace/Offer/offer-ad8EXAMPLE51",
  "LastModifiedDate": "2021-03-10T21:57:16Z",
  "DetailsDocument": {
    "Id": "offer-3rEXAMPLErn",
    "State": "Released",
    "Name": "Test Offer",
    "Description": "Worldwide offer for Test Product",
    "PreExistingAgreement": {
      "AcquisitionChannel": "External",
      "PricingModel": "Contract"
    },
    "ProductId": "prod-ad8EXAMPLE51",
    "OfferSetId": "offerset-b3f9EXAMPLE27",
    "Terms": [
      {
        "Type": "SupportTerm",
        "RefundPolicy": "If you need to request a refund for software sold by Amazon Web Services, LLC, please contact AWS Customer Service."
      },
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
          }
        ]
      }
    ],
    {
      "Type": "FreeTrialPricingTerm",
      "Duration": "P30D",
      "Grants": [
        {
          "DimensionKey": "m3.xlarge",
          "MaxQuantity": 10
        }
      ]
    }
  ]
}
```

```
{
  "DimensionKey": "m4.xlarge",
  "MaxQuantity": 10
}
],
},
{
  "Type": "ConfigurableUpfrontPricingTerm",
  "CurrencyCode": "USD",
  "RateCards": [
    {
      "Selector": {
        "Type": "Duration",
        "Value": "P365D"
      },
      "RateCard": [
        {
          "DimensionKey": "m3.large",
          "Price": "300.00"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "400.00"
        }
      ],
      "Constraints": {
        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
      }
    }
  ],
},
{
  "Type": "UsageBasedPricingTerm",
  "CurrencyCode": "USD",
  "RateCards": [
    {
      "RateCard": [
        {
          "DimensionKey": "m3.large",
          "Price": "0.10"
        },
        {
          "DimensionKey": "m4.xlarge",
```

```

        "Price": "0.20"
      }
    ]
  }
],
{
  "Type": "FixedUpfrontPricingTerm",
  "CurrencyCode": "USD",
  "Price": "200.00",
  "Grants": [
    {
      "DimensionKey": "Users",
      "MaxQuantity": 10
    }
  ]
},
{
  "Type": "RecurringPaymentTerm",
  "CurrencyCode": "USD",
  "BillingPeriod": "Monthly",
  "Price": "100.0"
},
{
  "Type": "PaymentScheduleTerm",
  "CurrencyCode": "USD",
  "Schedule": [
    {
      "ChargeDate": "2020-12-01T00:00:00.000Z",
      "ChargeAmount": "1000.00"
    },
    {
      "ChargeDate": "2021-06-15T00:00:00.000Z",
      "ChargeAmount": "1250.00"
    }
  ]
},
{
  "Type": "ByolPricingTerm"
},
{
  "Type": "RenewalTerm"
}
],

```



```

    "Rules": [
      {
        "Type": "TargetingRule",
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "CA"
          ],
          "BuyerAccounts": [
            "444455556666"
          ]
        },
        "NegativeTargeting": {
          "CountryCodes": [
            "XX"
          ]
        }
      },
      {
        "Type": "AvailabilityRule",
        "AvailabilityEndDate": "2024-08-30T01:56:03.000Z"
      }
    ]
  }
}

```

다음은 DescribeEntity 응답에 표시되는 필드에 대한 정보입니다.

- **EntityType**(문자열) - 엔터티의 명명된 유형으로, `입니다Offer@1.0`.
- **EntityIdentifier**(문자열) - 형식의 개체 식별자입니다 `EntityId@RevisionId`.
- **EntityArn**(문자열) - 이 요청에서 참조되는 변경 세트의 고유 식별자와 연결된 ARN입니다.
- **LastModifiedDate**(문자열) - 개체의 마지막 수정 날짜로, ISO 8601 형식입니다(예: `2018-02-27T13:45:22Z`).
- **세부 정보**(문자열) - 이 문자열화된 JSON 객체에는 개체에 대한 다음 세부 정보가 포함됩니다.
 - **Id**(문자열) - AWS Marketplace 및의 제안 엔터티에 대한 고유 식별자는 제안 생성 중에 생성됩니다.
 - **상태**(문자열) - 제안의 상태입니다.
 - **이름**(문자열) - 사용자와 고객의 가독성을 높이기 위해 제안과 연결된 이름입니다. 계약 정보의 일부로도 표시됩니다.

- 설명(문자열) - 설명은 본인만 사용할 수 있으며 구매자에게 절대 노출되지 않는 자유 형식 텍스트입니다.
- PreExistingAgreement(문자열) - 이 제안이 동일한 기본 제품에 대해 기존 고객과 체결한 기존 계약의 갱신인지 여부를 결정합니다. 기존 계약은 내부 또는 외부에 있을 수 있습니다. AWS Marketplace는 제안을 감사하고 갱신되었는지 확인할 수 있습니다. AWS가 제안을 확인할 수 없는 경우 AWS는 고객의 제안 및 권한을 취소할 수 있습니다.
- AcquisitionChannel(문자열) - 기존 계약이 외부 AWS Marketplace 또는 내부에서 서명되었는지 여부를 나타냅니다. 가능한 값: External, AwsMarketplace.
- PricingModel(문자열) - 기존 계약에서 사용하는 요금 모델을 나타냅니다. 가능한 값: Contract, Usage, Byol, Free.
- ProductId(문자열) - 제공되는 제품의 고유 식별자입니다.
- OfferSetId(문자열) - 이 제안을 연결할 제안 세트의 고유 식별자입니다.
- 용어(구조 배열) - 용어 목록입니다.
- 규칙(구조 배열) - 규칙 목록입니다.

AWS Marketplace APIs를 사용하여 재판매 권한 부여 작업

를 사용하여 재판매 권한 부여 작업을 자동화 AWS Marketplace Catalog API 할 수 있습니다.

제품은에서 판매 중인 제품을 설명하지만 AWS Marketplace 재판매 권한 부여(기회라고도 함)는 이 제품을 재판매할 수 있는 권한 부여 방법과 관련된 용어와 규칙을 설명합니다. CPPO는 재판매 권한 부여의 대상입니다.

재판매 승인에는 제조업체와 채널 파트너 간의 리셀러 계약에 대해 수락할 약관 및 규칙 모음이 있습니다. 재판매 승인 약관을 수락하면 리셀러는 약관에 명시된 조건에 따라 제품에 대한 제안을 생성할 수 있습니다.

재판매 권한 부여에는 두 가지 유형의 규칙이 있습니다.

- AvailabilityRule -에서 재판매 권한 부여의 수명 주기를 제어합니다.
- PartnerTargetingRule - 특정 채널 파트너 집합이 재판매 권한 부여에 액세스할 수 있어야 하는지 여부를 지정합니다.

다음 리소스를 참조하세요.

- 작업 코드 예제가 포함된 end-to-end 랩은 AWS Marketplace 판매자 워크숍의 [랩: 리셀러 권한 부여](#)를 참조하세요.
- API 요청의 코드 예제는 GitHub의 AWS 샘플에서 [Python](#) 및 [Java](#) 예제를 참조하세요.
- 재판매 권한 부여 생성에 대한 동영상은 YouTube의 [AWS Marketplace 카탈로그 API를 사용하여 재판매 권한 부여 생성을 참조하세요](#).

다음 주제에서는 카탈로그 API를 사용하여 재판매 권한 부여를 생성하고 업데이트하는 방법을 설명합니다.

주제

- [재판매 권한 부여 사전 조건](#)
- [새 재판매 권한 부여 생성](#)
- [구매자 대상 업데이트](#)
- [가용성 업데이트](#)
- [미래 날짜의 계약의 유효성 업데이트](#)
- [법률 리소스 업데이트](#)
- [요금 업데이트](#)
- [결제 일정 업데이트](#)
- [재판매 권한 부여 세부 정보 업데이트](#)
- [재판매 권한 부여 제한](#)
- [재판매 권한 부여를 해제하고 채널 파트너에게 표시](#)
- [기존 재판매 권한 부여 설명](#)

재판매 권한 부여 사전 조건

재판매 승인을 사용하려면 독립 소프트웨어 공급업체(ISVs)와 AWS Marketplace 채널 파트너 모두 리소스 공유 권한을 제공하는 서비스 연결 역할을 생성해야 합니다. 두 그룹 모두 이 사전 조건을 수행하지 않는 경우 AWS는 ISV의 권한 부여 리소스를 AWS Marketplace 채널 파트너와 공유할 수 없습니다. 자세한 내용은 AWS Marketplace 판매자 안내서의 [대한 재판매 권한 부여를 위한 역할 사용을 AWS Marketplace](#) 참조하세요.

새 재판매 권한 부여 생성

카탈로그 API를 사용하여 새 재판매 권한 부여를 생성할 수 있습니다 AWS Marketplace.

요청이 성공적으로 처리되면 Draft 상태의 재판매 승인을 AWS Marketplace Catalog API 생성합니다. 불완전한 재판매 권한 부여이며 채널 파트너에게 표시되지 않습니다 AWS Marketplace.

Update 변경 유형을 사용하여 재판매 승인을 완료합니다. 재판매 권한 부여가 완료되면 ReleaseResaleAuthorization 변경 유형을 사용하여 재판매 권한 부여 생성 프로세스를 완료하고 재판매 권한 부여를 릴리스합니다. 그러면 전체 재판매 권한 부여를 검증하고 채널 파트너가 이를 볼 수 있습니다 AWS Marketplace.

Draft 상태에서 재판매 권한 부여를 생성하려면 다음 예제와 같이 CreateResaleAuthorization 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "xyz",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument":
      {
        "ProductId": "prod-ad8EXAMPLE51",
        "Name": "Test ResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "777788889999"
      }
    }
  ]
}
```

입력 필드에 대한 정보를 제공하여 CreateResaleAuthorization 변경 유형을 추가합니다.

- 엔터티(객체)(필수) - 재판매 승인.
 - 유형(문자열)(필수) - 는 항상 Type입니다 ResaleAuthorization@1.0.

- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
- ProductId(문자열)(필수) - 재판매 권한 부여를 생성할 제품 ID입니다.
- 이름(문자열)(필수) - 사용자와 채널 파트너의 가독성을 높이기 위해 ResaleAuthorization과 연결된 이름입니다.
- 설명(문자열)(선택 사항) - ResaleAuthorization에 대한 세부 정보를 추가할 수 있는 자유 형식 텍스트 필드입니다.
- ResellerAccountId(문자열)(필수) - 이를 설명하고 사용하여 비공개 제안을 ResaleAuthorization 생성할 수 AWS 계정있는 대상 채널 파트너의를 추가합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

요청이 완료되면(가 Status인 경우SUCCEEDED) 새 ResaleAuthorization가 생성됩니다. SUCCEEDED 상태는 CreateResaleAuthorization 변경 유형 호출이 완료되었음을 나타내지만 ResaleAuthorization 상태는 여전히 Draft 상태입니다.

다음은 [DescribeChangeSet](#) API 작업의 응답을 보여줍니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2021-05-27T22:21:26Z",
  "EndTime": "2021-05-27T22:32:19Z",
  "Status": "SUCCEEDED",
  "ChangeSet":

```

```
[
  {
    "ChangeType": "CreateResaleAuthorization",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "resaleauthz-123456789"
    },
    "DetailsDocument": {
      "ProductId": "prod-ad8EXAMPLE51",
      "Name": "Test ResaleAuthorization",
      "Description": "Worldwide ResaleAuthorization for Test Product",
      "ResellerAccountId": "777788889999",
      "BulkRequestId": "84977023-5093-4a66-8b24-ef2c5a2f8b1f"
    },
    "ErrorDetailList": []
  }
]
```

동기 검증

스키마 검증은의 CreateResaleAuthorization 작업에 따라 다릅니다AWS Marketplace Catalog API. 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
ProductId	필수 null이거나 비어 있으면 안 됩니다. 길이는 1~50자여야 합니다.	422
ProductId	사용자는 지정된 제품에 대한 ResaleAuthorization 생성할 권한이 있어야 합니다.	403
ProductId	카탈로그의 기존 제품이어야 하며 Draft 상태가 아니어야 합니다.	404

입력 필드	검증 규칙	HTTP 코드
	제품을 재판매할 수 있도록 지원해야 합니다.	
이름	필수 null이거나 비어 있으면 안 됩니다. 길이는 1~100자여야 합니다. 특수 문자는 허용되지 않음	422
설명	선택 사항 길이는 1~255자여야 합니다. 특수 문자는 허용되지 않음	422
ResellerAccountId	필수 비어 있으면 안 됩니다. AWS 계정IDs 유효한 형식(12자리 숫자)이어야 합니다.	422
BulkRequestId	선택 사항 길이는 1~50자여야 합니다. UUID 형식이어야 합니다.	422
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 `CreateResaleAuthorization` 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 `DescribeChangeSet` 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_RESELLER_ACCOUNT	유효한 리셀러 계정을 제공합니다.

구매자 대상 업데이트

카탈로그 API를 사용하여 재판매 승인을 대상으로 하는 구매자를 업데이트할 수 있습니다AWS Marketplace.

최신 요청에 포함되지 않은 기존 대상 지정 옵션은 재판매 권한 부여에서 제거됩니다. 이 변경 유형은 재판매 권한 부여 릴리스의 경우 선택 사항입니다.

재판매 권한 부여를 대상으로 하는 구매자를 업데이트하려면 다음 예제와 같이 UpdateBuyerTargetingTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateBuyerTargetingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "BuyerTargetingTerm",
            "PositiveTargeting": {
              "BuyerAccounts": [
                "123456789012"
              ]
            }
          }
        ]
      }
    }
  ]
}
```



```

    }
  }
]
}
]
}

```

필드에 대한 정보를 제공하여 UpdateBuyerTargetingTerms 변경 유형을 추가합니다.

- 엔터티(객체)(필수) - 재판매 승인.
 - 유형(문자열)(필수) -는 항상 Type입니다 ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
 - 용어(구조 배열)(선택 사항) - 업데이트하려는 용어를 대상으로 하는 구매자 목록입니다. ResaleAuthorization 특정 구매자를 대상으로 하지 않으려는 경우 용어 필드를 건너뛸 수 있습니다. 기본적으로 ResaleAuthorization는 모든 구매자를 대상으로 합니다. 지원되는 용어는 다음과 같습니다.
 - BuyerTargetingTerms(객체)(선택 사항) - ResaleAuthorization에 대한 구매자별 대상 지정을 정의합니다.
 - 유형(문자열)(필수) - 업데이트 중인 용어의 범주입니다.
 - PositiveTargeting(객체)(필수) - 구매자의 프로필이에 대한 액세스를 허용하기 위해 충족해야 하는 기준을 정의합니다 ResaleAuthorization.
 - BuyerAccounts(문자열 배열)(선택 사항) - 선택 사항으로 나열합니다. 대상 구매자의를 추가할 수 있습니다 AWS 계정. 특정 구매자를 대상으로 ResaleAuthorization 하지 않으려는 경우이 필드를 생략해야 합니다. 기본적으로 모든 구매자를 대상으로 합니다. 지정된 경우 대상 채널 파트너는 비공개 제안을 생성하고 구매자 하위 집합을 대상으로 지정할 수 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",

```

```
"ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 판매자 운영 팀과 정보를 검증하여 지침을 충족하는 AWS Marketplace 지 확인하는 것이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

스키마 검증은 `UpdateBuyerTargetingTerms` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다. `StartChangeSet`. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
용어	선택 사항 null이거나 비어 있으면 안 됩니다. 목록에는 "BuyerTargetingTerm"만 허용됩니다. 목록 크기는 10이어야 합니다 (현재 여러 구매자 조건이 필요한 사용 사례는 없음).
BuyerTargetingTerm.PositiveTargeting	필수 비어 있으면 안 됩니다.
BuyerTargetingTerm.PositiveTargeting.BuyerAccounts	선택 사항 AWS 계정IDs 유효한 형식(12자리 숫자)이어야 합니다.

입력 필드	검증 규칙
	25개 이상의 계정을 포함할 수 없습니다.
알 수 없는 속성	추가 속성은 허용되지 않습니다.

비동기 오류

다음 오류는 UpdateBuyerTargetingTerms 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_BUYER_TARGETING	PreExistingBuyerAgreement를 사용한 ResaleAuthorization에는 하나 이상의 구매자 계정이 있어야 합니다.

가용성 업데이트

카탈로그 API를 사용하여 생성된 비공개 제안 수 또는 비공개 제안을 생성할 수 있는 특정 시점까지의 가용성을 제한할 수 있습니다.

규칙 목록에서 가용성을 확인할 수 있지만 기본적으로 이 값은 이 재판매 권한 부여의 무제한 사용입니다.

재판매 권한 부여의 가용성과 사용성을 제어하려면 다음 예제와 같이 UpdateAvailability 변경 유형을 사용하여 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
```

```

"Catalog": "AWSMarketplace",
"ChangeSet":
[
  {
    "ChangeType": "UpdateAvailability",
    "Entity":
    {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "resaleauthz-123456789"
    },
    "DetailsDocument":
    {
      "AvailabilityEndDate": "2022-05-31",
      "OffersMaxQuantity": 1
    }
  }
]
}

```

필드에 대한 정보를 제공하여 UpdateAvailability 변경 유형을 추가합니다.

- 엔터티(객체)(필수) - 재판매 승인.
 - 유형(문자열)(필수) -는 항상 Type입니다 ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
 - AvailabilityEndDate(문자열)(선택 사항) - 채널 파트너가를 활용하여 제안을 ResaleAuthorization 생성할 수 있을 때까지 종료 날짜를 정의합니다. 채널 파트너는 지정된 종료 날짜까지 이를 ResaleAuthorization 여러 번 사용할 수 있습니다. 날짜는 ISO_8601 형식으로 표시됩니다.
 - OffersMaxQuantity(정수)(선택 사항) - ResaleAuthorization을 사용하여 생성할 수 있는 최대 비공개 제안 수를 정의합니다. 구독 수는 정의되지 않습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",

```

```
"ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 판매자 운영 팀과 정보를 검증하여 지침을 충족하는 AWS Marketplace 지 확인하는 것이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

스키마 검증은 `UpdateAvailability` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 검증은 `StartChangeSet` 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
<code>OffersMaxQuantity</code>	선택 사항 음수가 아닌 정수여야 합니다. 허용된 값만 "1"(현재 여러 수량을 지원하는 사용 사례 없음)
<code>AvailabilityEndDate</code>	선택 사항 ISO_8601 형식이어야 합니다. 미래 날짜여야 합니다.
가용성	<code>OffersMaxQuantity</code> 또는 <code>AvailabilityEndDate</code> 를 제공합니다.
알 수 없는 속성	추가 속성은 허용되지 않습니다.

비동기 오류

다음 오류는 AWS Marketplace Catalog API의 `UpdateAvailability` 작업에만 해당됩니다. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청

의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_AVAILABILITY_END_DATE	ResalePaymentScheduleTerms의 모든 ChargeDate 앞에 있는 AvailabilityEndDate를 제공합니다. ChargeDate
INVALID_AVAILABILITY_END_DATE	향후 AvailabilityEndDate를 제공합니다.

미래 날짜의 계약의 유효성 업데이트

카탈로그 API를 사용하여 미래 날짜의 서비스 시작 날짜를 수정하고 제어할 수 있습니다AWS Marketplace.

이 변경 세트는 재판매 권한 부여를 해제하는 데 필수는 아닙니다.

재판매 권한 부여의 제품 계약 기간을 수정하고 제어하려면 다음 예제와 같이 UpdateBuyerValidityTerms 변경 유형을 사용하여 StartChangeSet API 작업을 호출합니다.

Note

장래 날짜의 계약은 SaaS 제품 유형에 대해서만 지원됩니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateBuyerValidityTerms",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
```

```

    "Identifier": "resaleauthz-123456789"
  },
  "DetailsDocument":
  {
    "Terms":
    [
      {
        "Type": "BuyerValidityTerm",
        "MaximumAgreementStartDate": "2024-05-31"
      }
    ]
  }
}

```

입력 필드에 대한 정보를 제공하여 UpdateBuyerValidityTerms 변경 유형을 추가합니다.

- 엔터티(객체)(필수) - 재판매 권한 부여.
 - 유형(문자열)(필수) -는 항상 Type입니다 ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
 - 용어(구조 배열) - 업데이트하려는 계약 유효 기간 목록입니다. 지원되는 용어는 다음과 같습니다.
 - BuyerValidityTerm(객체) - ResaleAuthorization에서 제품에 대한 서비스의 가용성을 정의합니다.
 - 유형(문자열) - 업데이트 중인 용어의 범주입니다.
 - MaximumAgreementStartDate(문자열)(필수) - 제공된 제품의 계약 시작 날짜를 정의합니다. 미래 날짜의 제안은 이 서비스 시작 날짜를 초과할 수 없습니다. 날짜는 ISO_8601 형식으로 표시됩니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

}

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 판매자 운영 팀과 정보를 검증하여 지침을 충족하는 AWS Marketplace 지 확인하는 것이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

스키마 검증은 `UpdateBuyerValidityTerms` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 검증은 호출할 때 수행됩니다. `StartChangeSet`. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
용어	필수 null이거나 비어 있으면 안 됩니다. 목록에는 "BuyerValidityTerm"만 허용됩니다. 목록 크기는 1이어야 합니다(현재 여러 서비스 가용성 조건이 필요한 사용 사례는 없음).
MaximumAgreementStartDate	필수 null이거나 비어 있으면 안 됩니다. 미래 날짜여야 하며 지금부터 3년을 초과해서는 안 됩니다. ISO_8601 형식이어야 합니다.
알 수 없는 속성	추가 속성은 허용되지 않습니다.

비동기 오류

다음 오류는 `UpdateBuyerValidityTerms` 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 `DescribeChangeSet` 후를 호출할 때 반환됩니다. 를 사용하

여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [요변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	제품에는 BuyerValidityTerm이 지원되지 않습니다.
INVALID_MAXIMUM_AGREEMENT_START_DATE	허용된 한도에 향후 MaximumAgreementStartDate를 입력합니다.

법률 리소스 업데이트

카탈로그 API를 사용하여에서 기존 법률 용어를 완전히 바꿀 수 있습니다AWS Marketplace.

최신 요청에 포함되지 않은 법적 약관은 재판매 권한 부여에서 제거됩니다. BuyerLegalTerm에는 최종 구매자 계약에 포함될 EULA와 채널 파트너와 ISV 간의 리셀러 계약에 포함될 리셀러 계약이 LegalTerm 포함되어 있습니다.

의 법적 조건을 업데이트하려면 다음 예제와 같이 UpdateLegalTerms 변경 유형으로 StartChangeSet API 작업을 ResaleAuthorization호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument":
      {
```

```

    "Terms":
    [
      {
        "Type": "BuyerLegalTerm",
        "Documents":
        [
          {
            "Type": "CustomEula",
            "Url": "https://my-public-bucket.s3.amazonaws.com/eula-
example12345.txt"
          }
        ]
      },
      {
        "Type": "ResaleLegalTerm",
        "Documents":
        [
          {
            "Type": "CustomResellerContract",
            "Url": "https://my-public-bucket.s3.amazonaws.com/reseller-
example12345.txt"
          }
        ]
      }
    ]
  }
}

```

필드에 대한 정보를 제공하여 UpdateLegalTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - 재판매 승인.
 - 유형(문자열)(필수) -는 항상 Type입니다ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
 - 용어(구조 배열)(필수) - 법적 용어 목록입니다. 지원되는 법률 용어는 다음과 같습니다.
 - BuyerLegalTerm(객체)(필수) - 수락자에게 제안할 텍스트 계약 목록을 정의합니다. 최종 사용자 라이선스 계약(EULA)을 예로 들 수 있습니다.
 - 유형(문자열)(필수) - 업데이트 중인 용어의 범주입니다.

- 문서(구조 배열)(필수) - 구매자에게 제안할 법적 리소스에 대한 참조 목록입니다. EULA를 예로 들 수 있습니다. 각 참조는 Type 및 로 구성됩니다URL.
- 유형(문자열)(필수) - 문서 유형입니다. 사용 가능한 문서 유형은 다음과 같습니다.
 - StandardEula - 표준 계약AWS Marketplace. 자세한 내용은 AWS Marketplace 판매자 안내서의 [SCMP](#)를 참조하세요. 이 유형은에서 관리하므로 URL을 제공할 필요가 없습니다AWS Marketplace.
 - EnterpriseEula -에 대한 엔터프라이즈 계약입니다AWS Marketplace. 자세한 내용은 AWS Marketplace판매자 안내서의 DSA를 참조하세요. 이 유형은에서 관리하므로 URL을 제공할 필요가 없습니다AWS Marketplace.
 - CustomEula - 제조업체가 제공하는 사용자 지정 EULA입니다. 이 문서 유형에는 액세스 가능한 S3 버킷에 저장된 EULA의 URL이 필요합니다.
- URL(문자열)(조건부 필수) - 구매자가 읽을 수 있는 법률 문서의 URL입니다. 이는 범주 유형이 인 경우에 필요합니다CustomEula.
- ResaleLegalTerm(객체)(선택 사항) - 채널 파트너에게만 제안할 텍스트 계약 목록을 정의합니다. 이 기간은 구매자가 사용할 수 없습니다.
- 유형(문자열)(필수) - 업데이트 중인 용어의 범주입니다.
- 문서(구조 배열)(필수) - 채널 파트너에게 제안할 리셀러 법률 리소스에 대한 참조 목록입니다.
 - 유형(문자열)(필수) - 문서의 범주입니다. 사용 가능한 문서 유형은 다음과 같습니다.
 - StandardResellerContract - 표준 리셀러 계약AWS Marketplace.
 - CustomResellerContract - 제조업체가 체결한 사용자 지정 리셀러 계약입니다. 리셀러 계약의 URL은 액세스 가능한 S3 버킷에 저장되며이 문서 유형에 필요합니다.
 - URL(문자열)(조건부 필수) - 채널 파트너가 읽을 수 있는 리셀러 계약 문서의 URL입니다. 유형이 CustomResellerContract인 경우 필요합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

스키마 검증은 `UpdateLegalTerms` 작업에 따라 다릅니다. AWS Marketplace Catalog API. 검증은 `StartChangeSet`을 호출할 때 수행됩니다. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
용어	필수 null이거나 비어 있으면 안 됩니다.	422
용어[].BuyerLegalTerm	필수 null이거나 비어 있으면 안 됩니다.	422
용어[].ResaleLegalTerm	선택 사항 있는 경우 null이거나 비어 있으면 안 됩니다.	422
용어[].BuyerLegalTerm.Documents	필수 null이거나 비어 있으면 안 됩니다.	422
용어[].BuyerLegalTerm.Documents[]. 유형	필수 null이거나 비어 있으면 안 됩니다. 허용된 값: - StandardEula - EnterpriseEula - CustomEula	422

입력 필드	검증 규칙	HTTP 코드
용어[].BuyerLegalTerm.Documents[].URL	필수이며 "Type"이 "CustomEula"인 경우 유효한 URL이어야 합니다. "Type"이 ["StandardEula", "EnterpriseEula"] 중 하나인 경우 제공해서는 안 됩니다.	422
용어[].ResaleLegalTerm.Documents	필수 null이거나 비어 있으면 안 됩니다.	422
용어[].ResaleLegalTerm.Documents[].유형	필수 null이거나 빈 값이어서는 안 됩니다. 허용된 값: • StandardEula • CustomResellerContract	422
용어[].ResaleLegalTerm.Documents[].URL	필수이며 "Type"이 "CustomResellerContract"인 경우 유효한 URL이어야 합니다. "Type"이 ["StandardContract"] 중 하나인 경우 제공해서는 안 됩니다.	422
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 UpdateLegalTerms 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_BUYER_LEGAL_DOCUMENTS	액세스 가능한 S3 버킷에 저장된 구매자 법률 문서의 URLs을 제공합니다.
유효하지 않은_재판매_법적_문서	액세스 가능한 S3 버킷에 저장된 재판매 법률 문서의 URLs을 제공합니다.
MISSING_MANDATORY_TERMS	BuyerLegalTerm을 제공합니다.

요금 업데이트

카탈로그 API를 사용하여에서 기존 요금 조건을 완전히 바꿀 수 있습니다AWS Marketplace.

최신 요청에 포함되지 않은 요금 조건은 재판매 권한 부여에서 제거됩니다. 이 API를 통해 제품의 할인 요금을 업데이트할 수 있습니다.

재판매 권한 부여에 대한 요금 세부 정보를 업데이트하려면 다음 예제와 같이 UpdatePricingTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument":
      {
        "PricingModel": "Contract",
        "Terms":
        [
```

```
{
  "Type": "ResaleUsageBasedPricingTerm",
  "CurrencyCode": "USD",
  "RateCards":
  [
    {
      "RateCard":
      [
        {
          "DimensionKey": "m3.large",
          "Price": "0.10"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "0.20"
        }
      ]
    }
  ],
},
{
  "Type": "ResaleConfigurableUpfrontPricingTerm",
  "CurrencyCode": "USD",
  "RateCards":
  [
    {
      "Selector":
      {
        "Type": "Duration",
        "Value": "P12M"
      },
      "RateCard":
      [
        {
          "DimensionKey": "m3.large",
          "Price": "300"
        },
        {
          "DimensionKey": "m4.xlarge",
          "Price": "400"
        }
      ],
      "Constraints":
      {
```

```

        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
    }
}
],
},
{
    "Type": "ResaleFixedUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "Duration": "P2M",
    "Price": "200.0",
    "Grants":
    [
        {
            "DimensionKey": "Users",
            "MaxQuantity": 10
        }
    ]
}
]
}
}
]
}
}

```

필드에 대한 정보를 제공하여 UpdatePricingTerms 변경 유형을 추가합니다.

- 엔터티(객체)(필수) - 재판매 권한 부여.
 - 유형(문자열)(필수) -는 항상 Type입니다 ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
 - PricingModel(문자열)(필수) - 제안에 대한 요금 모델입니다. 요금 모델의 가능한 값은 다음과 같습니다.
 - 사용량 - 구매자에게 제품 사용량에 대한 요금이 청구되는 사용량 기반 요금 모델입니다.
 - 계약 - 계약 기반 요금 모델에서 구매자에게는 제품 사용에 대한 요금이 미리 청구되거나 유연한 결제 일정이 제공됩니다. 구매자는 계약 이상의 추가 사용량에 대해서도 비용을 지불할 수 있습니다. 채널 파트너는 각 차원의이 결제 일정 및 요금에 마크업을 추가할 수 있습니다.
 - 용어(구조 배열)(필수) - 업데이트하려는 요금 조건 목록입니다. 지원되는 요금 조건은 다음과 같습니다.

- ResaleUsageBasedPricingTerm(객체) - 제품 사용량에 따라 고객에게 요금이 부과되는 종pay-as-you-go(PAYG) 요금 모델을 정의합니다.
- 유형(문자열)(필수) - 용어의 범주입니다.
- CurrencyCode(문자열) -이 용어에 언급된 가격의 통화를 정의합니다. 현재는 USD만 지원됩니다.
- RateCards(구조 배열) - 요금 카드 목록입니다.
- RateCard(구조 배열) - Rate Card는 제품 차원의 단위당 요금을 정의합니다.
 - DimensionKey(문자열) - 지정된 권한이 적용되는 차원입니다. 차원은 제품의 용량 범주를 나타내며 제품이에 나열될 때 지정됩니다AWS Marketplace.
 - 가격(문자열) - 청구할 금액을 계산하는 데 사용되는 제품 차원의 단위 가격입니다.
- 제약 조건(객체)(선택 사항) - 수락자가 용어를 구성하는 방법에 대한 제한을 정의합니다.
 - MultipleDimensionSelection(문자열)(선택 사항) - 구매자가 요금 카드에서 여러 차원을 선택할 수 있는지 여부를 결정합니다. 가능한 값은 Allowed 및 Disallowed입니다. 기본값은 Allowed입니다.
 - QuantityConfiguration(문자열)(선택 사항) - 수락자가 요금 카드의 각 차원에 대한 수량을 구성할 수 있는지 여부를 결정합니다. 가능한 값은 Allowed 및 Disallowed입니다. 기본값은 Allowed입니다.
- ResaleFixedUpfrontPricingTerm(객체) - 고객에게 고정 선결제 금액이 청구되는 선결제 요금 모델을 정의합니다.
- 유형(문자열)(필수) - 업데이트 중인 용어의 범주입니다.
- CurrencyCode(문자열) -이 기간에 언급된 가격의 통화를 정의합니다. 이 용어에 언급된 가격의 통화를 정의합니다. USD, AUD, EUR, GBP 및 JPY가 지원됩니다.
- 가격(문자열)(필수) -이 기간이 수락될 때 고객에게 청구되는 고정 금액입니다.
- 기간(문자열)(필수) - ResaleAuthorization 부여의 계약 기간입니다. 이 필드는 ISO 8601 형식을 지원합니다.
- 권한 부여(구조 배열)(필수) - 계약 실행의 일부로 고정 선결제 요금을 수락자에게 부여할 권한입니다.
 - DimensionKey(문자열)(필수) - 제품 문서에 정의된 고유한 차원 키입니다. 차원은 제품의 용량 범주를 나타내며 제품이에 나열될 때 지정됩니다AWS Marketplace.
 - MaxQuantity(정수)(필수) - 구매자에게 제품의 지정된 차원에 대한 권한을 부여할 수 있는 최대 용량입니다. MaxQuantity가 제공되지 않은 경우 구매자는 지정된 차원을 무제한으로 사용할 수 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 AWS Marketplace 카탈로그 API의 UpdatePricingTerms 작업에 따라 다릅니다. 검증은호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
용어	필수 null이거나 비어 있으면 안 됩니다. 각 용어는 일회성 허용 용어만 표시해야 합니다. * ResaleUsageBasedPricingTerm * ResaleConfigurableUpfrontPricingTerm * ResaleFixedupfrontPricingTerm
용어[].ResaleUsageBasedPricingTerm.CurrencyCode	필수 허용되는 값: USD
용어[].ResaleUsageBasedPricingTerm.Validity	필수

입력 필드	검증 규칙
	null이거나 비어 있으면 안 됩니다. 예상 형식: ISO 8601 기간
<code>용어[].ResaleUsageBasedPricingTerm.RateCards</code>	필수 null이거나 비어 있으면 안 됩니다.
<code>용어[].ResaleUsageBasedPricingTerm.RateCards[].DimensionKey</code>	필수 null이거나 비어 있으면 안 됩니다. 길이는 1~60이어야 합니다.
<code>용어[].ResaleUsageBasedPricingTerm.RateCards[].가격</code>	필수 null이거나 비어 있으면 안 됩니다. 데이터 형식은 "문자열"입니다. 음수가 아니어야 합니다. 최대 8십진수 지원 특수 문자가 지원되지 않음
<code>용어[].ResaleConfigurableUpfrontPricingTerm.CurrencyCode</code>	필수 허용되는 값: ["USD", "AUD", "EUR", "GBP", "JPN"]
<code>용어[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Selector.Type</code>	필수 null이거나 비어 있으면 안 됩니다. 허용된 값: 기간

입력 필드	검증 규칙
<code>용어[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Selector.Value</code>	필수 null이거나 비어 있으면 안 됩니다. 예상 형식: ISO 8601 기간
<code>용어[].ResaleConfigurableUpfrontPricingTerm.RateCards[].RateCard.DimensionKey</code>	필수 null이거나 비어 있으면 안 됩니다. 길이는 1~60이어야 합니다.
<code>용어[].ResaleConfigurableUpfrontPricingTerm.RateCards[].RateCard.Price</code>	필수 null이거나 비어 있으면 안 됩니다. 데이터 형식은 "문자열"입니다. 음수가 아니어야 합니다. 최대 6십진수 지원 특수 문자가 지원되지 않음
<code>용어[].ResaleConfigurableUpfrontPricingTerm.RateCards[].계약 조건</code>	선택 사항
<code>용어[].ResaleFixedUpfrontPricingTerm.CurrencyCode</code>	필수 허용되는 값: ["USD", "AUD", "EUR", "GBP", "JPN"]

입력 필드	검증 규칙
용어[].ResaleFixedUpfrontPricingTerm.Price	필수 null이거나 비어 있으면 안 됩니다. 데이터 형식은 "문자열"입니다. 음수가 아니어야 합니다. 최대 6십진수 지원 특수 문자가 지원되지 않음 허용되는 값: 0.0
용어[].ResaleFixedUpfrontPricingTerm.Duration	필수 null이거나 비어 있으면 안 됩니다. 예상 형식: ISO 8601 기간
용어[].ResaleFixedUpfrontPricingTerm.Grants[].DimensionKey	필수 null이거나 비어 있으면 안 됩니다. 길이는 1~60이어야 합니다.
용어[].ResaleFixedUpfrontPricingTerm.Grants[].MaxQuantity	필수 null이거나 비어 있으면 안 됩니다.
알 수 없는 속성	추가 속성은 허용되지 않습니다.

비동기 오류

다음 오류는 UpdatePricingTerms 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
INCOMPATIBLE_PRODUCT	[x]의 제품에서 사용 가능한 기존 차원을 사용합니다.
DUPLICATE_DIMENSION_KEYS	[x]에서 차원 키의 고유한 목록을 비율 카드에 제공
INVALID_RATE_CARD	[x]에서 단위가 동일한 차원 제공
INVALID_RATE_CARD	ResaleUsageBasedPricingTerm에서 측정된 차원에 대해서만 요금 카드를 제공합니다.
INVALID_RATE_CARD	ResaleUsageBasedPricingTerm에서 사용할 수 있는 모든 측정 차원에 대한 사용량 기반 요금을 제공합니다.
TOO_MANY_RATES	ResaleUsageBasedPricingTerm의 허용된 한도 내에서 RateCards를 제공합니다. ResaleUsageBasedPricingTerm
DUPLICATE_SELECTORS	ResaleConfigurableUpfrontPricingTerm에 고유한 선택기 목록을 제공합니다.
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm에 기간 [x] 동안 하나 이상의 차원 키가 누락되었습니다. 모든 기간 동안 동일한 차원 키 세트에 대한 가격을 제공합니다.
INVALID_RATE_CARD	모든 측정된 차원 또는 권한이 있는 모든 차원을 [x]로 입력합니다.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	PricingModel 대한 ResaleConfigurableUpfrontPricingTerm에서 MultipleDimensionSelection 및 QuantityConfiguration을 허용되지

오류 코드	오류 메시지
	없음으로 설정합니다. QuantityConfiguration ResaleConfigurableUpfrontPricingTerm
TOO_MANY_RATE_CARDS	ConfigurableUpfrontPricingTerm의 요금 카드 하나만 제품에 허용됩니다.
INCOMPATIBLE_TERMS	다음 조건은 PricingModel [x,y,z]과 호환되지 않습니다.
TOO_MANY_RATES	허용되는 한도 내에서 RateCards를 [x term]으로 제공합니다.
TOO_MANY_GRANTS	[x term]에 최대 [N]개의 권한 부여를 제공합니다.
INVALID_SELECTOR_DURATION_VALUE	ResaleConfigurableUpfront에서 [x]~[y]개월의 기간 제공
TOO_MANY_GRANTS	[x]~[y]개월의 기간을 입력합니다.
INVALID_SELECTOR_DURATION_VALUE	ResaleConfigurableUpfront에서 측정된 차원의 지속 시간 세부 수준이 일 수준인지 확인
INVALID_DURATION	FixedUpfront에서 측정된 차원의 지속 시간 세부 수준이 일 수준인지 확인합니다.
INVALID_RATE_CARD	[x]에 권한이 있는 차원만 제공합니다.
누락_지속 시간	[x]로 기간을 입력합니다.
DUPLICATE_DIMENSION_KEYS	Grants에 [x]의 고유한 차원 키 목록을 제공합니다.
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.

오류 코드	오류 메시지
INVALID_CURRENCY_CODE	지원되는 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
호환되지 않는_통화_코드	제안이 릴리스된 후에는 CurrencyCode를 변경할 수 없습니다.

결제 일정 업데이트

카탈로그 API를 사용하여 유연한 결제 일정과 같은 결제 관련 세부 정보를 변경할 수 있습니다AWS Marketplace.

재판매 권한 부여에 대한 결제 관련 세부 정보를 업데이트하려면 다음 예제와 같이 UpdatePaymentScheduleTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdatePaymentScheduleTerms",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument":
      {
        "Terms":
        [
          {
            "Type": "ResalePaymentScheduleTerm",
            "CurrencyCode": "USD",
```



```

    "Schedule":
    [
      {
        "ChargeDate": "2021-12-01",
        "ChargeAmount": "200.00"
      },
      {
        "ChargeDate": "2022-03-01",
        "ChargeAmount": "250.00"
      }
    ]
  }
}

```

필드에 대한 정보를 제공하여 UpdatePaymentScheduleTerms 변경 유형을 추가합니다.

- 엔터티(객체)(필수) - 재판매 승인.
 - 유형(문자열)(필수) -는 항상 Type입니다 ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다.
 - 용어(구조 배열) - 업데이트하려는 결제 조건 목록입니다. 지원되는 결제 조건은 다음과 같습니다.
 - ResalePaymentScheduleTerm(객체) - 계약 유효 기간 동안 고객에게 다른 날짜에 고정 요금이 청구되는 할부 기반 요금 모델을 정의합니다.
 - 유형(문자열) - 업데이트 중인 용어의 범주입니다.
 - CurrencyCode(문자열)(필수) - 일정에 언급된 결제에 대한 통화를 정의합니다. USD, AUD, EUR, GBP 및 JPY가 지원됩니다.
 - 일정(구조 배열) - 각 요소가 하나의 결제 분할을 정의하는 결제 일정 목록입니다. 여기에는 지불할 가격과 고객에게 요금이 청구되는 날짜를 계산하는 데 필요한 정보가 포함되어 있습니다.
 - ChargeDate(문자열)(필수) - 고객이이 결제 일정 기간에 정의된 요금을 지불하는 날짜입니다. 이 필드는 ISO 8601 형식을 지원합니다.
 - ChargeAmount(문자열)(필수) - 고객이 예정된 날짜(ChargeDate)에 지불할 가격입니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

스키마 검증은의 UpdatePaymentScheduleTerms 작업에 따라 다릅니다AWS Marketplace Catalog API. 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP
Terms.Type	필수 [x] 제품에는 지원되지 않음 허용된 조건: ResalePaymentScheduleTerm	422
용어[].CurrencyCode	필수 허용되는 값: USD	422
용어[].ResalePaymentScheduleTerm.Schedule	필수 길이는 1~60이어야 합니다.	422
용어[].ResalePaymentScheduleTerm.Shedule.ChargeDate	필수	422

입력 필드	검증 규칙	HTTP
	ISO 8601 형식이어야 합니다. 날짜는 미래여야 합니다.	
용어[].ResalePaymentScheduleTerm.Schedule.ChargeAmount	필수 음수가 아니어야 합니다.	422
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 UpdatePaymentScheduleTerms 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 또는를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_TERMS	OffersMaxQuantity 및 AvailabilityEndDate는 ResalePaymentScheduleTerm에 있어야 합니다.
TOO_MANY_SCHEDULED_결제	ResalePaymentScheduleTerm에서 최대 60개의 예약된 결제를 제공합니다.
DUPLICATE_™_DATES	ResalePaymentScheduleTerm에 고유한 청구 날짜를 입력합니다.
INVALID_™_DATES	향후 ChargeDate를 제공합니다.
INVALID_™_DATES	[x] 이전의 마지막 청구 날짜를 입력합니다.
MISSING_MANDATORY_TERMS	ResaleFixedUpfrontPricingTerm과 ResalePaymentScheduleTerm을 함께 제공합니다.

오류 코드	오류 메시지
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.
INVALID_CURRENCY_CODE	지원되는 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
호환되지 않는_통화_코드	제안이 릴리스된 후에는 CurrencyCode를 변경할 수 없습니다.

재판매 권한 부여 세부 정보 업데이트

카탈로그 API를 사용하여에서 재판매 권한 부여 세부 정보를 업데이트할 수 있습니다AWS Marketplace.

재판매 권한 부여 세부 정보를 업데이트하려면 다음 예제와 같이 UpdateInformation 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

UpdateInformation 변경 유형은 요청에 제공된 섹션만 업데이트하며 다른 모든 정보는 변경되지 않습니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
```

```

"Catalog": "AWSMarketplace",
"ChangeSet":
[
  {
    "ChangeType": "UpdateInformation",
    "Entity":
    {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "resaleauthz-123456789"
    },
    "DetailsDocument":
    {
      "Name": "TestResaleAuthorization",
      "Description": "Worldwide ResaleAuthorization for Test Product",
      "PreExistingBuyerAgreement":
      {
        "AcquisitionChannel": "AwsMarketplace",
        "PricingModel": "Contract"
      }
    }
  }
]
}

```

UpdateInformation 변경 유형을 추가할 필드에 대한 정보를 제공합니다.

- 엔터티(객체)(필수) - 재판매 권한 부여.
 - 유형(문자열)(필수) -는 항상 Type입니다ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 재판매 권한 부여에 대한 정보를 업데이트하려는 정보를 포함하여 요청의 세부 정보입니다.
 - 이름(문자열)(선택 사항) - 사용자와 채널 파트너의 가독성을 높이기 위해 ResaleAuthorization과 연결된 이름입니다.
 - 설명(문자열)(선택 사항) - 설명은 ResaleAuthorization에 대한 세부 정보를 추가할 수 있는 자유 형식 텍스트입니다.
 - PreExistingBuyerAgreement(객체)(선택 사항) -이 제안이 동일한 기본 제품에 대해 기존 고객과 체결한 기존 계약의 갱신인지 여부를 결정합니다. 기존 계약은 내부 또는 외부에 있을 수 AWS있습니다AWS Marketplace.는 제안을 감사하고 갱신되었는지 확인할 수 있습니다. 가 제안을 확인할 수 없는 경우AWS AWS는 고객의 제안 및 권한을 취소할 수 있습니다.

- AcquisitionChannel(문자열)(필수) - 기존 구매자 계약이 외부에서 서명되었는지 AWS Marketplace 아니면 로그인되었는지 나타냅니다 AWS Marketplace.

가능한 값: External, AwsMarketplace

- PricingModel(문자열)(필수) - 기존 계약에서 사용하는 요금 모델을 나타냅니다.

가능한 값: Contract, Usage, BYOL, Free

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 UpdateInformation 작업에 따라 다릅니다 AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행 StartChangeSet 되며 입력이 다음 요구 사항을 충족하지 않으면 HTTP 오류와 함께 요청이 실패합니다.

입력 필드	검증 규칙	HTTP 코드
이름	선택 사항 null 이거나 비어 있으면 안 됩니다. 길이는 1~100 자여야 합니다. 패턴 <code>^[A-Za-z0-9]*\$</code>	422

입력 필드	검증 규칙	HTTP 코드
	특수 문자 또는 공백은 허용되지 않음	
설명	선택 사항 길이는 1~255자여야 합니다. 패턴 <code>^[A-Za-z0-9\\s]*\$</code> 특수 문자는 허용되지 않음	422
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 UpdateInformation 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 또는를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_BUYER_TARGETING	PreExistingBuyerAgreement를 사용한 ResaleAuthorization에는 하나 이상의 구매자 계정이 있어야 합니다.

재판매 권한 부여 제한

카탈로그 API를 사용하여에서 제한 규칙을 재판매 권한 부여로 설정할 수 있습니다AWS Marketplace.

채널 파트너는 더 이상 제한된 재판매 권한 부여를 사용하여 비공개 제안을 생성할 수 없습니다. 기존 비공개 제안은 영향을 받지 않습니다.

재판매 승인을 제한하려면 다음 예제와 같이 RestrictResaleAuthorization 변경 유형으로 StartChangeSet API 작업을 호출합니다.

⚠ Important

이는 되돌릴 수 없는 작업입니다. 재판매 승인이 로 표시된 후에는 다시 Active 상태가 될 Restricted 수 없습니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

필드에 대한 정보를 제공하여 RestrictResaleAuthorization 변경 유형을 추가합니다.

- 개체(객체)(필수) - 재판매 승인.
 - 유형(문자열)(필수) -는 항상 Type입니다ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다. 의 빈 객체여야 합니다 RestrictResaleAuthorization.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
```



```

"ChangeSetId": "example123456789012abcdef",
"ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

스키마 검증은 AWS Marketplace Catalog API의 RestrictResaleAuthorization 작업에 따라 다릅니다. 이러한 검증은 호출할 때 수행됩니다. StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
DetailsDocument	비어 있어야 합니다.	422
RestrictResaleAuthorization	만료된 ResaleAuthorization은 로 표시할 수 없습니다. Restricted	422
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 RestrictResaleAuthorization 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 또는를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요. [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_STATUS	만료된 ResaleAuthorization은 제한으로 표시할 수 없습니다.

재판매 권한 부여를 해제하고 채널 파트너에게 표시

카탈로그 API를 사용하여 Active 상태로 ResaleAuthorization를 시작할 수 있습니다.

ReleaseResaleAuthorization는 채널 파트너가 재판매 승인을 사용하여 비공개 제안을 생성할 수 있도록 재판매 승인을 활성화합니다.

재판매 권한 부여를 해제하려면 다음 예제와 같이 ReleaseResaleAuthorization 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity":
      {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

필드에 대한 정보를 제공하여 ReleaseResaleAuthorization 변경 유형을 추가합니다.

- 엔터티(객체)(필수) - 재판매 권한 부여.
 - 유형(문자열)(필수) -는 항상 Type입니다 ResaleAuthorization@1.0.
 - 식별자(문자열)(필수) - 재판매 권한 부여 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청의 세부 정보입니다. 의 경우 비어 있어야 합니다 ReleaseResaleAuthorization.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

스키마 검증은의 ReleaseResaleAuthorization 작업에 따라 다릅니다AWS Marketplace Catalog API. 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
알 수 없는 속성	추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는의 ReleaseResaleAuthorization 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [요변경 세트 작업](#).

오류 코드	오류 메시지
MISSING_MANDATORY_TERMS	BuyerLegalTerm을 제공합니다.
MISSING_MANDATORY_TERMS	PricingTerm 제공합니다.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	제한적 또는 공개 상태의 활성 제품을 사용합니다.
INCOMPATIBLE_PRICING_TERM	PaymentScheduleTerm과 FixedUpfrontPricingTerm이 함께 있어야 합니다.
INCOMPATIBLE_BUYER_TARGETING	PreExistingBuyerAgreement를 사용한 ResaleAuthorization에는 하나 이상의 구매자 계정이 있어야 합니다.
MISSING_MANDATORY_TERMS	[x,y,z] 중 하나 이상을 제공합니다.
INCOMPATIBLE_STATUS	[x] 재판매 권한이 릴리스된 후에는 요청을 수행할 수 없습니다.

기존 재판매 권한 부여 설명

재판매 권한 부여 세부 정보를 설명하려면 다음 예제와 같이 ResaleAuthorization@1.0 엔터티 유형으로 DescribeEntity API 작업을 호출합니다.

요청 구문

```
GET /DescribeEntity?catalog=<Catalog>&entityId=<EntityId> HTTP/1.1
```

필드에 대한 정보를 제공하여 DescribeEntity 변경 유형을 추가합니다.

- 카탈로그(문자열) - 요청과 관련된 카탈로그입니다. 고정 값: AWSMarketplace.
- entityId(문자열) - 설명할의 고유 ID ResaleAuthorization입니다.

응답 구문

이 요청에 대한 응답은 제안 세부 정보를 제공하며 다음과 같습니다.

```
{
  "EntityType": "ResaleAuthorization@1.0",
  "EntityIdentifier": "resaleauthz-123456789",
```

```

"EntityArn": "arn:aws:aws-marketplace:us-east-1:111122223333:AWSMarketplace/
ResaleAuthorization/resaleauthz-123456789",
"LastModifiedDate": "2021-03-10T21:57:16Z",
"DetailsDocument": {
  "Name": "TestResaleAuthorization",
  "Description": "ResaleAuthorization for Test Product",
  "ProductId": "prod-ad8EXAMPLE51",
  "ProductName": "TestProduct",
  "Status": "Active", /*Draft, Active, Restricted*/
  "PreExistingBuyerAgreement": {
    "AcquisitionChannel": "Unknown",
    "PricingModel": "Unknown"
  },
  "CreatedDate": "2023-07-18T16:39:31.335Z",
  "ManufacturerLegalName": "ChannelCAPI.Inc",
  "ManufacturerAccountId": "123456789012",
  "Dimensions": [
    {
      "Name": "Protected Resources",
      "Description": "Additional 100 protected resources",
      "Key": "hundredresources",
      "Unit": "Units",
      "Types": [
        "Entitled"
      ]
    }
  ],
  "OfferDetails": {
    "OfferExtendedStatus": "Not Started", /* Not Started, Completed-Used, Completed-
Usable*/
    "OfferCreatedCount": 0
  },
  "Terms": [
    {
      "Type": "ResaleUsageBasedPricingTerm",
      "Id": "term_id_placeholder",
      "CurrencyCode": "USD",
      "RateCards": [
        {
          "RateCard": [
            {
              "DimensionKey": "resource_number",
              "Price": "0.05"
            }
          ]
        }
      ]
    }
  ],

```

```

        {
            "DimensionKey": "scanned_data",
            "Price": "0.05"
        }
    ]
}
],
{
    "Type": "ResaleConfigurableUpfrontPricingTerm",
    "Id": "term_id_placeholder",
    "CurrencyCode": "USD",
    "RateCards": [
        {
            "Selector": {
                "Type": "Duration",
                "Value": "P24M"
            },
            "RateCard": [
                {
                    "DimensionKey": "hundredresources",
                    "Price": "0.04"
                },
                {
                    "DimensionKey": "tenTBData",
                    "Price": "0.03"
                },
                {
                    "DimensionKey": "channel_custom",
                    "Price": "0.02"
                }
            ],
            "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
            }
        }
    ]
},
{
    "Type": "ResaleFixedUpfrontPricingTerm",
    "Id": "term-sdh27fb2",
    "CurrencyCode": "USD",
    "Duration": "P180D",

```

```

    "Price": "0.0",
    "Grants": [
      {
        "DimensionKey": "sdf73rbns93nl120d10xm1",
        "MaxQuantity": 1
      }
    ]
  },
  {
    "Type": "ResalePaymentScheduleTerm",
    "Id": "term-sdh27fb2",
    "CurrencyCode": "USD",
    "Schedule": [
      {
        "ChargeDate": "2018-07-01T00:00:00.000Z",
        "ChargeAmount": "200.00"
      },
      {
        "ChargeDate": "2019-05-01T00:00:00.000Z",
        "ChargeAmount": "200.00"
      }
    ]
  },
  {
    "Type": "BuyerLegalTerm",
    "Id": "term_id_placeholder",
    "Documents": [
      {
        "Type": "StandardEula",
        "Url": "https://resale-auth-legal-terms-iad-beta.s3.us-east-1.amazonaws.com/09ae57d6-c75a-3a4c-aadf-9b866bae64ab/a85cace8-6d9d-40ca-a053-78fc265479bf?isSigned=yes"
      }
    ]
  },
  {
    "Type": "ResaleLegalTerm",
    "Id": "term_id_placeholder",
    "Documents": [
      {
        "Type": "StandardResellerContract",
        "Url": "https://resale-auth-legal-terms-iad-beta.s3.us-east-1.amazonaws.com/09ae57d6-c75a-3a4c-aadf-9b866bae64ab/bed55b56-7ab4-4c4c-b633-3bf4f6efcb98?isSigned=yes"
      }
    ]
  }
}

```

```

    }
  ]
},
{
  "Type": "BuyerValidityTerm",
  "Id": "term_id_placeholder",
  "MaximumAgreementStartDate": "2023-09-25T23:59:59.000Z"
},
{
  "Type": "BuyerTargetingTerm",
  "Id": "term_id_placeholder",
  "PositiveTargeting": {
    "BuyerAccounts": [
      {
        "AwsAccountId": "444455556666"
      }
    ]
  }
}
],
"Rules": [
  {
    "Type": "AvailabilityRule",
    "Id": "availability_rule_id_placeholder",
    /* If the AvailabilityEndDate and OffersMaxQuantity not present Usage will be
Unlimited*/
    "Usage": "Limited",
    "AvailabilityEndDate": "2022-05-31T23:59:59Z",
    "OffersMaxQuantity": 1
  },
  {
    "Type": "PartnerTargetingRule",
    "Id": "partner_targeting_rule_id_placeholder",
    "ResellerAccountId": "777777777777",
    "ResellerLegalName": "ChannelCAPICP.Inc"
  }
]
}
}

```

다음은 DescribeEntity 응답에 표시되는 필드에 대한 정보입니다.

- EntityType(문자열) - 엔터티의 명명된 유형으로, ResaleAuthorization@1.0입니다.

- **EntityIdentifier(문자열)** - EntityId@RevisionId 형식의 개체 식별자입니다.
- **EntityArn(문자열)** - 이 요청에서 참조되는 변경 세트의 고유 식별자와 연결된 ARN입니다.
- **LastModifiedDate(문자열)** - 개체의 마지막 수정 날짜로, ISO 8601 형식(2018-02-27T13:45:22Z)입니다.
- **DetailsDocument(객체)(필수)** - 이 JSON 문자열에는 개체의 세부 정보가 포함됩니다.
 - **이름(문자열)** - 사용자와 채널 파트너의 가독성을 높이기 위해 ResaleAuthorization과 연결된 이름입니다. 계약 정보의 일부로 표시됩니다.
 - **설명(문자열)** - 설명은 본인만 사용할 수 있으며 구매자에게 절대 노출되지 않는 자유 형식 텍스트입니다.
 - **ProductId(문자열)** - 설명은 본인만 사용할 수 있으며 구매자에게 절대 노출되지 않는 자유 형식 텍스트입니다.
 - **AgreementToken(문자열)** - ResaleAuthorization의 콘텐츠에서 생성됩니다. 여기에는 계약을 생성하는 동안 용어, 규칙 및 제안자에 대한 정보가 포함되어 있습니다. 조달 중 권한 부여 확인 및 검증에 사용됩니다.
 - **용어(구조 배열)** - 수락을 위해 제시된 용어 목록입니다.
 - **규칙(구조 배열)** - 규칙 또는 지침 세트의 목록입니다.

AWS Marketplace APIs를 사용하여 채널 파트너 비공개 제안 작업

를 사용하여 채널 파트너 비공개 제안(CPPO) 작업을 자동화AWS Marketplace Catalog API할 수 있습니다.CPPOs

CPPO를 생성하거나 업데이트할 때 제안 초안에는 재판매 승인의 약관과 규칙이 포함되며 구매자가 볼 수 없습니다. 조직에 여러 페르소나를 참여시켜 비공개 제안을 생성할 수 있지만 필수는 아닙니다.

예를 들어 한 페르소나는 가격 업데이트를 담당하고 다른 페르소나는 결제 일정 업데이트를 담당할 수 있습니다. 그런 다음 세 번째 페르소나가 법률 용어를 업데이트할 책임이 있습니다. 페르소나에게 제안의 특정 부분을 업데이트할 수 있는 권한을 부여할 수 있습니다. 그러나 재판매 권한 부여에는 읽기 권한만 부여할 수 있습니다.

변경 유형을 호출하기 위한 사전 조건으로 하나 이상의 재판매 승인을 받고 작업을 숙지해야 합니다 AWS Marketplace Catalog API.

자세한 내용은 AWS Marketplace판매자 안내서의 [채널 파트너 비공개 제안을](#) 참조하세요.

다음 주제에서는 카탈로그 API를 사용하여 CPPOs를 생성하고 업데이트하는 방법을 설명합니다.

주제

- [CPPO 사전 조건](#)
- [CPPO 생성](#)
- [채널 파트너 비공개 대체 제안 생성](#)
- [마크업 업데이트](#)
- [대상 지정 구성 업데이트](#)
- [법률 리소스 업데이트](#)
- [CPPO의 검색 가능성 업데이트](#)
- [계약의 만료 날짜 정의](#)
- [요금 업데이트](#)
- [결제 일정 세부 정보 업데이트](#)
- [CPPO 게시](#)
- [기존 CPPO 정의](#)

CPPO 사전 조건

ResaleAuthorization(SLR) 설정을 위한 서비스 연결 역할은 재판매 권한 부여를 사용하여 CPPO를 생성하기 위한 필수 사전 조건입니다. 재판매 승인을 사용하려면 독립 소프트웨어 공급업체(ISVs)와 AWS Marketplace 채널 파트너 모두에 리소스 공유 권한을 제공하는 서비스 연결 역할을 생성해야 합니다. AWS, 두 그룹 모두가 사전 조건을 수행하지 않는 경우 AWS는 ISV의 권한 부여 리소스를 AWS Marketplace 채널 파트너와 공유할 수 없습니다. 자세한 내용은 AWS Marketplace 판매자 안내서의 [에 대한 재판매 권한 부여를 위한 역할 사용](#)을 AWS Marketplace 참조하세요.

CPPO 생성

사용자를 대상으로 하는 재판매 권한 부여를 사용하여의 Draft 상태에서 채널 파트너 비공개 제안 (CPPO)을 생성합니다. AWS Marketplace.

요청이 성공적으로 처리되면 재판매 승인 조건을 사용하여 Draft 상태의 제안을 AWS Marketplace Catalog API 생성합니다. DescribeEntity를 사용하여 재판매 권한 부여의 초안 제안에 적용된 조건을 볼 수 있습니다. 이는 불완전한 제안이며에서 구매자에게 표시되지 않습니다. AWS Marketplace. 그런 다음 CPPO와 연결된 변경 유형을 사용하여 제안을 완료합니다.

제안이 완료되면 ReleaseOffer 변경 유형을 사용하여 제안 생성 프로세스를 완료하고 제안을 릴리스합니다. 이렇게 하면 전체 제안이 검증되고 구매자가 제안을 볼 수 있습니다. AWS Marketplace.

채널 파트너 비공개 제안을 생성하려면 다음 예제와 같이 CreateOfferUsingResaleAuthorization 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "resaleauthz-123456789",
        "Name": "Test Offer",
        "OfferSetId": "offerset-b3f9EXAMPLE27"
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 CreateOfferUsingResaleAuthorization 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - ResaleAuthorizationId(문자열)(필수) - 제공되는 제품, 조건 및 규칙을 포함하는 고유 식별자입니다. 채널 파트너는 업데이트 변경 유형을 사용하여 추가 용어 및 규칙을 추가할 수 있습니다. ResaleAuthorization을 사용할 수 있어야 하며 파트너로 지정해야 합니다.
 - 이름(문자열)(선택 사항) - 가독성을 높이기 위해 제안과 연결된 이름입니다. 계약 정보의 일부로 표시됩니다.
 - OfferSetId(문자열)(선택 사항) -이 제안을 연결할 제안 세트의 ID입니다. 제안 세트의 일부가 될 제안을 생성할 때만이 필드를 지정합니다. OfferSetId를 제공하지 않으면 독립 실행형으로 구매할 수 있는 개별 제안이 생성됩니다. 단, 제안 생성 중에 OfferSetId를 지정하는 것은 제안을 해당 제안 세

트와 연결하려는 의도만 나타냅니다. 연결을 완료하려면 제안이 생성된 후 [AssociateOffers 변경 유형을 사용해야](#) 합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다.

를 통해 AWS Marketplace Management Portal 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

이 요청에 대한 응답은 요청 상태를 제공합니다. 상태가 이면 SUCCEEDED 새 OfferId이 생성됩니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef",
  "ChangeSetName": "Submitted by 123456789012",
  "StartTime": "2021-05-27T22:21:26Z",
  "EndTime": "2021-05-27T22:32:19Z",
  "Status": "SUCCEEDED",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "resaleauthz-123456789",
        "Name": "Test Offer"
      },
      "ErrorDetailList": []
    }
  ]
}
```

```
]
}
```

GET DescribeEntity 요청을 사용하여 AWS Marketplace Catalog API 참조에서 생성된 제안 규칙 및 조건 초안 ResaleAuthorization을 설명할 수 있습니다. 자세한 내용은 [DescribeChangeSet](#) 단원을 참조하십시오.

ResaleAuthorization의 용어 및 규칙

- LegalTerms - ResaleAuthorization의 EULA와 같은 법적 조건을 설명하는 조항이 초안 제안에 추가됩니다. UpdateLegalTerms 변경 유형을 사용하여 법률 용어를 추가할 수 있습니다. 자세한 내용은 [UpdateLegalTerms](#) 단원을 참조하십시오.
- PricingTerms - ResaleAuthorization에서 제조업체가 설명하는 모든 요금 조건(ConfigurableUpfrontPricingTerm, FixedUpfrontPricingTerm, UsageBasedPricingTerm, PaymentScheduleTerms)이 초안 제안에 추가됩니다. UpdateMarkup 변경 유형을 사용하여 대상 구매자의 요금(각 차원에 대해)을 늘리도록 선택할 수 있습니다. 자세한 내용은 이 안내서의 [UpdateMarkup](#) 섹션을 참조하세요.
- PaymentTerms - 제조업체가 ResaleAuthorization에서 향후 결제 일정을 정의한 경우 초안 제안에서 결제 조건을 볼 수 있습니다. UpdateMarkup 변경 유형을 사용하여 대상 구매자의 결제 일정 금액을 늘리도록 선택할 수 있습니다. 구매자의 결제 일정을 설정하려면 사용할 수 있습니다 UpdatePaymentScheduleTerms. 자세한 내용은 [UpdatePaymentScheduleTerms](#) 단원을 참조하십시오.
- TargetingRule - ResaleAuthorization이 특정 구매자를 대상으로 하는 경우 채널 파트너는 PositiveTargeting을 사용하여 일부 구매자에게 비공개 제안을 제공할 수 있습니다. 기본적으로 이 규칙은 ResaleAuthorization의 모든 구매자를 포함합니다. UpdateTargeting 변경 유형을 사용하여 특정 구매자를 선택하고 초안 제안을 업데이트할 수 있습니다. 자세한 내용은 [UpdateTargeting](#) 단원을 참조하십시오.

```
{
  "EntityType": "Offer@1.0",
  "EntityIdentifier": "offer-a5EXAMPLEwzpu@1",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:444555666777:AWSMarketplace/Offer/offer-a5oEXAMPLEzpu",
  "LastModifiedDate": "2021-03-10T21:57:16Z",
  "DetailsDocument": {
    "Id": "offer-3rb23tu92rn",
    "Name": "Test Offer",
    "Description": "Worldwide private offer for Test Product",
```

```
"ProductId": "prod-0bc848d78b51",
"ResaleAuthorizationId": "resaleauthz-123456789",
"Terms": [
  {
    "Type": "LegalTerm",
    "Documents": [
      {
        "Type": "CustomEula",
        "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
      }
    ]
  },
  {
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
      {
        "Selector": {
          "Type": "Duration",
          "Value": "P12M"
        },
        "RateCard": [
          {
            "DimensionKey": "m3.large",
            "Price": "300.00"
          },
          {
            "DimensionKey": "m4.xlarge",
            "Price": "400.00"
          }
        ]
      },
      {
        "Constraints": {
          "MultipleDimensionSelection": "Allowed",
          "QuantityConfiguration": "Allowed"
        }
      }
    ]
  },
  {
    "Type": "UsageBasedPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
      {
        "RateCard": [
```

```

        {
            "DimensionKey": "m3.large",
            "Price": "0.10"
        },
        {
            "DimensionKey": "m4.xlarge",
            "Price": "0.20"
        }
    ]
}
],
{
    "Type": "PaymentScheduleTerm",
    "CurrencyCode": "USD",
    "Schedule": [
        {
            "ChargeDate": "2020-12-01T00:00:00.000Z",
            "ChargeAmount": "1000.00"
        },
        {
            "ChargeDate": "2021-06-15T00:00:00.000Z",
            "ChargeAmount": "1250.00"
        }
    ]
}
],
"Rules": [
    {
        "Type": "TargetingRule",
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111222333444"
            ]
        }
    }
]
}
}

```

동기 검증

다음 스키마 검증은의 CreateOfferUsingResaleAuthorization 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
이름	필수 길이는 1~150자여야 합니다.	422
설명	필수 길이는 1~255자여야 합니다.	422
ResaleAuthorizationId	필수 길이는 1~50자여야 합니다.	422
ResaleAuthorizationId	ResaleAuthorization은 채널 파트너를 대상으로 해야 합니다.	422
ResaleAuthorizationId	ResaleAuthorization이 활성화되어 있어야 합니다.	422
채널 파트너	채널 파트너는에서 유료 판매자여야 합니다.AWS Marketplace	422

비동기 오류

다음 오류는의 CreateOfferUsingResaleAuthorization 작업에만 해당됩니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_RESALE_AUTHORIZATION	활성 상태에서 ResaleAuthorization을 사용합니다.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	제품 제안에서는 OfferSetId가 지원되지 않습니다.

채널 파트너 비공개 대체 제안 생성

카탈로그 API를 사용하여에서 채널 파트너 비공개 대체 제안을 생성할 수 있습니다AWS Marketplace.

다음 예와 같이 사용자 ResaleAuthorization 대상과 제안자인 계약을 사용하여 CreateReplacementOfferUsingResaleAuthorization 변경 유형StartChangeSet으로 호출하여 카탈로그 API에서 Draft 상태의 채널 파트너 비공개 대체 제안을 생성합니다. 대체 제안은 종료되기 전에 이전 제안의 계약을 대체하는 데 사용할 수 있습니다.

CreateReplacementOfferUsingResaleAuthorization는 대상 지정 시 계약 수락자와 함께 초안 제안을 생성합니다. 이 대상 지정은 나중에 변경할 수 없습니다. 초안 제안에는 계약의 소스 제안 ID도 포함됩니다.

채널 파트너 비공개 대체 제안을 생성하려면 다음 예제와 같이 CreateReplacementOfferUsingResaleAuthorization 변경 유형을 사용하여 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateReplacementOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "2bd2c761-3b7f-3771-a9a7-e8ad36517698",
        "Name": "CAPI-saas-abo-contract-fps",
        "AgreementId": "agmt-f2ooEXAMLEamtm7mjj0j59gu"
      }
    }
  ]
}
```

```

    }
  ]
}
```

필드에 대한 정보를 제공하여 CreateReplacementOfferUsingResaleAuthorization 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - ResaleAuthorizationId(문자열)(필수) - 제공되는 제품, 약관 및 규칙을 포함하는 ResaleAuthorizationId 고유 식별자입니다. 채널 파트너는 변경 유형 업데이트를 사용하여 추가 용어 및 규칙을 추가할 수 있습니다.

ResaleAuthorization를 사용할 수 있고 파트너로 지정해야 합니다.

- 이름(문자열)(선택 사항) - 사용자와 고객의 가독성을 높이기 위해 제안과 Name 연결됩니다. 계약 정보의 일부로도 표시됩니다.
- AgreementId(문자열)(필수) - 대상 구매자가 대체하려는 이전 제안을 수락할 때 생성된 계약의 고유 식별자AgreementId입니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 CreateReplacementOfferUsingResaleAuthorization 작업에 따라 다릅니다 AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다 StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
이름	길이는 1~150자여야 합니다.
AgreementId	제공된 계약이 활성 상태여야 함 제공된 계약이 존재해야 함 제공된 계약은 채널 파트너가 소유해야 합니다.
ResaleAuthorizationId	필수 길이는 1~50자여야 합니다.
ResaleAuthorizationId	ResaleAuthorization은 채널 파트너를 대상으로 해야 합니다.
ResaleAuthorizationId	ResaleAuthorization이 활성화되어 있어야 합니다.
채널 파트너	채널 파트너는에서 유료 판매자여야 합니다. 다.AWS Marketplace

비동기 오류

다음 오류는의 CreateReplacementOfferUsingResaleAuthorization 작업과 관련이 있습니다 AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
호환되지 않는_재판매_권한 부여	활성 상태에서 ResaleAuthorization을 사용합니다.

오류 코드	오류 메시지
호환되지 않는_재판매_권한 부여	ResaleAuthorization은 계약과 연결된 동일한 제품에 대한 것이어야 합니다.
호환되지 않는_재판매_권한 부여	계약 수락자를 대상으로 하는 ResaleAuthorization을 사용합니다.

마크업 업데이트

카탈로그 API를 사용하여서 제안의 백분율 값별로 요금 조건을 업데이트할 수 있습니다AWS Marketplace.

이렇게 하면 제조업체가 ResaleAuthorization에 정의한 모든 요금 조건 및 결제 조건(향후 결제 일정의 경우)에 지정된 백분율 마크업이 적용됩니다. 기존 마크업은 덮어씁니다. 를 사용하여 업데이트된 요금 및 결제 조건을 볼 수 있습니다DescribeEntity.

마크업을 업데이트하려면 다음 예제와 같이 UpdateMarkup 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateMarkup",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Percentage": "5.0"
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateMarkup 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - 백분율(문자열)(필수) - 백분율 값이 제조업체 요금 또는 결제 조건에 추가됩니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace판매자 운영 팀과 정보를 검증하여 지침을 충족하는AWS Marketplace지 확인하는 것이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간이 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdateMarkup 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
백분율	필수 데이터 형식은 "문자열"입니다. 음수가 아니어야 합니다.

입력 필드	검증 규칙
	최대 9개의 소수점 허용

비동기 오류

다음 오류는 UpdateMarkup 작업에만 해당됩니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_MARKUP	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
호환되지 않는_재판매_권한 부여	MarkupPercentage PaymentScheduleTerm 또는 FixedUpfrontPricingTerm 가 제안에 있고에 없는 경우를 업데이트할 수 없습니다. ResaleAuthorization .
INCOMPATIBLE_TERMS	특정 결제 금액으로 UpdatePaymentScheduleTerms를 사용하거나 예약된 날짜에 대해 단일 마크업 백분율로 UpdateMarkup을 사용합니다.
INVALID_MARKUP_PERCENTAGE	UpdateMarkup은 ResaleAuthorization을 사용하여 생성된 제안에 대해서만 호출할 수 있습니다.

대상 지정 구성 업데이트

카탈로그 API를 사용하여 기존 대상 지정 구성을 완전히 바꿀 수 있습니다. AWS Marketplace.

최신 요청에 포함되지 않은 기존 대상 지정 옵션은 제안에서 제거됩니다. 제조업체는에서 특정 대상 구매자를 언급할 수 있습니다. ResaleAuthorization. 채널 파트너는 UpdateTargeting 변경 유형 PositiveTargeting에서를 사용하여 일부 구매자에게 비공개 제안을 제공할 수 있습니다.

제안의 대상 지정 구성을 업데이트하려면 다음 예제와 같이 UpdateTargeting 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "CA"
          ],
          "BuyerAccounts": [
            "111222333444"
          ]
        },
        "NegativeTargeting": {
          "CountryCodes": [
            "XX"
          ]
        }
      }
    }
  ]
}

```

필드에 대한 정보를 제공하여 UpdateTargeting 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.

- **PositiveTargeting(객체)(선택 사항)** - 긍정적인 타겟팅은 구매자의 프로필이 제안에 액세스할 수 있도록 허용하기 위해 충족해야 하는 기준을 정의합니다. 이 필드는 선택 사항이지만이 필드가 있는 경우 하나 이상의 대상 지정 옵션을 제공해야 합니다.
- **CountryCodes(문자열 배열)(선택 사항)** - 국가에 따라 대상 지정을 허용하는 옵션으로 나열합니다. 국가에 대한 제안을 대상으로 하지 않으려는 경우이 필드를 생략해야 합니다. 있는 경우 목록에 하나 이상의 국가 코드가 포함되어야 합니다. 이 목록의 각 요소는 ISO 3166-1 alpha-2 형식을 사용하는 유효한 2자 국가 코드여야 합니다.
- **BuyerAccounts(문자열 배열)(선택 사항)** -를 기반으로 대상 지정을 허용하는 옵션으로 나열합니다. AWS 계정(프라이빗 제안이라고도 함). 에 대한 제안을 대상으로 지정하지 않으려는 경우 AWS 계정이 필드를 생략해야 합니다.
- **NegativeTargeting(객체)(선택 사항)** - 부정 타겟팅은 제안에 액세스하기 위해 고객의 프로필이 충족해야 하는 기준을 정의합니다. 이 필드는 선택 사항이지만이 필드가 있는 경우 하나 이상의 대상 지정 옵션을 제공해야 합니다.
- **CountryCodes(문자열 배열)(필수)** - 국가에 따라 대상 지정을 허용하는 옵션으로 나열합니다. 특정 국가를 대상으로 하지 않으려는 경우이 필드를 생략해야 합니다. 있는 경우 목록에 하나 이상의 국가 코드가 포함되어야 합니다. 이 목록의 각 요소는 ISO 3166-1 alpha-2 형식을 사용하는 유효한 2자 국가 코드여야 합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 `ChangeSetArn` 대한 `ChangeSetId` 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdateTargeting 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
세부 정보	필수	422
PositiveTargeting	선택 사항	422
NegativeTargeting	선택 사항	422
PositiveTargeting.CountryCodes	선택 사항 국가 코드는 유효해야 합니다(ISO 3166-1 alpha-2).	422
PositiveTargeting.BuyerAccounts	선택 사항 AWS 계정IDs 유효한 형식(12자리 숫자)이어야 합니다. 25개 이상의 계정을 포함할 수 없습니다.	422
NegativeTargeting.CountryCodes	선택 사항 국가 코드는 유효해야 합니다(ISO 3166-1 alpha-2).	422
NegativeTargeting.BuyerAccounts	제공해서는 안 됩니다(BuyerAccounts의 부정 타겟팅은 지원되지 않음).	422

비동기 오류

다음 오류는의 UpdateTargeting 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#)[변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_BUYER_ACCOUNTS	유효한 구매자 계정을 제공합니다. 잘못된 계정: [x].
INVALID_COUNTRY_CODES	지원되는 국가 코드를 제공합니다.
INVALID_TARGETING	동일한 속성에서 음수 또는 양수 대상 지정을 사용합니다.
INCOMPATIBLE_PRODUCT	제품에 대해서는 국가 기반 대상 지정이 지원되지 않습니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization 부여와 호환되는 BuyerAccounts를 제공합니다.
INCOMPATIBLE_TARGETING	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TARGETING	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TARGETING	대체 제안에서는 대상 지정을 업데이트할 수 없습니다. 구매자가 제공된 AgreementId와 연결되어 있지 않은 경우 구매자와 연결된 AgreementId를 제공하여 새 비공개 제안을 생성합니다.
TOO_MANY_BUYER_ACCOUNTS	허용된 한도 내에서 BuyerAccounts를 제공합니다.

법률 리소스 업데이트

카탈로그 API를 사용하여 재판매 권한 부여 법적 조건을 병합하고 기존 법적 조건을 완전히 바꿀 수 있습니다AWS Marketplace.

이 변경 사항은 기존 계약에 영향을 주지 않습니다. 최신 요청에 포함되지 않은 법률 약관은 제안에서 제거됩니다. 를 호출하여 병합된 법률 용어를 볼 수 있습니다DescribeEntity.

제안의 법적 조건을 업데이트하려면 다음 예제와 같이 UpdateLegalTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateLegalTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.

- 용어(구조 배열)(필수) - 업데이트하려는 법적 용어 목록입니다. 지원되는 법률 용어는 다음과 같습니다.
- LegalTerm(객체)(필수) - 수락자에게 제안할 텍스트 계약 목록을 정의합니다. 이러한 계약의 한 예는 최종 사용자 라이선스 계약(EULA)입니다.
- 유형(문자열)(필수) - 업데이트 중인 용어의 범주입니다.
- 문서(구조 배열)(필수) - 구매자에게 제안할 법적 리소스에 대한 참조 목록입니다. 이러한 리소스의 한 가지 예는 최종 사용자 라이선스 계약(EULA)입니다. 각 참조는 유형과 URL로 구성됩니다.
- 유형(문자열)(필수) - 문서 유형입니다. 사용 가능한 문서 유형은 다음과 같습니다.
 - CustomEula - 판매자가 제공하는 사용자 지정 EULA입니다. 이 문서 유형에는 액세스 가능한 S3 버킷에 저장된 EULA의 URL이 필요합니다.
 - StandardEula - 표준 계약AWS Marketplace(SCMP). SCMP에 대한 자세한 내용은 AWS Marketplace판매자 안내서를 참조하세요. 이 유형은에서 관리하므로 URL을 제공하지 않습니다AWS Marketplace.
- URL(문자열)(조건부 필수) - 구매자가 읽을 수 있는 법률 문서의 URL입니다. Type가 다음 [] 중 하나일 때 필요합니다CustomEula.
- 버전(문자열)(조건부 필수) -에서 제공하는 표준 계약의 버전입니다AWS Marketplace. 이는가 Type인 경우에 필요합니다StandardEula. 사용 가능한 버전은 다음과 같습니다.
 - 2022-07-14 -에 대한 표준 계약 버전AWS Marketplace은 다음 Amazon S3 버킷에서 사용할 수 있습니다. <https://s3.amazonaws.com/aws-mp-standard-contracts/Standard-Contract-for-AWS-Marketplace-2022-07-14.pdf>

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트의 ID를 제공합니다.

응답 구문

```
{
  "ChangeSetId": "example123456789012abcdef", "ChangeSetArn": "arn:aws:aws-
  marketplace:us-east-
  1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace판매자 운영 팀과 정보를 검증하여 지침을 충족하는AWS Marketplace지 확인하는 것이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다. 를 통해 AWS Marketplace Management Portal또는 DescribeChangeSet 작업을 사용하여 카탈로그 API에서 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdateLegalTerms 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
용어	필수 목록에는 LegalTerm만 허용됩니다. 목록 크기는 10이어야 합니다.	422
용어[].LegalTerm.Documents	필수	422
용어[].LegalTerm.Documents[].유형	필수 허용된 값: • CustomEula • StandardEula	422
용어[].LegalTerm.Documents[].URL	필수이며 "Type"이 다음 중 하나인 경우 유효한 URL이어야 합니다. • CustomEula	422

비동기 오류

다음 오류는의 UpdateLegalTerms 작업에만 해당됩니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.

오류 코드	오류 메시지
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_LEGAL_DOCUMENTS	액세스 가능한 S3 버킷에 저장된 법적 문서의 URLs을 제공합니다.
INVALID_LEGAL_DOCUMENTS	새 제안에는 최신 버전의 StandardEula만 지원됩니다.
INVALID_LEGAL_DOCUMENTS	지원되는 파일 형식으로 법률 문서를 제공합니다.
INVALID_LEGAL_DOCUMENTS	지원되는 문서 유형을 사용하여 법률 문서를 제공합니다.
LIMIT_EXCEEDED_LEGAL_DOCUMENT_SIZE	허용되는 크기 제한 내에서 법률 문서를 제공합니다.

CPPO의 검색 가능성 업데이트

카탈로그 API를 사용하여에서 제안의 검색 가능성을 관리할 수 있습니다AWS Marketplace. 이 변경 유형은 기존 계약에 영향을 주지 않습니다.

향후 특정 날짜를 설정하여 제안 검색 가능성을 제한하거나 과거에 제안을 만료하도록 선택할 수 있습니다.

제안의 검색 가능성을 관리하려면 다음 예제와 같이 UpdateAvailability 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
```

```

    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "offer-123456789"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2024-05-31"
    }
  }
]
}

```

필드에 대한 정보를 제공하여 UpdateAvailability 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - AvailabilityEndDate(문자열) - 제안을 검색하고 구매할 수 있는 날짜입니다 AWS Marketplace. 향후 특정 날짜를 설정하여 가용성을 제한하거나 과거에 제안을 만료하도록 선택할 수 있습니다. 날짜는 YYYY-MM-DD 형식으로 표시됩니다. 제안은 제공된 날짜의 23:59:59.999 UTC에 만료됩니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdateAvailability 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
AvailabilityEndDate	필수 형식: "YYYY-MM-DD"	422

비동기 오류

다음 오류는의 UpdateAvailability 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_AVAILABILITY_END_DATE	AvailabilityEndDate는 공개 제안에서 지원되지 않습니다.
INVALID_AVAILABILITY_END_DATE	향후 AvailabilityEndDate를 제공합니다.
INVALID_AVAILABILITY_END_DATE	AgreementEndDate 이전의 AvailabilityEndDate를 제공합니다. AgreementEndDate
MISSING_AVAILABILITY_END_DATE	계약 종료 날짜 이전의 AvailabilityEndDate를 제공합니다.

계약의 만료 날짜 정의

카탈로그 API를 사용하여이 제안을 사용하여 생성된 계약의 만료 날짜를 정의할 수 있습니다AWS Marketplace.

이 변경 사항은 기존 계약에 영향을 주지 않습니다. 제조업체는 재판매 승인에서 최대 계약 시작일을 언급할 수 있습니다. 그러나 채널 파트너는 해당 날짜 이후의 계약 시작 날짜를 제공할 수 없습니다.

계약의 만료 날짜를 정의하려면 다음 예제와 같이 UpdateValidityTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateValidityTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "ValidityTerm",
            "AgreementDuration": "P12M",
            "AgreementStartDate": "2021-08-01",
            "AgreementEndDate": "2022-08-01"
          }
        ]
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateValidityTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - 용어(구조 배열) - 업데이트하려는 유효 용어 목록입니다. 지원되는 유효 기간은 다음과 같습니다.
 - ValidityTerm(객체) -이 제안에서 생성된 계약을 유효하게 유지할 조건을 정의합니다.
 - 유형(문자열) - 업데이트 중인 용어의 범주입니다.

- **AgreementDuration(문자열)** - 계약이 활성 상태로 유지되는 기간을 정의합니다. AgreementStartDate이 제공되지 않은 경우 계약 기간은 계약 서명 시간을 기준으로 합니다. 기간은 ISO_8601 형식으로 표시됩니다.
- **AgreementStartDate(문자열)** - 계약이 시작되는 날짜를 정의합니다. AgreementStartDate는 YYYY-MM-DD 형식으로 표시됩니다. 계약은 제공된 날짜의 00:00:00.000 UTC에 시작됩니다. AgreementStartDate이 제공되지 않은 경우 계약 시작 날짜는 계약 서명 시간을 기준으로 결정됩니다.
- **AgreementEndDate(문자열)** - 계약이 종료되는 날짜를 정의합니다. AgreementEndDate는 YYYY-MM-DD 형식으로 표시됩니다. 계약은 제공된 날짜의 23:59:59.999 UTC에 종료됩니다. AgreementEndDate이 제공되지 않은 경우 계약 종료일은 개별 조건의 유효성에 따라 결정됩니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은 UpdateValidityTerms 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP 코드
용어	필수	422

입력 필드	검증 규칙	HTTP 코드
	목록에는 "ValidityTerm"만 허용됩니다. 비어 있거나 용어를 하나만 포함해야 합니다.	
용어[].ValidityTerm	지원되는 사용 사례: 1. AgreementDuration만 있는 ValidityTerm 2. AgreementStartDate만 포함된 ValidityTerm 3. AgreementEndDate만 있는 ValidityTerm 4. AgreementStartDate 및 Agreement EndDate가 모두 포함된 ValidityTerm	422
용어[].ValidityTerm.Agreement Duration	선택 사항 ISO_8601 형식으로 표시됩니다.	422
용어[].ValidityTerm.Agreement StartDate	선택 사항 형식: "YYYY-MM-DD"	422
용어[].ValidityTerm.Agreement EndDate	선택 사항 형식: "YYYY-MM-DD"	422

비동기 오류

다음 오류는 UpdateValidityTerms 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_AGREEMENT	대체할 현재 계약이 미래 날짜가 아닌 경우 AgreementStartDate는 미래 날짜가 될 수 없습니다.
INCOMPATIBLE_AGREEMENT_END_DATE	제안이 릴리스된 후에는 AgreementEndDate를 업데이트할 수 없습니다.
INCOMPATIBLE_AGREEMENT_START_DATE	제안이 릴리스된 후에는 AgreementStartDate를 업데이트할 수 없습니다.
INCOMPATIBLE_PRODUCT	향후 AgreementStartDate는 지원되지 않습니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementStartDate와 AgreementEndDate 사이의 기간이 ResaleAuthorization 부여와 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementStartDate가 ResaleAuthorization과 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementEndDate가 ResaleAuthorization 부여와 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementStartDate와 AgreementEndDate 사이의 기간이 ResaleAuthorization 부여와 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	AgreementDuration이 ResaleAuthorization에 지정된 기간과 일치하는지 확인합니다.
INCOMPATIBLE_TERMS	공개 제안에는 ValidityTerm이 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_AGREEMENT_DURATION	[x]일 이상의 AgreementDuration을 제공합니다.

오류 코드	오류 메시지
INVALID_AGREEMENT_END_DATE	향후 AgreementEndDate를 제공합니다.
INVALID_AGREEMENT_END_DATE	[x] 이후이거나 같은 AgreementEndDate를 제공합니다.
INVALID_AGREEMENT_START_DATE	AvailabilityEndDate 이후의 Agreement StartDate를 제공합니다. AvailabilityEndDate
INVALID_AGREEMENT_START_DATE	AgreementStartDate 이전의 Agreement EndDate를 제공합니다.
INVALID_AGREEMENT_START_DATE	오늘부터 [x]년 이내에 있는 Agreement StartDate를 제공합니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementDuration과 AgreementEndDate가 모두 포함된 ValidityTerm은 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementStartDate와 AgreementDuration이 모두 포함된 ValidityTerm은 제품에 대한 제안에서 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementStartDate가 포함된 ValidityTerm은 제품에 대한 제안에서 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementStartDate만 있는 ValidityTerm은 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementEndDate는 향후 Agreement StartDate와 함께 사용하거나 대체 제안에 사용하지 않는 한 지원되지 않습니다.
INVALID_AGREEMENT_TIME_INTERVAL	차이가 [x]년 이하인 경우 AgreementStartDate 및 AgreementEndDate를 제공합니다.
MISSING_AGREEMENT_START_DATE	ConfigurableUpfrontPricingTerm과 함께 사용할 때 AgreementStartDate가 ValidityTerm에 있는지 확인합니다. ValidityTerm

요금 업데이트

카탈로그 API를 사용하여 기존 요금 조건을 완전히 바꿀 수 있습니다.

최신 요청에 포함되지 않은 요금 조건은 제안에서 제거됩니다. 채널 파트너는 이 변경 유형을 전달에만 사용할 수 있습니다 `FixedUpfrontPricingTerm`.

제안의 요금 조건을 업데이트하려면 다음 예제와 같이 `UpdatePricingTerms` 변경 유형으로 `StartChangeSet` API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "Details": {
        "PricingModel": "Contract",
        "Terms": [
          {
            "Type": "FixedUpfrontPricingTerm",
            "CurrencyCode": "USD",
            "Price": "200.00",
            "Duration": "P465D",
            "Grants": [
              {
                "DimensionKey": "Users",
                "MaxQuantity": 10
              }
            ]
          }
        ]
      }
    ]
  }
}
```

}

필드에 대한 정보를 제공하여 UpdatePricingTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- 세부 정보(문자열)(필수) - 요청의 세부 정보입니다. 이 필드는 JSON 문자열 필드입니다. 문자열에 있을 수 없는 이스케이프 문자(예: 인용 부호)를 포함하여 한 줄 문자열 필드에 대해 올바른 형식을 지정해야 합니다.
- PricingModel(문자열)(필수) - 제안에 대한 요금 모델입니다. 요금 모델의 가능한 값은 다음과 같습니다.
 - 계약 - 구매자에게 제품 사용에 대한 요금이 미리 청구되거나 유연한 결제 일정을 제공하는 계약 기반 요금 모델입니다. 구매자는 계약 이상의 추가 사용량에 대해서도 비용을 지불할 수 있습니다.
 - 약관(구조 배열)(필수) - 업데이트하려는 요금 조건 목록입니다. 지원되는 요금 조건은 다음과 같습니다.
 - FixedUpfrontPricingTerm(객체) - 고객에게 고정 선결제 금액이 청구되는 선결제 요금 모델을 정의합니다.
 - 유형(문자열)(필수) - 업데이트 중인 용어의 유형입니다.
 - CurrencyCode(문자열)(필수) -이 기간에 언급된 가격의 통화를 정의합니다. 공개 제안의 경우 USD만 지원됩니다. 비공개 제안의 경우 USD, AUD, EUR, GBP 및 JPY가 지원됩니다.
 - 가격(문자열)(필수) -이 기간이 수락되면 고객에게 청구되는 고정 금액입니다.
 - 권한 부여(구조 배열)(필수) - 계약 실행의 일부로 고정 선결제 수락자에게 부여되는 권한입니다.
 - DimensionKey(문자열)(필수) - 제품 문서에 정의된 고유한 차원 키입니다. 차원은 제품의 용량 범주를 나타내며 제품이에 나열될 때 지정됩니다AWS Marketplace.
 - MaxQuantity(정수)(선택 사항) - 구매자에게 제품의 지정된 차원에 대한 권한을 부여할 수 있는 최대 용량입니다. MaxQuantity이 제공되지 않은 경우 구매자는 지정된 차원을 무제한으로 사용할 수 있습니다.
 - 기간(문자열)(선택 사항) - 용어가 활성 상태로 유지되는 기간을 정의합니다. 이 필드는 ISO 8601 형식을 지원합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdatePricingTerms 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
PricingModel	필수 허용된 요금 모델: 계약
용어	필수 허용되는 조건: FixedUpfrontPricingTerm
용어[].FixedUpfrontPricingTerm.CurrencyCode	필수 허용되는 값: ["USD", "AUD", "EUR", "GBP", "JPN"] 허용된 요금 모델:

입력 필드	검증 규칙
	계약
용어[].FixedUpfrontPricingTerm.Price	필수 데이터 형식은 "문자열"입니다. 음수가 아니어야 합니다. 최대 6개의 십진수 지원 특수 문자가 지원되지 않음
용어[].FixedUpfrontPricingTerm.Duration	필수 예상 형식: ISO 8601 기간
용어[].FixedUpfrontPricingTerm.Grants[].DimensionKey	필수 길이는 1~60이어야 합니다.
용어[].FixedUpfrontPricingTerm.Grants[].MaxQuantity	필수

비동기 오류

다음 오류는 UpdatePricingTerms 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
DUPLICATE_DIMENSION_KEYS	Grants에 [x]의 고유한 차원 키 목록을 제공합니다.
DUPLICATE_DIMENSION_KEYS	RateCard에 [x]의 고유한 차원 키 목록을 제공합니다.

오류 코드	오류 메시지
DUPLICATE_SELECTORS	ConfigurableUpfrontPricingTerm에 고유한 선택기 목록을 제공합니다.
DUPLICATE_TERM_TYPES	용어 유형의 고유한 목록을 제공합니다.
INCOMPATIBLE_AGREEMENT	대체 제안에서 [x, y, z]라는 용어는 제거할 수 없습니다.
INCOMPATIBLE_AGREEMENT	대체 제안에 [x, y, z]라는 용어를 추가할 수 없습니다.
호환되지 않는_통화_코드	제안이 릴리스된 후에는 CurrencyCode를 변경할 수 없습니다.
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_PRODUCT	제품에는 사용 요금 모델이 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에 대해서는 계약 요금 모델이 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에 대해서는 바이올 요금 모델이 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에는 무료 요금 모델이 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	[x]는 제품에 대한 제안에서 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제공된 결제 및 요금 조건이 호환되지 않습니다.
INCOMPATIBLE_PRODUCT	[x]의 제품에서 사용 가능한 기존 차원을 사용합니다.
INCOMPATIBLE_PRODUCT	제안의 유일한 요금 기간인 FreeTrialPricingTerm은 제품에 대해 지원되지 않습니다.
INCOMPATIBLE_PRODUCT	제품에는 [x,y,z]라는 용어가 지원되지 않습니다.

오류 코드	오류 메시지
INCOMPATIBLE_PRODUCT	대체 제안은 계약 요금 모델에 대해서만 지원됩니다.
INCOMPATIBLE_PRODUCT	제품 차원과 호환되는 요금 조건(들)을 제공합니다. 호환되지 않는 요금 조건: [x,y,z].
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	사용량 요금 모델에 대해 ConfigurableUpfrontPricingTerm에서 MultipleDimensionSelection 및 QuantityConfiguration을 허용으로 설정합니다. QuantityConfiguration ConfigurableUpfrontPricingTerm
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	사용량 요금 모델의 경우 ConfigurableUpfrontPricingTerm에서 MultipleDimensionSelection 및 QuantityConfiguration을 허용되지 않음으로 설정합니다. QuantityConfiguration ConfigurableUpfrontPricingTerm
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	제안이 릴리스된 후에는 ConfigurableUpfrontPricingTerm QuantityConfiguration을 변경할 수 없습니다.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	제안이 릴리스된 후에는 ConfigurableUpfrontPricingTerm의 MultipleDimensionSelection을 변경할 수 없습니다.
INCOMPATIBLE_RATES	무료 요금 모델을 사용할 때 모든 요금 금액과 가격을 0(0)으로 설정합니다.
INCOMPATIBLE_RATES	제품에 대한 무료 평가판 제안의 경우 UsageBasedPricingTerm에는 영(0) 가격만 허용됩니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization에 지정된 것과 동일한 CurrencyCode를 제공합니다.

오류 코드	오류 메시지
INCOMPATIBLE_RESALE_AUTHORIZATION	FixedUpfrontPricingTerm의 기간이 ResaleAuthorization에 지정된 기간과 일치하는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization과 호환되는 용어(들)를 제공합니다. 호환되지 않는 용어: [x, y, z].
INCOMPATIBLE_SELECTOR_DURATION	제안이 릴리스된 후에는 ConfigurableUpfrontPricingTerm의 요율 카드에서 기간을 제거할 수 없습니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.
INCOMPATIBLE_TERMS	[x]는 [y,z]라는 용어와 함께 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 [x,y,z]라는 용어를 추가할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 다음 조건을 제거할 수 없습니다. [x,y,z].
INCOMPATIBLE_TERMS	[x]는 비공개 제안에는 지원되지 않습니다.
INCOMPATIBLE_TERMS	무제한 사용을 허용하는 FreeTrialPricingTerm에서는 다음 용어가 지원되지 않습니다. [x,y,z].
INCOMPATIBLE_TERMS	제품의 FreeTrialPricingTerm에서는 [x,y,z]라는 용어가 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안에 PaymentScheduleTerm이 포함된 경우 FixedUpfrontPricingTerm에 대해 영(0) 가격을 제공합니다.
INCOMPATIBLE_TERMS	다음 조건은 PricingModel [x,y,z]와 호환되지 않습니다.

오류 코드	오류 메시지
INCOMPATIBLE_TERMS	FixedUpfrontPricingTerm은 MarkupPercentage가 0(0)보다 큰 경우 지원되지 않습니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_AGREEMENT_DURATION	[x]~[y]개월의 기간을 입력합니다.
INVALID_AGREEMENT_DURATION	측정된 차원에 대해 지속 시간 세부 수준이 일 수준인지 확인합니다.
INVALID_CURRENCY_CODE	지원되는 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.
INVALID_DURATION	FreeTrialPricingTerm의 기간이 허용된 범위 내에 있는지 확인합니다.
INVALID_DURATION	AgreementStartDate AgreementEndDate 사이의 기간과 일치하는 FixedUpfrontPricingTerm의 기간을 제공합니다.
INVALID_DURATION	[x]~[y]개월의 기간을 입력합니다.
INVALID_DURATION	측정된 차원에 대한 지속 시간 세부 수준이 일 수준인지 확인합니다.
INVALID_GRANTS	FreeTrialPricingTerm의 모든 권한 부여에 대해 동일한 MaxQuantity를 제공합니다.
INVALID_GRANTS	FreeTrialPricingTerm에서 사용 가능한 모든 측정 차원에 대한 권한 부여를 제공합니다.

오류 코드	오류 메시지
INVALID_PRICE_CHANGE	[x]는 지난 120일 동안 가격 인상을 요청했기 때문에 [y]까지 업데이트할 수 없습니다. 이전 가격 인상 요청을 취소하거나 자세한 내용은 AWS Marketplace관리형 카탈로그 운영 팀에 문의하십시오.
INVALID_PRICE_CHANGE	[x]의 가격 인상 및 차원 추가는 동일한 요청에서 지원되지 않습니다. 먼저 차원을 추가합니다.
INVALID_PRICE_CHANGE	UsageBasedPricingTerm의 가격 상승 및 하락은 동일한 요청에서 지원되지 않습니다. 먼저 가격을 낮춥니다.
INVALID_PRICE_CHANGE	동일한 요청에서는 RecurringPaymentTerm의 가격 인상과 UsageBasedPricingTerm의 가격 하락이 지원되지 않습니다. 먼저 가격을 낮춥니다.
INVALID_PRICE_CHANGE	동일한 요청에서는 RecurringPaymentTerm의 가격 감소와 UsageBasedPricingTerm의 가격 증가가 지원되지 않습니다. 먼저 가격을 낮춥니다.
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm에 기간 [x] 동안 하나 이상의 차원 키가 누락되었습니다. 모든 기간 동안 동일한 차원 키 세트에 대한 가격을 제공합니다.
INVALID_RATE_CARD	UsageBasedPricingTerm에서 측정된 차원에 대해서만 요금 카드를 제공합니다.
INVALID_RATE_CARD	[x]에서는 속도를 제거할 수 없습니다. 기존 요금 카드의 모든 차원에 대한 가격을 제공합니다.
INVALID_RATE_CARD	[x]에서 단위가 동일한 차원을 제공합니다.

오류 코드	오류 메시지
INVALID_RATE_CARD	모든 측정 차원 또는 모든 권한이 있는 차원을 [x]로 입력합니다.
INVALID_RATE_CARD	[x]에는 권한이 있는 차원만 제공합니다.
INVALID_RATE_CARD	UsageBasedPricingTerm 기반 요금을 제공합니다.
INVALID_RATE_CARD	모든 무료 평가판 차원에 대한 사용량 기반 요금을 제공합니다.
INVALID_RATE_CARD	UsageBasedPricingTerm에서 최대 소수점 8자리까지 가격을 제공합니다.
INVALID_SELECTOR_DURATION_VALUE	[x]~[y]개월의 기간을 입력합니다.
INVALID_SELECTOR_DURATION_VALUE	측정된 차원에 대한 지속 시간 세부 수준이 일 수준인지 확인합니다.
INVALID_SELECTOR_DURATION_VALUE	ConfigurableUpfrontPricingTerm의 기간이 허용된 범위 내에 있는지 확인합니다.
INVALID_SELECTOR_DURATION_VALUE	지원되는 계약 기간을 하나 이상 제공합니다.
INVALID_SELECTOR_DURATION_VALUE	지원되는 계약 기간 또는 단일 사용자 지정 기간을 하나 이상 제공합니다.
INVALID_SELECTOR_DURATION_VALUE	AgreementStartDate AgreementEndDate 간의 기간과 일치하는 ConfigurableUpfrontPricingTerm의 기간을 제공합니다.
누락_지속 시간	FixedUpfrontPricingTerm에 기간을 입력합니다.
MISSING_MANDATORY_TERMS	FixedUpfrontPricingTerm은 ByolPricingTerm 또는 PaymentScheduleTerm과 페어링된 경우에만 지원됩니다.
MISSING_MANDATORY_TERMS	[x,y,z] 중 하나 이상을 제공합니다.

오류 코드	오류 메시지
MISSING_MANDATORY_TERMS	Byol 요금 모델을 사용할 때 ByolPricingTerm을 제공합니다.
TOO_MANY_GRANTS	[y]에 최대 [x]개의 권한 부여를 제공합니다.
TOO_MANY_RATE_CARDS	ConfigurableUpfrontPricingTerm의 요금 카드 하나만 제품에 허용됩니다.
TOO_MANY_RATE_CARDS	제품의 ConfigurableUpfrontPricingTerm에는 최대 [x]개의 요금 카드가 허용됩니다.
TOO_MANY_RATES	ConfigurableUpfrontPricingTerm의 허용된 한도 내에서 RateCards를 제공합니다.
TOO_MANY_RATES	UsageBasedPricingTerm의 허용된 한도 내에서 RateCards를 제공합니다.

결제 일정 세부 정보 업데이트

카탈로그 API를 사용하여 유연한 결제 일정과 같은 결제 일정 세부 정보를 변경할 수 있습니다AWS Marketplace.

제조업체가 재판매 승인에 결제 일정을 제공한 경우 채널 파트너는 다음 중 하나를 수행할 수 있습니다.

- UpdateMarkup 변경 유형을 사용하여 모든 결제 일정에 균일한 백분율 마크업을 적용합니다.
- UpdatePaymentScheduleTerms 변경 유형을 사용하여 사용자 지정 결제 금액을 재판매 승인에 제공된 것보다 크거나 같은 값으로 설정합니다. 결제 날짜는 변경할 수 없습니다. 두 옵션을 동시에 적용할 수는 없습니다. 마크업이 먼저 적용되고 결제 일정을 적용하기 위해 마크업을 변경해야 하는 경우 UpdateMarkup 변경 유형을 0 통해 마크업을 0 로 설정합니다. 결제 일정이 먼저 적용되는 경우 변경 사항을 되돌려 마크업을 적용할 수 없습니다.

제안에 대한 결제 관련 세부 정보를 업데이트하려면 다음 예제와 같이

UpdatePaymentScheduleTerms 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문


```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePaymentScheduleTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "PaymentScheduleTerm",
            "Schedule": [
              {
                "ChargeDate": "2021-12-01",
                "ChargeAmount": "200.00"
              },
              {
                "ChargeDate": "2022-03-01",
                "ChargeAmount": "250.00"
              }
            ]
          }
        ]
      }
    }
  ]
}

```

필드에 대한 정보를 제공하여 UpdatePaymentScheduleTerms 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다 Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다.
 - 용어(구조 배열) - 업데이트하려는 결제 조건 목록입니다. 지원되는 결제 조건은 다음과 같습니다.

- **PaymentScheduleTerm(객체)** - 계약 유효 기간 동안 고객에게 다른 날짜에 고정 요금이 청구되는 할부 기반 요금 모델을 정의합니다.
- **유형(문자열)** - 업데이트되는 용어의 유형입니다. 객체 값입니다 "PaymentScheduleTerm".
- **일정(구조 배열)** - 각 요소가 하나의 결제 분할을 정의하는 결제 일정 목록입니다. 여기에는 지불할 가격과 고객에게 요금이 청구되는 날짜를 계산하는 데 필요한 정보가 포함되어 있습니다.
- **ChargeDate(문자열)** - 고객이이 결제 일정 기간에 정의된 요금을 지불하는 날짜입니다. ChargeDate는 YYYY-MM-DD 형식으로 표시됩니다. 인보이스는 제공된 날짜에 생성됩니다.
- **ChargeAmount(문자열)** - 고객이 예약된 날짜()에 지불하는 가격입니다 ChargeDate.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 UpdatePaymentScheduleTerms 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙	HTTP
용어	필수	422

입력 필드	검증 규칙	HTTP
	"PaymentScheduleTerm"만 허용됩니다. 목록 크기는 2보다 작아야 합니다.	
용어[].PaymentScheduleTerm.CurrencyCode	필수 지원되는 통화: ["USD", "AUD", "EUR", "GBP", "JPN"]	422
용어[].PaymentScheduleTerm.Schedule[]	필수 목록 크기는 1~60이어야 합니다.	422
용어[].PaymentScheduleTerm.Schedule[].ChargeDate	필수 형식: "YYYY-MM-DD"	422
용어[].PaymentScheduleTerm.Schedule[].ChargeAmount	필수 데이터 형식은 "문자열"입니다. 최대 2개의 소수 자릿수가 지원되는 음수가 아닌 소수 자릿수 추가 속성은 허용되지 않습니다.	422

비동기 오류

다음 오류는 UpdatePaymentScheduleTerms 작업과 관련이 있습니다. AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#) [변경 세트 작업](#).

오류 코드	오류 메시지
DUPLICATE_™_DATES	PaymentScheduleTerm에 고유한 청구 날짜를 입력합니다.

오류 코드	오류 메시지
호환되지 않는_마크업_백분율	PaymentScheduleTerm은 MarkupPercentage가 0(0)보다 큰 경우 지원되지 않습니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization과 호환되는 용어(들)를 제공합니다. 호환되지 않는 용어: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	PaymentScheduleTerm의 총 ChargeAmounts가 ResaleAuthorization과 호환되는지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	PaymentScheduleTerm의 청구 날짜가 ResaleAuthorization의 청구 날짜와 동일한지 확인합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	PaymentScheduleTerm의 청구 금액이 해당 날짜의 ResaleAuthorization의 청구 금액보다 크거나 같은지 확인합니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	특정 결제 금액으로 UpdatePaymentScheduleTerms를 사용하거나 예약된 날짜에 대해 단일 마크업 백분율로 UpdateMarkup을 사용합니다.
INVALID_™_DATES	AgreementEndDate 이전의 청구 날짜를 입력합니다.
TOO_MANY_BACKDATED_CHARGES	AvailabilityEndDate 전에 최대 1개의 예약된 결제를 제공합니다.

오류 코드	오류 메시지
DUPLICATE_™_DATES	PaymentScheduleTerm에 고유한 청구 날짜를 입력합니다.
호환되지 않는_통화_코드	제안이 릴리스된 후에는 CurrencyCode를 변경할 수 없습니다.
호환되지 않는_마크업_백분율	MarkupPercentage가 0(0)보다 큰 경우 PaymentScheduleTerm이 지원되지 않습니다.
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_RESALE_AUTHORIZATION	ResaleAuthorization과 호환되는 용어(들)를 제공합니다. 호환되지 않는 용어: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	PaymentScheduleTerm의 총 ChargeAmounts가 ResaleAuthorization과 호환되는지 확인합니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.
INCOMPATIBLE_TERMS	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
INCOMPATIBLE_TERMS	제안이 만료된 후에는 요청된 변경을 수행할 수 없습니다.
INVALID_™_DATES	AgreementEndDate 이전의 청구 날짜를 입력합니다.
INVALID_CURRENCY_CODE	지원되는 CurrencyCode를 제공합니다.
INVALID_CURRENCY_CODE	모든 요금 및 결제 조건에 동일한 CurrencyCode를 제공합니다.

오류 코드	오류 메시지
TOO_MANY_BACKDATED_CHARGES	AvailabilityEndDate 전에 최대 1개의 예약된 결제를 제공합니다.

CPPO 게시

카탈로그 API를 사용하여 모든 업데이트 변경 유형에서 수집된 정보를 병합한 다음 제안을 게시할 수 있습니다AWS Marketplace.

제안ReleaseOffer은가 호출될 때까지 Draft 상태로 유지됩니다. 제안이 릴리스되면에서 검색할 수 있습니다AWS Marketplace.

제안을 게시하려면 다음 예제와 같이 ReleaseOffer 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

필드에 대한 정보를 제공하여 ReleaseOffer 변경 유형을 추가합니다.

- 개체(객체)(필수) - CPPO.
 - 유형(문자열)(필수) -는 항상 Type입니다Offer@1.0.
 - 식별자(문자열)(필수) - 제안 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.

- DetailsDocument(객체)(필수) - 요청 세부 정보의 JSON 값입니다. 의 경우 비어 있어야 합니다 ReleaseOffer.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분에서 몇 시간 정도 걸릴 수 있습니다.

를 통해 AWS Marketplace Management Portal또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

동기 검증

다음 스키마 검증은의 ReleaseOffer 작업에 따라 다릅니다AWS Marketplace Catalog API. 이러한 검증은를 호출할 때 수행됩니다StartChangeSet. 요청이 다음 요구 사항을 충족하지 않으면 HTTP 응답으로 실패합니다.

입력 필드	검증 규칙
세부 정보	비어 있어야 합니다({}).

비동기 오류

다음 오류는의 ReleaseOffer 작업과 관련이 있습니다AWS Marketplace Catalog API. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요[변경 세트 작업](#).

오류 코드	오류 메시지
호환되지 않는_결제_설정	CurrencyCode와 호환되도록 결제 설정을 업데이트합니다.
INCOMPATIBLE_PRODUCT	먼저 제품에 대한 공개 제안을 생성합니다.
INCOMPATIBLE_SELLER_VERIFICATION	필요한 모든 판매자 확인 프로세스를 완료합니다.
INVALID_UPDATE_REQUEST	제안이 릴리스된 후에는 요청된 변경을 수행할 수 없습니다.
MISSING_AGREEMENT_END_DATE	대체 제안에 대한 AgreementEndDate를 제공합니다.
MISSING_AVAILABILITY_END_DATE	비공개 제안에 대한 AvailabilityEndDate를 제공합니다.
MISSING_MANDATORY_TERMS	제안에 PaymentScheduleTerm이 포함된 경우 FixedUpfrontPricingTerm을 제공합니다. PaymentScheduleTerm
누락_구매자_계정	ResaleAuthorization을 사용하여 생성된 제안에 대해 PositiveTargeting에 BuyersAccounts를 제공합니다.
누락_구매자_계정	제품에 대한 모든 제안은 비공개여야 합니다. BuyersAccounts에 PositiveTargeting을 제공합니다.
누락_설명	제안을 릴리스하기 전에 설명을 설정합니다.
MISSING_MANDATORY_TERMS	제안에 [x]를 추가합니다.
MISSING_MANDATORY_TERMS	제안에 PaymentScheduleTerm이 포함된 경우 FixedUpfrontPricingTerm을 제공합니다. PaymentScheduleTerm

오류 코드	오류 메시지
누락_이름	제안을 릴리스하기 전에 이름을 설정합니다.
TOO_MANY_OFFERS	제품당 하나의 공개 제안만 생성할 수 있습니다.
TOO_MANY_OFFERS	제품당 하나의 공개 무료 평가판 제안만 생성할 수 있습니다.

기존 CPPO 정의

카탈로그 API를 사용하여에서 CPPO 세부 정보를 정의할 수 있습니다AWS Marketplace.

기존 CPPO를 정의하려면 다음 예제와 같이 Offer@1.0 엔티티 유형으로 DescribeEntity API 작업을 호출합니다.

요청 구문

```
GET /DescribeEntity?catalog=<Catalog>&entityId=<EntityId> HTTP/1.1
```

필드에 대한 정보를 제공하여 DescribeEntity 변경 유형을 추가합니다.

- 카탈로그(문자열) - 요청과 관련된 카탈로그입니다. 고정 값: AWSMarketplace.
- entityId(문자열) - 설명할 제안의 고유 ID입니다.

응답 구문

이 요청에 대한 응답은 제안 세부 정보를 제공하며 다음과 같습니다.

```
{
  "EntityType": "Offer@1.0",
  "EntityIdentifier": "offer-a5oEXAMPLEzpu@1",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:446235747164:AWSMarketplace/Offer/offer-a5oEXAMPLEzpu",
  "LastModifiedDate": "2021-03-10T21:57:16Z",
  "Details": {
    "Id": "offer-3rEXAMPLErn",
    "State": "Released",
    "Name": "Test Offer",
    "Description": "Worldwide private offer for Test Product",
```

```
"PreExistingAgreement": {
  "AcquisitionChannel": "External",
  "PricingModel": "Contract"
},
"ProductId": "prod-0bEXAMPLEb51",
"ResaleAuthorizationId": "resaleauthz-123456789",
"MarkupPercentage": "5.0",
"Terms": [
  {
    "Type": "LegalTerm",
    "Documents": [
      {
        "Type": "CustomEula",
        "Url": "https://s3.amazonaws.com/EULA/custom-eula-1234.txt"
      }
    ]
  },
  {
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
      {
        "Selector": {
          "Type": "Duration",
          "Value": "P12M"
        },
        "RateCard": [
          {
            "DimensionKey": "m3.large",
            "Price": "300.00"
          },
          {
            "DimensionKey": "m4.xlarge",
            "Price": "400.00"
          }
        ]
      },
      {
        "Constraints": {
          "MultipleDimensionSelection": "Allowed",
          "QuantityConfiguration": "Allowed"
        }
      }
    ]
  }
],
{
```

```

    "Type": "UsageBasedPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
      {
        "RateCard": [
          {
            "DimensionKey": "m3.large",
            "Price": "0.10"
          },
          {
            "DimensionKey": "m4.xlarge",
            "Price": "0.20"
          }
        ]
      }
    ],
  },
  {
    "Type": "PaymentScheduleTerm",
    "CurrencyCode": "USD",
    "Schedule": [
      {
        "ChargeDate": "2020-12-01T00:00:00Z",
        "ChargeAmount": "1000.00"
      },
      {
        "ChargeDate": "2021-06-15T00:00:00Z",
        "ChargeAmount": "1250.00"
      }
    ]
  }
],
"Rules": [
  {
    "Type": "TargetingRule",
    "PositiveTargeting": {
      "CountryCodes": [
        "US",
        "CA"
      ],
      "BuyerAccounts": [
        "118033953248"
      ]
    }
  },

```

```

    "NegativeTargeting": {
      "CountryCodes": [
        "XX"
      ]
    },
    {
      "Type": "AvailabilityRule",
      "AvailabilityEndDate": "2050-08-30T01:56:03Z"
    }
  ]
}

```

DescribeEntity 응답의 입력 필드에 대한 정보를 제공합니다.

- **EntityType(문자열)** - 엔터티의 명명된 유형으로, `Offer@1.0`.
- **EntityIdentifier(문자열)** - 형식의 개체 식별자입니다 `EntityId@RevisionId`.
- **EntityArn(문자열)** - 이 요청에서 참조되는 변경 세트의 고유 식별자와 연결된 ARN입니다.
- **LastModifiedDate(문자열)** - 개체의 마지막 수정 날짜로, ISO 8601 형식()입니다 `2018-02-27T13:45:22Z`.
- **DetailsDocument(객체)** - JSON 객체에는 개체의 세부 정보가 포함됩니다.
 - **Id(문자열)** - AWS Marketplace 및의 제안 엔터티에 대한 고유 식별자는 제안을 생성하는 동안 생성됩니다.
 - **상태(문자열)** - 제안의 상태입니다.
 - **이름(문자열)** - 사용자와 고객의 가독성을 높이기 위해 제안과 연결된 이름입니다. 계약 정보의 일부로도 표시됩니다.
 - **설명(문자열)** - 설명은 본인만 사용할 수 있으며 구매자에게 절대 노출되지 않는 자유 형식 텍스트입니다.
 - **PreExistingAgreement 전(문자열)** - 이 제안이 동일한 기본 제품에 대해 기존 고객과 체결한 기존 계약의 갱신인지 여부를 결정합니다. 기존 계약은 내부 또는 외부에 있을 수 있습니다 AWS Marketplace.는 제안을 감사하고 갱신되었는지 확인할 수 있습니다. AWS가 제안을 확인할 수 없는 경우 AWS는 고객의 제안 및 권한을 취소할 수 있습니다.
 - **AcquisitionChannel(문자열)** - 기존 계약이 외부 AWS Marketplace 또는 내부에서 서명되었는지 여부를 나타냅니다 AWS Marketplace. 가능한 값: `External`, `AwsMarketplace`.

- PricingModel(문자열) - 기존 계약에서 사용하는 요금 모델을 나타냅니다. 가능한 값: Contract, Usage, Byo1, Free.
- ProductId(문자열) - 설명은 본인만 사용할 수 있으며 구매자에게 절대 노출되지 않는 자유 형식 텍스트입니다.
- ResaleAuthorizationId(문자열) - ResaleAuthorization은 비공개 제안을 생성하는 데 사용됩니다.
- MarkupPercentage(문자열) - 채널 파트너가 UpdateMarkup 변경 유형에서 전달한 백분을 값입니다. 이 마크업은 이미 용어에 적용됩니다.
- 용어(구조 배열) - 용어 목록입니다.
- 규칙(구조 배열) - 규칙 목록입니다.

갱신 작업

의 갱신을 AWS Marketplace 통해 독립 소프트웨어 공급업체(ISVs) 및 채널 파트너를 포함한 판매자는 고객과 기존 계약을 원활하게 확장할 수 있습니다. 갱신 프로세스 중에 판매자는 새 제안을 생성하여 조건을 조정할 수 있습니다.

갱신에는 AWS Marketplace 네 가지 유형이 있습니다.

- 계약 기반 제안: 판매자는 기존 계약을 교체하여 요금, 기간, 조건을 조정하고 기존 계약이 종료되기 전에 갱신할 수 있습니다. 계약 기반 제안은 현재 계약 종료일 이후에 고객에게 연장하여 고객의 필요에 따라 새로운 권한, 할인 또는 결제 일정을 부여할 수 있습니다.

이 지원은 유연한 결제 옵션을 제공하는지 여부에 관계없이 계약 및 소비 기반 요금(CCP)을 사용하는 제품을 포함한 서비스형 소프트웨어(SaaS) 제품에 적용됩니다.

- 장래 날짜의 제안: 판매자는 장래 날짜의 제안을 생성하여 장래 날짜부터 시작하여 기존 약관이 진행되는 동안 갱신을 사전 예약하는 데 사용할 수 있습니다. 고객이 수락하면 계약이 지정된 미래 날짜에 시작되어 현재 계약이 종료된 후가 시작될 수 있습니다.

이 지원은 유연한 결제 옵션을 제공하는지 여부에 관계없이 계약 및 소비 기반 요금(CCP)을 사용하는 제품을 포함한 서비스형 소프트웨어(SaaS) 제품에 적용됩니다.

- 새 비공개 제안: 판매자는 기존 계약을 갱신하기 위해 언제든지 수락할 수 있는 새 약관(AMI 시간당, AMI 연간 및 SaaS pay-as-you-go제 구독의 경우)을 사용하여 [새 비공개 제안을 생성할](#) 수 있습니다.
- 자동 갱신: 고객은 공개 제안 및 비공개 제안이 있는 AWS Data Exchange 제품에 대해 자동 갱신을 활성화하여 이전 계약이 종료될 때 새 계약을 자동으로 생성할 수 있습니다.

주제

- [대체 제안](#)
- [미래 날짜의 계약](#)
- [리소스](#)

대체 제안

판매자는 고객이 공개 제안 또는 비공개 제안을 수락할 때 원래 생성된 활성 계약을 교체하여 갱신을 제안할 수 있습니다. 대체 제안을 사용하면 현재 계약 종료일을 넘어 고객에게 새 제안을 연장하거나, 새 권한을 부여하거나, 요금 할인을 제공하거나, 결제 일정을 조정하거나, 결제 일정을 변경하거나, 최종 사용자 라이선스 계약(EULA)을 변경할 수 있습니다.

카탈로그 API를 사용하여 [지원되는 제품 유형에](#) AWS Marketplace에서 [대체 제안](#)(계약 기반 제안이라고도 함)을 생성할 수 있습니다. 교체하려는 현재 계약의 고유 식별자(계약 ID)를 제공해야 합니다. 이 계약 ID는 계약 섹션 [AWS Marketplace Management Portal](#)의 또는 [SearchAgreements](#) API 작업을 사용하여 찾을 수 있습니다.

고객이 대체 제안을 수락하면 현재 계약이 이전 종료 날짜 이후에 연장될 수 있는 새 계약으로 대체됩니다.

Note

원래 계약과 다른 레코드 판매자를 지정하는 대체 계약은 생성할 수 없습니다.

대체 제안에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [AWS Marketplace에서 비공개 제안 수정](#)을 참조하세요.

미래 날짜의 계약

미래 날짜의 계약은 고객이 미래 서비스 시작일이 있는 비공개 제안을 수락할 때 생성됩니다. 에서 예정된 갱신의 사전 예약을 용이하게 하기 위해 AWS Marketplace ISVs 및 채널 파트너는 수락 즉시가 아니라 현재 계약이 종료된 다음날 시작하는 비공개 제안을 생성할 수 있습니다.

갱신 계약의 시작 날짜를 현재 계약의 종료 날짜 다음 날로 설정하려면 이미 게시된 비공개 제안에 [카탈로그 API UpdateValidityTerms 변경 유형](#)을 사용할 수 있습니다. 판매자는 향후 최대 3년까지 서비스 시작 날짜를 선택할 수 있습니다.

고객은 비공개 제안의 이용 약관을 검토하고 적용 전에 수락할 수 있습니다. 시작 날짜가 미래인 비공개 제안을 수락해도 현재 계약은 대체되지 않습니다. 대신 이전 계약이 종료된 직후 시작되는 갱신 계약을 생성합니다.

미래 날짜의 제안 및 계약을 지원하는 제품 유형에 대한 자세한 내용은 AWS Marketplace 판매자 안내서의 [미래 날짜의 계약 생성](#)을 참조하세요.

리소스

- 작업 코드 예제가 포함된 end-to-end 랩은 다음을 참조하세요.
 - [랩: 비공개 제안 생성\(향후 서비스 시작 날짜 포함\)](#)
 - [랩: 대체 비공개 제안 생성](#)
- 대체 제안 생성에 대한 동영상은 YouTube의 [SaaS 계약 비공개 제안 갱신 -AWS Marketplace](#) 섹션을 참조하세요.
- 미래 날짜의 제안 생성에 대한 동영상은 YouTube에서 [AWS Marketplace 미래 날짜의 비공개 제안 생성](#)을 참조하세요.

다중 제품 솔루션 작업

AWS Marketplace 다중 제품 솔루션은 여러 제품과 서비스를 결합하여 특정 고객 요구 사항을 해결합니다. 이를 지원하는 두 가지 기능은 다음과 같습니다.

- 솔루션 - AWS Marketplace 제품이 함께 작동하는 방식을 고객에게 보여주는 전용 페이지입니다.
- 제안 세트 - 고객이 단일 작업으로 모든 제안을 수락할 수 있도록 여러 비공개 제안을 그룹화하는 통합 트랜잭션입니다.

이러한 기능을 독립적으로 또는 함께 사용할 수 있습니다. 솔루션을 사용하여 제품 검색 개선, 조달 간소화를 위한 제안 세트 또는 완벽한 멀티 제품 경험을 위한 두 가지 모두를 제공합니다.

주제

- [솔루션 작업](#)
- [제안 세트 작업](#)

솔루션 작업

솔루션은 특정 고객 문제를 해결하기 위해 함께 작동하는 제품 및 서비스 모음입니다. 솔루션에는 AWS Marketplace 제품 및 AWS Marketplace가 아닌 제품이 포함될 수 있습니다. 솔루션은 여러 제품이 통합되어 산업별 사용 사례를 해결하는 방법을 보여주는 데 도움이 됩니다. 이 솔루션은 고객 문제, 사용 사례, 통합 세부 정보 및 관련 제품을 설명합니다.

AWS Marketplace 및 AWS Partner Central의 솔루션

Partner Central 2.0에서 Partner Central 3.0으로 마이그레이션하면 AWS 계정을 생성하고 AWS Marketplace 판매자로 등록하라는 메시지가 표시됩니다. 자세한 내용은 [판매자 프로필 등록 및 생성을 참조하세요](#). 이 등록은 다음을 위해 필요합니다.

- 기존 Partner Central 솔루션을 AWS Marketplace 카탈로그로 마이그레이션(모든 솔루션이 AWS Marketplace 카탈로그로 마이그레이션됨)
- AWS Marketplace 솔루션이 API 판매 APIs와 같은 Partner Central API와 함께 작동하도록 합니다. [APIs](#)

프로필 연결 및 마이그레이션 이해

Partner Central 계정과 AWS Marketplace 계정을 연결하는 두 가지 고유한 시나리오가 있습니다.

프로필 연결 전용 - Partner Central 3.0으로 마이그레이션하지 않고 AWS Partner Central 및 AWS 계정을 연결하는 경우 AWS Marketplace 솔루션과 Partner Central 솔루션은 별도로 유지됩니다. 각 플랫폼을 독립적으로 관리합니다.

프로필 마이그레이션 - 연결된 AWS Marketplace 판매자 프로필을 사용하여 Partner Central 3.0으로 마이그레이션하는 경우:

- Partner Central 2.0 솔루션을 AWS Marketplace로 마이그레이션
- AWS Marketplace를 통해서만 솔루션을 관리합니다.
- Partner Central APIs 마이그레이션된 솔루션에 액세스할 수 있습니다.

Note

AWS Partner Central 및 AWS 계정을 이미 연결한 경우([AWS Partner Central 및 AWS 계정 연결 참조](#)) 새 AWS 계정을 생성할 필요가 없습니다. 그러나 기존 AWS 계정이 AWS Marketplace

판매자로 등록되어 있는지 확인합니다. AWS Marketplace 구매자 웹 사이트에서 솔루션을 공개적으로 사용하려면 AWS Marketplace 판매자로서 공개 프로필이 필요합니다. AWS Marketplace 판매자 등록 지침은 [판매자 프로필 등록 및 생성을 참조하세요](#).

Partner Central APIs에서 기존 AWS Marketplace 솔루션 사용 가능

Partner Central 3.0 프로파일 마이그레이션 전에 생성한 AWS Marketplace 솔루션은 Partner Central APIs에서 자동으로 인식되지 않습니다. 이는 마이그레이션을 완료하고 AWS 계정을 Partner Central 계정과 연결한 후에도 적용됩니다.

기존 사전 마이그레이션 솔루션을 Partner Central APIs에 표시하려면 솔루션의 필드를 업데이트합니다. 현재 있는 것과 동일한 정보를 입력할 수 있습니다. Partner Central 3.0 프로파일 마이그레이션 전에 생성한 각 솔루션에 대해 이 프로세스를 반복합니다.

예를 들어 UpdateInformation 변경 유형을 사용하여 솔루션의 이름을 현재 값으로 업데이트합니다. 솔루션을 한 번 업데이트하면 Partner Central APIs는 해당 솔루션에 대한 향후 업데이트를 자동으로 인식합니다.

솔루션 개체

솔루션은 구매자가 상품을 검색하고 이해하는 데 도움이 되는 마케팅 문서입니다. 솔루션에는 제목, 설명, 사용 사례, AWS Marketplace 제품 및 AWS Marketplace가 아닌 제품이 포함되어 있습니다. 솔루션은 제품, 제안 및 제안 세트와는 별도의 엔터티 유형입니다. 각 솔루션에는 구매자가 개별 제품과 별도로 찾아볼 수 있는 자체 검색 및 나열 경험이 있습니다.

주요 특징

- AWS Marketplace 제품과 비 AWS Marketplace 제품을 모두 포함하여 여러 제품을 그룹화합니다.
- 구매자를 위한 전용 검색 및 목록 환경을 생성합니다.
- 소유한 제품을 다른 판매자의 제품과 그룹화합니다.

솔루션 개체 유형은 `Solution@1.0`입니다.

개체 식별자

솔루션은 `soln-*` 식별자 형식(예: `soln-abc123def456`)을 사용합니다.

개체 ARN 형식

솔루션은 표준 AWS ARN 형식을 따릅니다.

```
arn:{aws-partition}:aws-marketplace:{region}:{account-id}:{aws-marketplace-catalog}/
Solution/{solution-id}
```

예제:

```
arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/soln-
abc123def456
```

솔루션 시작하기

이 섹션에서는 AWS Marketplace 카탈로그 API [변경 유형을](#) 사용하여 솔루션을 생성하고 관리하는 방법에 대한 자세한 정보를 제공합니다.

시작하기 전에, 다음 사항을 확인해야 합니다.

- 유효한 AWS Marketplace 판매자 등록 및 공개 프로필. 자세한 내용은 [판매자 프로필 등록 및 생성을 참조하세요](#).
- API에 대한 액세스 및 완료된 판매자 사전 조건. 자세한 내용은 [AWS Marketplace Catalog API에 대한 액세스 제어를 참조하세요](#).
- AWS Marketplace Catalog API 사용의 기본 사항에 대해 알아보려면 [AWS Marketplace Catalog API 사용을 참조하세요](#).

솔루션 나열

계정의 모든 솔루션을 나열하려면 ListEntities API 작업을 호출하고를 솔루션EntityType으로 설정합니다.

요청

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "Solution"
}
```

응답

```
{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
Solution/soln-abc123456",
      "EntityId": "soln-abc123456",
      "EntityType": "Solution",
      "LastModifiedDate": "2024-10-10T19:50:43Z",
      "Name": "Test Solution 1",
      "Visibility": "Limited"
    }
  ],
  "NextToken": null
}
```

가시성을 기준으로 솔루션 필터링

FilterList 파라미터를 사용하여 가시성을 기준으로 솔루션을 필터링할 수 있습니다.

요청

```
POST /ListEntities HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "EntityType": "Solution",
  "FilterList": [
    {
      "Name": "Visibility",
      "ValueList": [
        "Public"
      ]
    }
  ]
}
```

솔루션 세부 정보 가져오기

특정 솔루션에 대한 자세한 정보를 가져오려면 DescribeEntity API 작업을 호출합니다.

Note

DescribeEntity 응답은 RevisionId, For 솔루션EntityId과 결합하고 솔루션 IDEntityId이며 낙관적 잠금에 사용할 RevisionId 수 EntityIdentifier 있는 반환합니다. 자세한 내용은 [식별자를 참조하세요](#).

요청

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=soln-abc123456 HTTP/1.1
```

응답

```
{
  "Details": "{...}",
  "DetailsDocument": {
    "Lifecycle": {
      "Visibility": "Limited"
    },
    "Identifiers": {
      "Id": "soln-abc123456"
    },
    "Presentation": {
      "Title": "My Solution",
      "Name": "solution_for_customer_X",
      "ShortDescription": "Solution value proposition",
      "LongDescription": "Detailed solution description",
      "LogoUrl": "https://s3.amazonaws.com/logo.jpg",
      "UseCases": [
        {
          "Group": "advertising_and_marketing",
          "SubGroup": "ad_intelligence_and_measurement",
          "Item": "amazon_ads_insights",
          "Description": "Use case description"
        }
      ]
    },
    "RelatedProducts": {
      "AwsMarketplaceProducts": [
        {
          "ProductId": "prod-123"
        }
      ]
    }
  }
}
```

```

    ],
    "IntegrationDescription": "How products work together",
    "NonAwsMarketplaceProducts": [
      {
        "Type": "Hardware",
        "Title": "A hardware product",
        "Url": "https://example.com",
        "Description": "My hardware product description"
      }
    ]
  },
  "BuyerEngagement": [
    {
      "EngagementOption": "RequestPrivateOffer"
    }
  ]
},
"EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/soln-abc123456",
"EntityIdentifier": "soln-abc123456@12",
"EntityType": "Solution@1.0",
"LastModifiedDate": "2025-04-22T01:02:48Z"
}

```

솔루션의 변경 기록 나열

특정 솔루션의 변경 기록을 보려면 개체 필터를 사용하여 ListChangeSets API 작업을 호출합니다.

요청 구문

```

POST /ListChangeSets HTTP/1.1
Content-type: application/json

```

```

{
  "Catalog": "AWSMarketplace",
  "FilterList": [
    {
      "Name": "EntityId",
      "ValueList": [
        "soln-abc123456"
      ]
    }
  ]
}

```

}

솔루션 및 변경 세트에 태그 지정

솔루션을 생성하는 동안 솔루션에 태그를 추가하고 세트를 변경하거나 나중에 업데이트할 수 있습니다.

생성 중 태그

다른 솔루션 CreateSolution 변경 유형이 아닌 변경 유형에 EntityTags만 추가할 수 있습니다. 모든 StartChangeSet페이로드ChangeSetTags에를 추가할 수 있습니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateSolution",
      "Entity": {
        "Type": "Solution@1.0"
      },
      "DetailsDocument": {
        "Name": "Test solution resource name"
      },
      "EntityTags": [
        {
          "Key": "Team",
          "Value": "Solutions"
        }
      ]
    }
  ],
  "ChangeSetTags": [
    {
      "Key": "Environment",
      "Value": "Production"
    }
  ]
}
```

기존 리소스의 태그 관리

TagResource 및 UntagResource API 작업을 사용하여 기존 솔루션의 태그를 관리하고 세트를 변경합니다. ListTagsForResource를 사용하여 현재 태그를 봅니다.

자세한 내용은 [리소스에 대한 태그 관리](#) 및 [개체에 태그 추가 및 생성 중 세트 변경을 참조하세요](#).

솔루션을 생성

CreateSolution 변경 유형을 사용하여 솔루션 ID와 이름만 있는 새 솔루션 문서를 생성할 수 있습니다. 다른 값은 아직 채워지지 않았습니다.

요청이 성공적으로 처리되면 AWS Marketplace에서 자동으로 Draft 상태의 솔루션을 생성합니다. 이 불완전한 솔루션은 AWS Marketplace의 구매자에게 표시되지 않습니다.

그런 다음 UpdateInformation, UpdateRelatedProducts, UpdateBuyerEngagementOptions (솔루션 생성에 선택 사항) 및와 같은 다른 변경 유형을 사용하여 Limited 솔루션을 완료합니다ReleaseSolution. ReleaseSolution 변경 유형은 Limited 솔루션에 필요한 모든 필수 필드가 솔루션에 있는지 확인한 다음 Limited 가시성으로 이동합니다.

Note

솔루션을 퍼블릭 상태로 이동하거나 솔루션 상태를 변경하려면 필요한 값을 제공한 후 UpdateVisibility 변경 유형을 사용합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateSolution",
      "Entity": {
        "Type": "Solution@1.0"
      },
      "ChangeName": "CreateSolutionChange",
      "DetailsDocument": {
        "Name": "solution_for_customerX_en",
        "Title": "New Partner Solution"
      }
    }
  ]
}
```

```

    }
  }
]
}

```

필드에 대한 정보를 제공하여 CreateSolution 변경 유형을 추가합니다.

- Entity (객체)(필수) - 생성 중인 솔루션 개체의 명명된 유형입니다.
 - Type (문자열)(필수) -는 이어야 Type 합니다Solution@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - Name (문자열)(필수) - 판매자의 참조용 AWS 리소스 이름(구매자가 볼 수 없음). 최대 길이: 100자. 선행 또는 후행 공백, 줄 바꿈 또는 제어 문자를 포함할 수 없습니다. UpdateInformation 변경 유형을 통해 솔루션 타일을 업데이트할 수도 있습니다.
 - Title (문자열)(선택 사항) - 솔루션의 제목입니다. 최대 길이: 255자. 선행 또는 후행 공백, 줄 바꿈 또는 제어 문자를 포함할 수 없습니다. UpdateInformation 변경 유형을 통해 솔루션 제목을 설정하거나 업데이트할 수도 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 응답은 ChangeSetId 및를 제공합니다ChangeSetArn.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. AWS Marketplace Management Portal 또는 DescribeChangeSet API 작업을 사용하여 상태를 확인할 수 있습니다.

요청이 성공적으로 완료되면(가 Status인 경우SUCCEEDED) 새 솔루션 ID가 생성됩니다.

비동기 오류

CreateSolution 작업은 [일반적인 비동기 오류 외에도 특정 오류](#)를 반환합니다. 변경 세트 처리가 완료된 DescribeChangeSet 후에서 이러한 오류가 발생합니다. 변경 요청 상태를 가져오는 방법을 알아보려면 [변경 세트 작업을](#) 참조하세요.

오류 코드	오류 메시지
INVALID_NAME	솔루션 이름 길이가 유효하지 않습니다. 1~100자의 이름을 입력하고 다시 시도하세요.
누락_이름	이름 값은 필수입니다. 이름 값을 지정하고 다시 시도하십시오.
INVALID_TITLE	솔루션 제목 길이가 유효하지 않습니다. 1~255자의 제목을 입력하고 다시 시도하세요.

솔루션 정보 업데이트

AWS Marketplace에 이미 솔루션이 있는 경우 카탈로그 API를 사용하여 솔루션 정보를 업데이트할 수 있습니다.

솔루션 정보를 업데이트하려면 UpdateInformation 변경 유형과 업데이트된 세부 정보를 사용하여 StartChangeSet API 작업을 호출합니다. 다음 예제를 참조하세요.

Note

UpdateInformation 변경 유형은 최상위 속성에 대한 HTTP PATCH 작업처럼 동작합니다. 필드 업데이트 방법:

- 에 최상위 속성이 있는 경우 DetailsDocument워크플로는 입력을 검증하고 기존 값을 덮어씁니다.
- 에 최상위 필드가 없으면 DetailsDocument기존 값이 보존됩니다.
- 선택적 필드의 설정을 해제하려면 해당 필드에 null 대해 명시적 JSON을 전송합니다.

AWS Marketplace는 로고 및 프로모션 미디어 자산을 확인하고 AWS Marketplace 소유 위치로 수집합니다. DescribeEntity는 StartChangeSet의 원래 입력이 아닌 수집된 파일에 대해 액세스 가능한 URL을 반환합니다. 제출된 원래 URLs 보려면 DescribeChangeSet API를 사용합니다.

Note

Public 솔루션에서 PromotionalMedia 자산 파일을 수정하는 경우(새 URLs 제공)는 AWS Marketplace 판매자 운영 팀의 수동 검토를 UpdateInformation 필요로 합니다. 이렇게 하

면 실행 시간이 늘어납니다. 이 시나리오에서는 자체 변경 세트에서 UpdateInformation 별도로 사용합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Identifier": "soln-1234abcd",
        "Type": "Solution@1.0"
      },
      "DetailsDocument": {
        "Name": "project_cold_harbor",
        "Title": "My Awesome Solution",
        "ShortDescription": "Solution value proposition",
        "LongDescription": "Detailed solution description explaining how this addresses customer needs",
        "LogoUrl": "https://s3.amazonaws.com/awsmmp-logos/logo.jpg",
        "PromotionalMedia": [
          {
            "Type": "Image",
            "Url": "https://s3.amazonaws.com/awsmmp-media/image.jpg",
            "Title": "Sample image",
            "Description": "Sample image description"
          },
          {
            "Type": "Video",
            "Url": "https://s3.amazonaws.com/awsmmp-media/video.mp4",
            "Title": "Sample video",
            "PreviewUrl": "https://s3.amazonaws.com/awsmmp-media/preview.png",
            "Description": "Sample video description"
          }
        ],
        "AdditionalResources": [
          {
            "Text": "Troubleshooting guide",
```

```

        "Url": "https://example.com/troubleshooting"
    },
    ],
    "UseCases": [
        {
            "Group": "advertising_and_marketing",
            "SubGroup": "ad_intelligence_and_measurement",
            "Item": "amazon_ads_insights",
            "Description": "Highlight description of this use case"
        }
    ]
}

```

필드에 대한 정보를 제공하여 UpdateInformation 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 명명된 엔터티 유형입니다.
 - Identifier (문자열)(필수) - 솔루션 ID입니다.
 - Type (문자열)(필수) - 여야 합니다Solution@1.0.
- DetailsDocument (객체)(필수) - 솔루션에 대해 업데이트하려는 정보를 포함하여 요청의 세부 정보입니다. 각 필드는 선택 사항이지만 업데이트할 변경 사항을 하나 이상 포함해야 합니다.
 - Name (문자열) - 카탈로그 API 읽기 작업을 통해 판매자에게만 표시되는 솔루션의 리소스 이름으로, 구매자가 볼 수 없습니다. 최대 길이: 100자. 선행 또는 후행 공백, 줄 바꿈 또는 제어 문자를 포함할 수 없습니다.
 - Title (문자열) - 구매자에게 표시할 솔루션의 제목입니다. 최대 길이: 255자. 선행 또는 후행 공백, 줄 바꿈 또는 제어 문자를 포함할 수 없습니다.
 - ShortDescription (문자열) - 구매자에게 표시할 솔루션의 주요 측면에 대한 가치 제안 설명입니다. 이는 일반적으로 2~3개의 문장입니다. 최대 길이: 1,000자. 선행 또는 후행 공백이나 제어 문자를 포함할 수 없습니다.
 - LongDescription (문자열) - 구매자에게 표시할 솔루션에 대한 더 긴 설명입니다. 일반적으로 1~3단락입니다. 최대 길이: 5,000자. 선행 또는 후행 공백이나 제어 문자를 포함할 수 없습니다.
 - LogoUrl (문자열) - 공개적으로 액세스할 수 있는 Amazon S3 버킷의 이미지 URL 또는 미리 서명된 S3 URL입니다. (CloudFront 또는 API Gateway 뒤에 있지 않은) 직접 S3 URL이어야 합니다. 최대 길이: 2,048자. 로고 사양은 [회사 및 제품 로고 요구 사항](#)을 참조하세요.

- **PromotionalMedia** (객체 배열) - 프로모션 이미지 및 비디오 목록입니다. 최대 15개 항목(비디오 5개 및 이미지 10개). 프로모션 미디어 사양은 [프로모션 미디어로 AWS Marketplace 제품 개선을 참조하세요](#).
- **Type** (문자열)(필수) - Image 또는 Video.
- **Url** (문자열)(필수) - 미디어 파일의 URL입니다. 직접 S3 URL 또는 미리 서명된 S3 URL이어야 합니다. 최대 길이: 2,048자. https URL이어야 합니다.
- **Title** (문자열)(필수) - 미디어의 제목입니다. 최대 길이: 100자. 선행 또는 후행 공백, 줄 바꿈 또는 제어 문자를 포함할 수 없습니다.
- **Description** (문자열)(필수) - 미디어에 대한 설명입니다. 최대 길이: 200자. 선행 또는 후행 공백이나 제어 문자를 포함할 수 없습니다.
- **PreviewUrl** (문자열)(선택 사항) - 비디오의 경우에만 미리 보기 이미지의 URL입니다. 최대 길이: 2,048자. https URL이어야 합니다.
- **AdditionalResources** (객체 배열) - 추가 리소스에 대한 참조 목록입니다. 최대 8개 항목.
 - **Text** (문자열) - 리소스의 이름 또는 제목입니다. 최대 길이: 500자. 선행 또는 후행 공백, 줄 바꿈 또는 제어 문자를 포함할 수 없습니다.
 - **Url** (문자열) - 리소스의 URL입니다. 최대 길이: 2,048자. https URL이어야 합니다.
- **UseCases** (객체 배열) - 이 솔루션이 처리하는 사용 사례 목록입니다. 최대 3개의 항목. 유효한 사용 사례 범주(그룹, SubGroup 및 항목의 조합)는 허용되는 활성 값과 [소개 레이블](#)에 대한 [사용 사례 범주 목록](#)을 참조하세요. 그룹, SubGroup 및 항목의 조합은 사용 사례 범주 목록에서 유효한 항목이어야 합니다.
 - **Group** (문자열) - 산업 또는 기술 범주입니다. [사용 사례 범주의 그룹 값 목록](#) 중 하나여야 합니다.
 - **SubGroup** (문자열) - 범주 내의 사용 사례 그룹입니다. [사용 사례 범주 목록](#)의 SubGroup 값 중 하나여야 합니다.
 - **Item** (문자열) - 특정 사용 사례. [사용 사례 범주의 항목 값 목록](#) 중 하나여야 합니다.
 - **Description** (문자열) - 이 솔루션이 사용 사례를 해결하는 방법에 대한 설명입니다. 최대 길이: 500자. 선행 또는 후행 공백이나 제어 문자를 포함할 수 없습니다.

미리 서명된 URL 정보는 [미리 서명된 URLs](#).

Note

UpdateInformation 변경 유형을 호출할 때 DetailsDocument 객체에서 업데이트할 필드의 하위 집합을 항상 제공할 수 있습니다. 그러나 솔루션을에서

로 이동Draft하기 전에 Limited, Title, 및 필드를 올바르게 설정해야 합니다ShortDescriptionLongDescriptionLogoUrl.
그러나 솔루션의 기존 필드를 업데이트할 때 변경 유형의 DetailsDocument 객체에 UpdateInformation 변경해야 하는 속성만 포함할 수 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 요청 상태를 확인하려면 AWS Marketplace 관리 포털을 사용하거나 DescribeChangeSet API를 호출합니다.

Public 솔루션에 대해 수정된 PromotionalMedia 자산으로이 변경 유형을 트리거한 후 완료하는데 최대 37일이 걸릴 수 있습니다. 여기에는 AWS Marketplace 판매자 운영 팀이 검토, 감사 및 승인해야 하는 시간이 포함됩니다.

비동기 오류

[일반적인 비동기 오류](#) 외에도 다음 오류는 AWS Marketplace Catalog API의 UpdateInformation 작업과 관련이 있습니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [변경 세트 작업을](#) 참조하세요.

오류 코드	오류 메시지
INVALID_SHORT_DESCRIPTION	솔루션 ShortDescription 길이가 유효하지 않습니다. 1~1,000자의 ShortDescription을 입력하고 다시 시도하세요.
INVALID_LONG_DESCRIPTION	솔루션 LongDescription 길이가 유효하지 않습니다. 1~5,000자의 LongDescription을 입력하고 다시 시도하세요.

오류 코드	오류 메시지
INVALID_NAME	솔루션 이름 길이가 유효하지 않습니다. 1~100자의 이름을 입력하고 다시 시도하세요.
INVALID_TITLE	솔루션 제목 길이가 유효하지 않습니다. 1~255자의 제목을 입력하고 다시 시도하세요.
INVALID_ADDITIONAL_RESOURCES	AdditionalResources[%d] 솔루션.텍스트 길이가 유효하지 않습니다. 1~500자의 텍스트를 입력하고 다시 시도하세요.
INVALID_ADDITIONAL_RESOURCES	제공된 추가 리소스의 URLs은 액세스할 수 없습니다. [%s]. 유효하고 액세스 가능한 URLs 제공
INVALID_ADDITIONAL_RESOURCES	URL 값이 필요합니다. URL 값을 지정하고 다시 시도하십시오.
INVALID_ADDITIONAL_RESOURCES	텍스트는 누락될 수 없습니다. 텍스트 제공
INVALID_USE_CASE	솔루션 UseCases[%d].설명 길이가 유효하지 않습니다. 1~500자의 설명을 입력하고 다시 시도하세요.
INVALID_USE_CASE	UseCases[%d].그룹: 제공된 그룹(%s)이 잘못되었습니다. 유효한 그룹을 제공합니다.
INVALID_USE_CASE	UseCases[%d].SubGroup: 제공된 그룹(%s)이 잘못되었습니다. 유효한 SubGroup 제공합니다.
INVALID_USE_CASE	UseCases[%d].항목: 제공된 항목(%s)이 잘못되었습니다. 유효한 항목을 입력합니다.
INVALID_USE_CASE	UseCases[%d].항목: 제공된 사용 사례가 비활성 상태입니다. 활성 사용 사례 제공
INVALID_USE_CASE	UseCases[%d].그룹: 그룹이 누락될 수 없습니다. 그룹을 제공합니다.
INVALID_USE_CASE	UseCases[%d].SubGroup: SubGroup 누락될 수 없습니다. SubGroup 제공합니다.

오류 코드	오류 메시지
INVALID_USE_CASE	UseCases[%d].항목: SubGroup 누락될 수 없습니다. 항목을 제공합니다.
INVALID_MEDIA	솔루션은 PromotionalMedia[%d]입니다.제목 길이가 유효하지 않습니다. 1~100자의 제목을 입력하고 다시 시도하세요.
INVALID_MEDIA	솔루션은 PromotionalMedia[%d]입니다.설명 길이가 유효하지 않습니다. 1~200자의 설명을 입력하고 다시 시도하세요.
INVALID_MEDIA	잘못된 URL: %s S3에 저장된 미디어에 대한 새 URL을 제공합니다.
INVALID_MEDIA	미디어 위치에 액세스할 수 없음: %s [S3] 위치 중 하나에 저장된 미디어에 대한 액세스 가능한 새 URL을 제공합니다.
INVALID_MEDIA	이미지 크기가 %s를 초과합니다. %s 미만의 이미지를 제공합니다.
INVALID_MEDIA	비디오 크기가 %s를 초과합니다. %s 미만의 비디오를 제공합니다.
INVALID_MEDIA	로고 크기가 %s를 초과합니다. %s 미만의 로고를 제공합니다.
INVALID_MEDIA	%s에서 맬웨어가 감지되었습니다. 맬웨어 없이 미디어를 제공합니다.
INVALID_MEDIA	부적절한 콘텐츠: %s 감지됨. 부적절한 콘텐츠가 없는 미디어 제공
INVALID_MEDIA	%s의 명시적 콘텐츠: '{ExplicitContent}'가 감지되었습니다. 명시적 콘텐츠 없이 미디어를 제공합니다.
INVALID_MEDIA	PromotionalMedia[%s].URL: %s가 지원되는 형식(%s)이 아닙니다. 지원되는 형식의 올바른 이미지 사용: [JPEG, PNG].
INVALID_MEDIA	PromotionalMedia[%s].URL: %s가 지원되는 형식(%s)이 아닙니다. 지원되는 형식인 [MP4, MOV]의 잘 구성된 비디오를 사용합니다.

오류 코드	오류 메시지
INVALID_MEDIA	%s의 LogoUrl에 잘못된 종횡비가 있습니다. 이미지에 1:1(정사각형) 또는 2:1(넓은 쪽) 종횡비를 제공합니다.
INVALID_MEDIA	로고: %s에 잘못된 차원이 있습니다. 120x120~1080x1080픽셀의 이미지를 제공합니다.
INVALID_MEDIA	PromotionalMedia[%d].유형은 지원되지 않습니다. 지원되는 유형으로 PromotionalMedia 제공: %s
DUPLICATE_MEDIA	중복 미디어 감지됨: [%s] 중복이 없는 미디어 항목을 제공합니다.
INVALID_MEDIA	PromotionalMedia[%d].URL: %s에서 맬웨어가 감지되었습니다. 맬웨어 없이 미디어를 제공합니다.
TOO_MANY_ITEMS	%d%s 이하를 제공합니다.

관련 제품 업데이트

카탈로그 API를 사용하여 AWS Marketplace 제품 및 비 AWS Marketplace 제품을 포함하여 솔루션에 포함된 제품 및 서비스를 업데이트할 수 있습니다.

관련 제품을 업데이트하려면 다음 예제와 같이 UpdateRelatedProducts 변경 유형으로 StartChangeSet API 작업을 호출합니다.

Note

UpdateRelatedProducts 변경 유형은 최상위 속성에 대한 HTTP PATCH 작업처럼 동작합니다. 예제와 같이 모든 현재 AWS Marketplace 제품 유형을 제공할 수 AwsMarketplaceProducts 있습니다UpdateRelatedProducts.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```



```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateRelatedProducts",
      "Entity": {
        "Type": "Solution@1.0",
        "Identifier": "soln-123456"
      },
      "DetailsDocument": {
        "IntegrationDescription": "How to use the products in this solution (usage guide)",
        "AwsMarketplaceProducts": [
          {
            "ProductId": "prod-123"
          },
          {
            "ProductId": "prod-456"
          }
        ],
        "NonAwsMarketplaceProducts": [
          {
            "Type": "Hardware",
            "Title": "Diamond engine",
            "Description": "Hardware details",
            "Url": "https://example.com/hardware-product-specs.html"
          }
        ]
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateRelatedProducts 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 명명된 엔터티 유형입니다.
 - Identifier (문자열)(필수) - 솔루션 ID입니다.
 - Type (문자열)(필수) - 여야 합니다Solution@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다. 각 필드는 선택 사항이지만 업데이트할 변경 사항을 하나 이상 포함해야 합니다.

- **IntegrationDescription** (문자열) -이 솔루션에서 제품이 함께 작동하는 방식에 대한 설명입니다. 최대 길이: 5,000자. 선행 또는 후행 공백이나 제어 문자를 포함할 수 없습니다.
- **AwsMarketplaceProducts** (객체 배열) -이 솔루션에 포함된 AWS Marketplace 제품. 최대 5개 항목
 - **ProductId** (문자열) - AWS Marketplace 제품 ID(UUID 또는 로 시작하는 문자열prod-). 최대 길이: 50자.
- **NonAwsMarketplaceProducts** (객체 배열) - AWS Marketplace에서 사용할 수 없는 제품. 최대 5개 항목.
 - **Type** (문자열)(필수) - 제품 유형입니다. 가능한 값: Software, Consulting, Hardware, , Communication, Professional Service, Managed Service, Value-Added Resale Training
 - **Title** (문자열)(필수) - 제품의 제목입니다. 최대 길이: 255자. 선행 또는 후행 공백, 줄 바꿈 또는 제어 문자를 포함할 수 없습니다.
 - **Description** (문자열)(선택 사항) - 제품에 대한 설명입니다. 최대 길이: 220자. 선행 또는 후행 공백이나 제어 문자를 포함할 수 없습니다.
 - **Url** (문자열) (선택 사항) - 제품에 대한 자세한 정보를 제공하는 URL입니다. 최대 길이: 2,048자. https URL이어야 합니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 **ChangeSetArn** 대한 **ChangeSetId** 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다.

요청 상태를 확인하려면 AWS Marketplace 관리 포털을 사용하거나 **DescribeChangeSet** API를 호출합니다.

비동기 오류

일반적인 비동기 오류 외에도 다음 오류는 AWS Marketplace Catalog API의 UpdateRelatedProducts 작업에 고유합니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 변경 세트 작업을 참조하세요.

오류 코드	오류 메시지
INCOMPATIBLE_OPERATION_UPDATE_RELATED_PRODUCTS_SOLUTION	관련 제품을 업데이트하는 작업은 퍼블릭 솔루션에서 지원되지 않습니다. 이렇게 변경하려면 먼저 솔루션의 가시성을 제한으로 변경하고 다시 시도하세요.
INVALID_AWS_MARKETPLACE_PRODUCTS	계정에 AWS Marketplace 제품 '%s'를 사용할 수 없습니다. 허용 목록에 액세스할 수 있는 퍼블릭 제품 또는 제한된 제품을 선택하거나 판매자에게 문의하여 액세스를 요청합니다.
INVALID_AWS_MARKETPLACE_PRODUCTS	지정된 제품 유형 '%s'는 지원되지 않습니다. [AmiProduct, SaaSProduct, ContainerProduct, MachineLearningProduct, ProfessionalServicesProduct, DataProduct]와 같은 유효한 제품 유형 중 하나를 제공하고 다시 시도하세요.
INVALID_AWS_MARKETPLACE_PRODUCTS	'%s' 제품을 찾을 수 없습니다. 동일한 카탈로그에 유효한 제품 ID를 제공합니다.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	지정된 비 AWS Marketplace 제품 URLs 액세스할 수 없습니다. [%s]. URLs이 유효하고 공개적으로 액세스할 수 있는지 확인하고 다시 시도하세요.
INVALID_INTEGRATION_DESCRIPTION	1~5,000자의 솔루션 IntegrationDescription을 제공합니다.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	NonAwsMarketplaceProduct[%d] 솔루션.제목 길이가 유효하지 않습니다. 1~255자의 설명을 입력하고 다시 시도하세요.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	NonAwsMarketplaceProduct[%d] 솔루션.설명 길이가 유효하지 않습니다. 1~220자의 설명을 입력하고 다시 시도하세요.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	NonAwsMarketplaceProduct[%d] 솔루션.유형 길이가 유효하지 않습니다. 1~50자의 유형을 입력하고 다시 시도하세요.

오류 코드	오류 메시지
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	NonAwsMarketplaceProduct[%d].유형이 AWS Marketplace가 아닌 유효한 제품 유형이 아닙니다. AWS Marketplace가 아닌 다음 제품 유형 중 하나를 제공합니다. [%s]
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	NonAwsMarketplaceProduct[%d].제목 값은 필수입니다. 제목 값을 지정하고 다시 시도하십시오.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	NonAwsMarketplaceProduct[%d].유형 값은 필수입니다. 유형 값을 지정하고 다시 시도하십시오.
TOO_MANY_PRODUCTS	연결할 수 있는 최대 AWS Marketplace 제품 수는 %d개입니다. 새 제품을 추가하기 전에 기존에 연결된 제품을 제거합니다.
TOO_MANY_PRODUCTS	추가할 수 있는 AWS Marketplace 제품의 최대 수는 %d개입니다. 새 제품을 추가하기 전에 기존 AWS Marketplace 제품을 제거합니다.
INVALID_AWS_MARKETPLACE_PRODUCTS	중복 AWS Marketplace 제품 IDs는 허용되지 않습니다. 각 제품 ID는 솔루션에서 고유해야 합니다.
INVALID_REMOVE_OPERATION	잘못된 작업입니다. 모든 제품을 제거할 수는 없습니다.

솔루션 릴리스

카탈로그 API를 사용하여 AWS Marketplace의 Limited 가시성에 Draft 대한 솔루션을 게시할 수 있습니다.

ReleaseSolution 변경 유형은 Limited 솔루션의 모든 필수 필드가 있는지 확인합니다. 그런 다음 솔루션을에서 Limited 가시성Draft으로 이동합니다. 솔루션이 Limited 가시성 상태가 되면 Draft 가시성으로 되돌릴 수 없습니다.

솔루션을 릴리스하려면 다음 예제와 같이 ReleaseSolution 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
```

```
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseSolution",
      "Entity": {
        "Type": "Solution@1.0",
        "Identifier": "soln-1234abcd"
      },
      "DetailsDocument": {}
    }
  ]
}
```

필드에 대한 정보를 제공하여 ReleaseSolution 변경 유형을 추가합니다.

- Entity (객체)(필수) - 릴리스되는 명명된 유형의 개체입니다.
 - Identifier (문자열)(필수) - 솔루션 ID입니다.
 - Type (문자열)(필수) - 여야 합니다Solution@1.0.
- DetailsDocument (객체)(필수) - 빈 객체여야 합니다. 변경 유형은 세부 정보를 허용하지 ReleaseSolution 않습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 DescribeChangeSet API를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

[일반적인 비동기 오류](#) 외에도 다음 오류는 AWS Marketplace Catalog API의 ReleaseSolution 작업과 관련이 있습니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [변경 세트 작업을](#) 참조하세요.

오류 코드	오류 메시지
누락_필수_필드	요청에 필수 파라미터 [%s]가 없습니다. 모든 필수 파라미터의 값을 지정하고 다시 시도하세요.
INCOMPATIBLE_OPERATION_RELEASE_SOLUTION	변경 유형은 초안 상태의 솔루션에서만 호출할 수 있습니다. 솔루션의 가시성을 초안으로 업데이트하고 다시 시도하세요.

구매자 참여 옵션 업데이트

카탈로그 API를 사용하여 구매자의 솔루션 참여 방식을 구성할 수 있습니다.

Note

이 변경 유형을 사용하려면 [APN Customer Engagements\(ACE\)](#) 프로그램에 활성 등록이 있어야 합니다. 또한 AWS 계정을 ACE에 등록된 AWS Partner Central 계정과 연결해야 합니다. 지침은 [AWS Partner Central 및 AWS 계정 연결을 참조하세요](#).

구매자 참여 옵션을 업데이트하려면 다음 예제와 같이 UpdateBuyerEngagementOptions 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateBuyerEngagementOptions",
      "Entity": {
```

```

    "Type": "Solution@1.0",
    "Identifier": "soln-1234abcd"
  },
  "DetailsDocument": [
    {
      "EngagementOption": "RequestProductDemo"
    },
    {
      "EngagementOption": "RequestPrivateOffer"
    }
  ]
}

```

필드에 대한 정보를 제공하여 UpdateBuyerEngagementOptions 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트되는 엔터티의 명명된 유형입니다.
 - Identifier (문자열)(필수) - 솔루션 ID입니다.
 - Type (문자열)(필수) - 여야 합니다Solution@1.0.
- DetailsDocument (객체 배열)(필수) - 활성화할 구매자 참여 옵션입니다.
 - EngagementOption (문자열)(필수) - 참여 옵션의 유형입니다. 유효한 값: RequestPrivateOffer, RequestProductDemo.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다.

요청 상태를 확인하려면 AWS Marketplace 관리 포털을 사용하거나 DescribeChangeSet API를 호출합니다.

비동기 오류

[일반적인 비동기 오류](#) 외에도 다음 오류는 AWS Marketplace Catalog API의 UpdateBuyerEngagementOptions 작업에 고유합니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [변경 세트 작업을](#) 참조하세요.

오류 코드	오류 메시지
부적격_판매자	ACE 자격 검증이 완료되지 않았습니다. 계속하려면 1) AWS Marketplace 계정을 APN 계정에 연결, 2) ACE 자격 요구 사항을 완료합니다.

솔루션 가시성 업데이트

카탈로그 API를 사용하여 AWS Marketplace에서 솔루션의 가시성을 업데이트할 수 있습니다.

대상 가시성을 Limited, Public 또는 Restricted로 설정할 수 있습니다.

- **Limited** - 솔루션이 완료되었으며 ReleaseSolution 변경 유형을 성공적으로 완료했습니다. 이 가시성에서 솔루션의 구매자 웹 사이트 세부 정보를 미리 볼 수 있습니다. AWS Marketplace 구매자 웹 사이트에서 솔루션은 소유 계정, 솔루션에 포함된 각 AWS Marketplace 제품의 소유자 및 AWS Marketplace 판매자 운영 팀에만 표시됩니다.
- **Public** - 솔루션은 AWS Marketplace에서 볼 수 있습니다. 구매자는 솔루션을 보고 참여할 수 있습니다.
- **Restricted** - AWS Marketplace 구매자 웹 사이트에서 더 이상 솔루션이 공개되지 않습니다.

Note

솔루션을 Public 가시성으로 전환하려면 다음 요구 사항을 충족해야 합니다.

- AWS Marketplace에 퍼블릭 프로필이 있어야 합니다.
- 솔루션에는 제품이 Public 아닌 제품이 없어야 합니다.
- 솔루션에는 최소 2개의 Public AWS Marketplace 제품이 있어야 합니다.
- 솔루션에는 동일한 AWS 계정의 AWS Marketplace 제품이 1개 이상 있어야 합니다.
- 솔루션에는 RequestPrivateOffer 구매자 참여 옵션이 활성화되어 있어야 합니다.

- AWS Marketplace 판매자 운영 팀은 솔루션을 Public 가시성으로 전환하기 위한 모든 요청을 검토합니다. 지원은 [AWS Marketplace에 대한 지원 받기](#)를 참조하세요.

를 TargetVisibility로 설정하면 Public UpdateVisibility 변경 유형을 AWS Marketplace 판매자 운영 팀의 수동 검토가 필요합니다. 이렇게 하면 실행 시간이 늘어납니다. 자체 변경 세트에서 UpdateVisibility 별도로 사용합니다.

솔루션의 가시성을 업데이트하려면 다음 예제와 같이 UpdateVisibility 변경 유형으로 StartChangeSet API 작업을 호출합니다.

요청 구문

TargetVisibility가 Public 또는 인 경우 Limited.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateVisibility",
      "Entity": {
        "Type": "Solution@1.0",
        "Identifier": "soln-1234abcd"
      },
      "DetailsDocument": {
        "TargetVisibility": "Public"
      }
    }
  ]
}
```

TargetVisibility가 인 경우 입니다Restricted.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
```

```

"ChangeSet": [
  {
    "ChangeType": "UpdateVisibility",
    "Entity": {
      "Type": "Solution@1.0",
      "Identifier": "soln-1234abcd"
    },
    "DetailsDocument": {
      "TargetVisibility": "Restricted"
    }
  }
]
}

```

필드에 대한 정보를 제공하여 UpdateVisibility 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트되는 엔터티의 명명된 유형입니다.
 - Identifier (문자열)(필수) - 솔루션 ID입니다.
 - Type (문자열)(필수) - 여야 합니다Solution@1.0.
- DetailsDocument (객체)(필수) - 변경 세트를 실행하는 데 필요한 세부 정보입니다.
 - TargetVisibility (문자열)(필수) - 솔루션의 의도한 새로운 가시성입니다. 가능한 값: Public, Limited 및 Restricted

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 DescribeChangeSet API 를 통해 직접 요청 상태를 확인할 수 있습니다.

를 TargetVisibility로 사용하여이 변경 유형을 트리거Public한 후 완료하는 데 최대 37일이 걸릴 수 있습니다. 여기에는 AWS Marketplace 판매자 운영 팀이 검토, 감사 및 승인해야 하는 시간이 포함됩니다.

비동기 오류

다음 오류는 AWS Marketplace Catalog API의 UpdateVisibility 작업에만 해당됩니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [변경 세트 작업을](#) 참조하세요.

오류 코드	오류 메시지
INVALID_VISIBILITY_TRANSITION	솔루션을 초안 상태로 다시 이동할 수 없습니다. 일단 릴리스되면 솔루션은 제한, 퍼블릭 및 제한 상태 간에만 전환할 수 있습니다.
TOO_FEW_USE_CASES	솔루션을 공개하려면 하나 이상의 사용 사례가 필요합니다. 사용 사례를 추가하고 다시 시도하세요.
TOO_FEW_PUBLIC_PRODUCTS	솔루션을 공개하려면 최소 2개의 퍼블릭 AWS Marketplace 제품이 필요합니다. 필요한 제품을 추가하고 다시 시도하세요.
INCOMPATIBLE_PRODUCTS	솔루션에는 하나 이상의 비공개 AWS Marketplace 제품이 포함되어 있습니다. 솔루션을 공개하려면 모든 비공개 제품을 제거하고 다시 시도하세요.
TOO_FEW_SELF_OWNED_PRODUCTS	솔루션에는 솔루션 소유자가 소유한 AWS Marketplace 제품이 하나 이상 필요합니다. 계정에서 AWS Marketplace 제품을 추가하고 다시 시도하세요.
INCOMPATIBLE_BUYER_ENGAGEMENT_OPTIONS	이 솔루션을 공개하려면 RequestPrivateOffer 구매자 참여 옵션을 활성화해야 합니다. RequestPrivateOffer를 활성화하고 다시 시도하세요.
INCOMPATIBLE_OPERATION_UPDATE_VISIBILITY	초안 상태의 솔루션에는 UpdateVisibility가 지원되지 않습니다. 먼저 ReleaseSolution을 사용하여 초안에서 제한으로 솔루션을 게시한 다음 UpdateVisibility를 사용하여 퍼블릭 또는 제한으로 변경합니다.

오류 코드	오류 메시지
LIMIT_EXCEEDED_TOO_MANY_PUBLIC_SOLUTIONS	이 계정은 퍼블릭 솔루션 5개 한도에 도달했습니다. 기존 퍼블릭 솔루션을 제한 또는 제한된 가시성으로 업데이트하고 다시 시도하세요.
MISSING_SELLER_PROFILE_INFORMATION	솔루션을 공개하려면 공개 판매자 프로필이 필요합니다. 판매자 계정에서 퍼블릭 프로필을 생성하고 다시 시도하세요.

일반적인 비동기 오류 코드

URL 유효성 및 지원되지 않는 문자 확인과 같이 모든 솔루션 변경 유형에 적용되는 일부 비동기 검증이 있으며, 오류 코드와 오류 메시지가 여기에 나열되어 있습니다.

제공된 각 URL 필드는 HTTPS 프로토콜을 사용해야 하며 RFC2396 또는 RFC 2732 표준을 준수해야 합니다.

오류 코드	오류 메시지
INVALID_INPUT	'%s' 필드가 잘못되었습니다. 지원되지 않는 문자 %s를 제거합니다.
INVALID_INPUT	%s 필드에 부적절한 '%s' 콘텐츠가 있습니다. %s에 부적절한 콘텐츠를 제공하지 않습니다.
UNSUPPORTED_CATALOG	요청된 카탈로그 %s는 이 변경 유형에서 지원되지 않습니다.
INVALID_REMOVE_OPERATION	잘못된 작업, %s를 제거할 수 없음
감사_오류	(AWS Marketplace 판매자 운영 팀 감사 결과에 따라 다름)

솔루션 수명 주기

솔루션은 다음 가시성 값을 통해 진행됩니다.

- Draft - 생성 후 초기 가시성. 다른 변경 유형을 사용하여 정보를 점진적으로 추가할 수 있습니다.

- **Limited** - 솔루션이 완료되었으며 사용자와 제품 소유자가 미리 볼 수 있습니다. 를 사용하여 이를 달성할 수 있습니다 `ReleaseSolution`.
- **Public** - AWS Marketplace 웹 사이트를 탐색하는 모든 구매자가 솔루션을 볼 수 있습니다. 이를 위해서는 추가 검증이 필요합니다.
- **Restricted** - 신규 구매자가 더 이상 솔루션을 볼 수 없습니다.

⚠ Important

솔루션이에서 **Limited** 가시성 `Draft`으로 이동하면 다시 로 이동할 수 없습니다 `Draft`. 는 **Limited** 가시성으로 전환하기 전에 **Limited** 솔루션의 모든 필수 필드가 있는지 `ReleaseSolution` 확인합니다.

전체 솔루션의 최소 요구 사항

전체 솔루션을 생성하고 초안 상태 이후에 게시하려면 다음과 같은 변경 유형 및 속성이 필요합니다.

- **CreateSolution**: Name
- **UpdateInformation**: Title, ShortDescription, LongDescription, LogoUrl
- **UpdateRelatedProducts**: IntegrationDescription, AwsMarketplaceProducts 및에 대해 하나 이상의 항목 NonAwsMarketplaceProducts
- **ReleaseSolution**

다음 예제에서는 단일 변경 세트에서 **Limited** 가시성이 있는 완전한 솔루션을 생성하는 방법을 보여줍니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateSolution",
      "Entity": {
        "Type": "Solution@1.0"
      },
      "ChangeName": "CreateSolutionChange",
      "DetailsDocument": {
        "Name": "complete_solution_example",
        "Title": "Complete Solution Example"
      }
    }
  ]
}
```

```

    }
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Solution@1.0",
      "Identifier": "$CreateSolutionChange.Entity.Identifier"
    },
    "ChangeName": "UpdateInformationChange",
    "DetailsDocument": {
      "ShortDescription": "A comprehensive solution for customer needs",
      "LongDescription": "This solution combines multiple products to address
specific customer requirements in the advertising and marketing space.",
      "LogoUrl": "https://s3.amazonaws.com/awsmmp-logos/solution-logo.jpg",
      "UseCases": [
        {
          "Group": "advertising_and_marketing",
          "SubGroup": "ad_intelligence_and_measurement",
          "Item": "amazon_ads_insights",
          "Description": "Provides comprehensive advertising insights and measurement
capabilities"
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateRelatedProducts",
    "Entity": {
      "Type": "Solution@1.0",
      "Identifier": "$CreateSolutionChange.Entity.Identifier"
    },
    "ChangeName": "UpdateRelatedProductsChange",
    "DetailsDocument": {
      "AwsMarketplaceProducts": [
        {
          "ProductId": "prod-123"
        }
      ],
      "IntegrationDescription": "These products work together to provide end-to-end
advertising analytics and optimization capabilities."
    }
  },
  {
    "ChangeType": "UpdateBuyerEngagementOptions",

```

```

    "Entity": {
      "Type": "Solution@1.0",
      "Identifier": "$CreateSolutionChange.Entity.Identifier"
    },
    "ChangeName": "UpdateBuyerEngagementChange",
    "DetailsDocument": [
      {
        "EngagementOption": "RequestPrivateOffer"
      }
    ]
  },
  {
    "ChangeType": "ReleaseSolution",
    "Entity": {
      "Type": "Solution@1.0",
      "Identifier": "$CreateSolutionChange.Entity.Identifier"
    },
    "ChangeName": "ReleaseSolutionChange",
    "DetailsDocument": {}
  }
]
}

```

솔루션에 대한 IAM 권한

AWS Marketplace Catalog API는 표준 IAM 권한을 사용합니다. 자세한 내용은 [AWS Marketplace Catalog API에 대한 액세스 제어](#)를 참조하세요.

AWSMarketplaceSellerProductFullAccess를 사용하여 AWS Management Console에서 Catalog API 및 Partner Central 3.0을 통해 솔루션에 액세스하는 데 필요한 모든 작업을 수행할 수 있습니다.

솔루션에 대한 샘플 IAM 정책

다음 샘플 IAM 정책은 Solution@1.0 엔터티에 대한 특정 작업(DescribeEntity, ListEntities 및 StartChangeSet)을 허용합니다. 보안 주체의 액세스를 특정 엔터티 IDs의 특정 변경 유형에 제한하도록 추가로 범위를 좁히거나, 보안 주체가 Catalog API에서 지원하는 다른 엔터티 유형에서도 작동할 수 있도록 범위를 넓힐 수 있습니다.

```

{
  "Statement": [
    {
      "Effect": "Allow",

```

```

    "Action": [
      "aws-marketplace:StartChangeSet"
    ],
    "Resource": [
      "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/*",
      "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
    ],
    "Condition": {
      "StringEquals": {
        "catalog:ChangeType": [
          "CreateSolution",
          "UpdateInformation",
          "UpdateRelatedProducts",
          "ReleaseSolution",
          "UpdateBuyerEngagementOptions",
          "UpdateVisibility"
        ]
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:ListEntities",
      "aws-marketplace:DescribeEntity"
    ],
    "Resource": [
      "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "aws-marketplace:DescribeChangeSet",
      "aws-marketplace:ListChangeSets",
      "aws-marketplace:CancelChangeSet"
    ],
    "Resource": [
      "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [

```



```

        "aws-marketplace:TagResource",
        "aws-marketplace:UntagResource",
        "aws-marketplace:ListTagsForResource"
    ],
    "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/*",
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/*"
    ]
}
]
}
```

특정 솔루션에 대한 액세스 제한

리소스 ARN의 솔루션 ID를 사용하여 특정 솔루션에 대한 액세스를 제한할 수 있습니다.

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/soln-specific123"
      ]
    }
  ]
}
```

특정 변경 유형에 대한 액세스 제한

조건 키를 사용하여 특정 변경 유형에 대한 액세스를 제한할 수 있습니다.

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet"
      ],

```

```

    "Resource": [
      "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/*"
    ],
    "Condition": {
      "StringEquals": {
        "catalog:ChangeType": [
          "UpdateInformation",
          "UpdateRelatedProducts"
        ]
      }
    }
  }
}

```

제안 세트 작업

AWS Marketplace 제안 세트를 사용하면 판매자가 여러 비공개 제안을 구매자를 위한 통합 다중 제품 거래 가능 패키지로 패키징할 수 있습니다. 이 기능은 복잡한 엔터프라이즈 환경에서 포괄적인 end-to-end 솔루션에 대한 증가하는 고객 수요를 해결합니다.

일반적으로 구매자는 여러 판매자로부터 개별 제품을 검색하고 조달해야 했기 때문에 경험 조각화, 통합 문제 및 조달 주기가 길어졌습니다. 제안 세트는 AWS Marketplace 파트너가 여러 제안을 결합하여 고객이 일회성 승인으로 모든 구성 요소를 검토하고 수락할 수 있는 조정된 비공개 제안을 통해 트랜잭션 프로세스를 간소화함으로써 이를 간소화합니다.

제안 세트는 솔루션과 함께 작동하여 포괄적인 다중 제품을 제공합니다. 솔루션은 자세한 설명, 아키텍처 다이어그램 및 사용 사례 설명서를 비롯한 풍부한 마케팅 콘텐츠를 제공하는 반면, 제안 세트는 통합 조달을 위한 거래 가능한 패키징 및 상용 용어를 처리합니다.

제안 세트 엔터티

제안 세트는 2~7개의 비공개 제안을 단일 거래 가능 패키지로 그룹화하는 컨테이너입니다. 제안 세트 내의 각 제안은 고유한 요금, 결제 조건, 기간 및 최종 사용자 라이선스 계약(EULA)을 유지하는 반면, 제안 세트는 구매자에게 통합된 검색 및 수락 경험을 제공합니다.

주요 특성:

- 여러 비공개 제안을 그룹화합니다(최소 2개, 최대 7개).
- 단일 작업으로 모든 제안을 통합 구매자 수락 가능
- 각 제안에 대해 고유한 용어로 유연성을 유지합니다.

- 각 제품에 대해 별도의 계약을 생성하여 구매 후 독립 관리를 허용합니다.
- 향상된 마케팅 콘텐츠를 위해 솔루션과 선택적으로 연결할 수 있습니다.

제안 세트 엔터티 유형은 `OfferSet@1.0`.

개체 식별자

제안 세트는 `offerset-*` 식별자 형식(예: `offerset-abc123def456`)을 사용합니다.

개체 ARN 형식

제안 세트는 표준 AWS ARN 형식을 따릅니다.

```
arn:aws:aws-marketplace:{region}:{account-id}:AWSMarketplace/OfferSet/{offerset-id}
```

예제:

```
arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/OfferSet/offerset-abc123def456
```

제안 세트 시작하기

이 섹션에서는 AWS Marketplace 카탈로그 API [변경 유형을](#) 사용하여 제안 세트를 생성하고 관리하는 방법에 대한 자세한 정보를 제공합니다.

시작하기 전에, 다음 사항을 확인해야 합니다.

- 유효한 AWS Marketplace 판매자 등록 및 공개 프로필. 자세한 내용은 [판매자 프로필 등록 및 생성을 참조하세요](#).
- API에 대한 액세스 및 완료된 판매자 사전 조건. 자세한 내용은 [AWS Marketplace Catalog API에 대한 액세스 제어를 참조하세요](#).
- AWS Marketplace Catalog API 사용의 기본 사항에 대해 알아보려면 [AWS Marketplace Catalog API 사용을 참조하세요](#).

제안 세트 나열

계정의 모든 제안 세트를 나열하려면 `EntityType` 설정된 `ListEntities` API 작업을 호출합니다. `OfferSet`.

요청

```
POST /ListEntities HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "EntityType": "OfferSet"
}
```

응답

```
{
  "EntitySummaryList": [
    {
      "EntityType": "OfferSet",
      "EntityId": "offerset-xyz123",
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/OfferSet/offerset-xyz123",
      "LastModifiedDate": "2025-06-25T23:46:25Z",
      "OfferSetSummary": {
        "Name": "Enterprise Security Solution Offer Set",
        "State": "Released",
        "ReleaseDate": "2025-06-25T23:46:20Z",
        "SolutionId": "soln-abc987",
        "AssociatedOfferIds": [
          "offer-abc123",
          "offer-def456",
          "offer-ghi789"
        ]
      }
    }
  ],
  "NextToken": null
}
```

상태별로 제안 세트 필터링

EntityTypeFilters 파라미터를 사용하여 상태별로 제안 세트를 필터링할 수 있습니다.

요청

```
POST /ListEntities HTTP/1.1
```

```
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "OfferSet",
  "EntityTypeFilters": {
    "OfferSetFilters": {
      "State": {
        "ValueList": [
          "Released"
        ]
      }
    }
  }
}
```

솔루션 ID를 기준으로 제안 세트 필터링

EntityTypeFilters 파라미터를 사용하여 솔루션 ID별로 제안 세트를 필터링할 수 있습니다.

요청

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "OfferSet",
  "EntityTypeFilters": {
    "OfferSetFilters": {
      "SolutionId": {
        "ValueList": [
          "soln-abc987"
        ]
      }
    }
  }
}
```

릴리스 날짜별로 제안 세트 필터링

릴리스 날짜별로 제안 세트를 필터링하고 EntityTypeFilters 및 EntityTypeSort 파라미터를 사용하여 결과를 정렬할 수 있습니다.

요청

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "OfferSet",
  "EntityTypeFilters": {
    "OfferSetFilters": {
      "ReleaseDate": {
        "AfterValue": "2025-11-01"
      }
    }
  },
  "EntityTypeSort": {
    "OfferSetSort": {
      "SortBy": "ReleaseDate",
      "SortOrder": "DESCENDING"
    }
  }
}
```

제안 세트 설명

특정 제안 세트에 대한 자세한 정보를 가져오려면 DescribeEntity API 작업을 호출합니다.

요청

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=offerset-xyz123 HTTP/1.1
```

응답

```
{
  "EntityType": "OfferSet@1.0",
  "EntityId": "offerset-xyz123",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/OfferSet/offerset-xyz123",
  "LastModifiedDate": "2025-06-25T23:46:25Z",
  "DetailsDocument": {
    "Id": "offerset-xyz123",
    "Name": "Enterprise Security Solution Offer Set",
  }
}
```

```

    "BuyerNotes": "Complete security solution including endpoint protection and network monitoring",
    "State": "Released",
    "SolutionId": "soln-abc987",
    "ReleaseDate": "2025-06-25T23:46:20Z",
    "AssociatedOffers": [
      {
        "OfferId": "offer-abc123"
      },
      {
        "OfferId": "offer-def456"
      },
      {
        "OfferId": "offer-ghi789"
      }
    ]
  }
}

```

제안 세트 생성

CreateOfferSet 변경 유형을 사용하여 초안 상태에서 새 제안 세트 개체를 생성합니다. 이 단계는 제안 세트 수명 주기의 첫 번째 단계로, 구매자에게 릴리스하기 전에 나중에 제안 연결 및 메타데이터로 구성될 기본 엔터티를 설정합니다.

요청 구문

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferSet",
      "Entity": {
        "Type": "OfferSet@1.0"
      },
      "ChangeName": "CreateOfferSetChange",
      "DetailsDocument": {
        "Name": "Enterprise Security Solution Offer Set"
      }
    }
  ]
}

```

```
]
}
```

필드에 대한 정보를 제공하여 CreateOfferSet 변경 유형을 추가합니다.

- Entity (객체)(필수) - 생성 중인 엔터티 유형입니다.
 - Type (문자열)(필수) - 여야 합니다OfferSet@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - Name (문자열)(필수) - 사용자와 고객의 가독성을 높이기 위해 제안 세트와 연결된 이름입니다. 최소 길이: 1자. 최대 길이: 150자.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

처리가 성공하면는 접두사가 인 고유 식별자가 있는 초안 상태의 제안 세트를 AWS Marketplace 생성합니다offerset-.

제안 세트 정보 업데이트

UpdateInformation 변경 유형을 사용하여 이름 및 구매자 메모를 포함하여 기존 제안 세트의 세부 정보를 수정합니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
```



```
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "OfferSet@1.0",
        "Identifier": "offerset-abc123"
      },
      "DetailsDocument": {
        "Name": "Updated Enterprise Security Offer Set",
        "BuyerNotes": "Complete security solution including endpoint protection,
network monitoring, and professional services implementation"
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 UpdateInformation 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 엔터티입니다.
 - Identifier (문자열)(필수) - 제안 세트 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - 여야 합니다 OfferSet@1.0.
- DetailsDocument (객체)(필수) - 하나 이상의 필드를 제공해야 합니다.
 - Name (문자열)(선택 사항) - 구매자가 볼 수 있는 표시 이름입니다. 최소 길이: 1자. 최대 길이: 150자.
 - BuyerNotes (문자열)(선택 사항) - 구매자가 용도와 내용을 이해하는 데 도움이 되는 제안 세트에 대한 세부 정보입니다. 최소 길이: 1자. 최대 길이: 1,000자.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
```

}

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

제안을 제안 세트와 연결

AssociateOffers 변경 유형을 사용하여 여러 AWS Marketplace 제안을 제안 세트와 연결합니다. 이렇게 하면 제안과 제안 세트 간의 관계가 생성되어 제안 세트의 거래 가능한 구성의 일부가 됩니다.

작업은 멍등성이며 현재 상태에 관계없이 제안을 연결할 수 있으므로 개별 제안이 릴리스되기 전에 완전한 구성을 사용할 수 있습니다. 각 제안은 하나의 제안 세트에만 속할 수 있습니다.

Important

개별 제안 OfferSetId의는 변경할 수 없으며 제안 생성 중에만 설정할 수 있습니다. 를 사용하여 제안을 제안 세트와 연결 AssociateOffers하기 전에 먼저 개별 제안을 생성하고 제안 생성 OfferSetId 중어를 지정해야 합니다. 올바른가 없는 기존 제안을 포함해야 하는 경우 올바른가 OfferSetId 지정된 새 제안을 생성 OfferSetId해야 합니다.

필수 IAM 권한에 대한 자세한 내용은 섹션을 참조하세요 [제안 세트에 대한 IAM 권한](#).

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AssociateOffers",
      "Entity": {
        "Type": "OfferSet@1.0",
        "Identifier": "offerset-abc123"
      },
      "DetailsDocument": {
        "Offers": [
```

```

    {
      "OfferId": "offer-xyz789"
    },
    {
      "OfferId": "offer-def456"
    },
    {
      "OfferId": "offer-ghi123"
    }
  ]
}
]
}
]
}

```

필드에 대한 정보를 제공하여 AssociateOffers 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 엔터티입니다.
 - Identifier (문자열)(필수) - 제안 세트 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - 여야 합니다OfferSet@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - Offers (배열)(필수) - 연결할 AWS Marketplace 제안 목록입니다. 최소: 항목 1개. 최대: 항목 7개.
 - OfferId (문자열)(필수) - AWS Marketplace 제안의 식별자입니다. 최소 길이: 1자. 최대 길이: 36자.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

다음 오류는 AWS Marketplace 카탈로그 API의 AssociateOffers 작업과 관련이 있습니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
TOO_MANY_OFFERS	최대 7개의 제안을 제안 세트에 연결합니다.
INCOMPATIBLE_OFFER_SET_REFERENCE	모든 제안이이 제안 세트에 대해 특별히 생성되었는지 확인합니다.
INVALID_UPDATE_REQUEST	요청된 변경은 제안 세트가 릴리스된 후에는 수행할 수 없습니다.

제안 세트에서 제안 연결 해제

DisassociateOffers 변경 유형을 사용하여 AWS Marketplace 제안 세트에서 여러 제안을 제거합니다. 이 멍등성 작업은 제안의 OfferSetId 속성을 유지하면서 연결 관계를 제거합니다. 연결 해제된 제안은 동일한 제안 세트와 다시 연결될 때까지 구매자 검색에서 숨겨집니다. 다른 제안 세트와 연결할 수 없습니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "DisassociateOffers",
      "Entity": {
        "Type": "OfferSet@1.0",
```

```

    "Identifier": "offerset-abc123"
  },
  "DetailsDocument": {
    "Offers": [
      {
        "OfferId": "offer-xyz789"
      }
    ]
  }
}
]
}

```

필드에 대한 정보를 제공하여 DisassociateOffers 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 엔터티입니다.
 - Identifier (문자열)(필수) - 제안 세트 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - 여야 합니다OfferSet@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - Offers (배열)(필수) - 연결 해제할 제안 목록입니다. 최소: 항목 1개. 최대: 항목 7개.
 - OfferId (문자열)(필수) - AWS Marketplace 제안의 식별자입니다. 최소 길이: 1자. 최대 길이: 36자.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API 를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

다음 오류는 AWS Marketplace 카탈로그 API의 DisassociateOffers 작업에만 해당됩니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_UPDATE_REQUEST	요청된 변경은 제안 세트가 릴리스된 후에는 수행할 수 없습니다.

솔루션을 제안 세트와 연결

AssociateSolution 변경 유형을 사용하여 단일 AWS Marketplace 솔루션을 제안 세트와 연결합니다. 이렇게 하면 구매자가 제안 세트의 솔루션 연결을 검색하고 자세한 설명, 아키텍처 다이어그램 및 사용 사례 설명서와 같은 솔루션의 풍부한 마케팅 콘텐츠에 액세스할 수 있는 관계가 생성됩니다.

Note

이는 느슨하게 결합된 연결입니다. AWS Marketplace는 솔루션과 제안 세트 간에 일관성을 적용하지 않습니다. 솔루션에는 AWS Marketplace 제품, 제안 세트의 제품과 다른 제품 또는 겹치는 제품이 포함될 수 없습니다. 마케팅 및 검색 목적으로 소유한 솔루션을 연결할 수 있는 완전한 유연성이 있습니다.

필요한 IAM 권한에 대한 자세한 내용은 섹션을 참조하세요 [제안 세트에 대한 IAM 권한](#).

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AssociateSolution",
      "Entity": {
```

```

    "Type": "OfferSet@1.0",
    "Identifier": "offerset-abc123"
  },
  "DetailsDocument": {
    "SolutionId": "soln-xyz789"
  }
}
]
}

```

필드에 대한 정보를 제공하여 AssociateSolution 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 엔터티입니다.
 - Identifier (문자열)(필수) - 제안 세트 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - 여야 합니다 OfferSet@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - SolutionId (문자열)(필수) - 연결할 AWS Marketplace 솔루션의 식별자입니다. 최소 길이: 1자. 최대 길이: 50자. 패턴과 일치해야 합니다 `^soln-[A-Za-z0-9]+$`.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```

{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/ChangeSet/example123456789012abcdef"
}

```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API 를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

다음 오류는 AWS Marketplace 카탈로그 API의 AssociateSolution 작업에만 해당됩니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청

의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
INCOMPATIBLE_SOLUTION_ASSOCIATION	새 솔루션을 연결하기 전에 기존 솔루션의 연결을 해제합니다.
INVALID_UPDATE_REQUEST	요청된 변경은 제안 세트가 릴리스된 후에는 수행할 수 없습니다.

제안 세트에서 솔루션 연결 해제

DisassociateSolution 변경 유형을 사용하여 솔루션과 제안 세트 간의 연결을 제거합니다. 이렇게 하면 제안 세트의 거래 가능한 기능을 유지하면서 제안 세트에서 솔루션의 마케팅 콘텐츠가 제거됩니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "DisassociateSolution",
      "Entity": {
        "Type": "OfferSet@1.0",
        "Identifier": "offerset-abc123"
      },
      "DetailsDocument": {
        "SolutionId": "soln-xyz789"
      }
    }
  ]
}
```

필드에 대한 정보를 제공하여 DisassociateSolution 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 엔터티입니다.
 - Identifier (문자열)(필수) - 제안 세트 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - 여야 합니다OfferSet@1.0.
- DetailsDocument (객체)(필수) - 요청의 세부 정보입니다.
 - SolutionId (문자열)(필수) - 연결을 해제할 솔루션의 식별자입니다. 최소 길이: 1자. 최대 길이: 50자.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API 를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

다음 오류는 AWS Marketplace 카탈로그 API의 DisassociateSolution 작업과 관련이 있습니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 [섹션을 참조하세요](#)[변경 세트 작업](#).

오류 코드	오류 메시지
INVALID_UPDATE_REQUEST	요청된 변경은 제안 세트가 릴리스된 후에는 수행할 수 없습니다.

제안 세트 릴리스

ReleaseOfferSet 변경 유형을 사용하여 구매자가 제안 세트를 사용할 수 있도록 합니다. 릴리스되면 제안 세트가 초안에서 릴리스됨 상태로 전환되고,를 통해 검색할 수 있게 되며 AWS Marketplace, 관련 제안은 통합 패키지의 일부로만 찾을 수 있습니다.

릴리스된 제안 세트는 개별 제안과 제안 세트 자체의 두 가지 수준에서 알림을 생성합니다. 모든 개별 제안 알림에는 기존 알림 채널([이메일 알림](#) 및 [Amazon EventBridge 이벤트](#))의 OfferSetId가 포함됩니다. 제안 세트가 릴리스될 때와 같은 주요 이벤트에 대해 제안 세트 수준 알림이 전송됩니다.

요청 구문

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseOfferSet",
      "Entity": {
        "Type": "OfferSet@1.0",
        "Identifier": "offerset-abc123"
      },
      "DetailsDocument": {}
    }
  ]
}
```

필드에 대한 정보를 제공하여 ReleaseOfferSet 변경 유형을 추가합니다.

- Entity (객체)(필수) - 업데이트 중인 엔터티입니다.
 - Identifier (문자열)(필수) - 제안 세트 ID입니다. 자세한 내용은 [식별자](#) 단원을 참조하십시오.
 - Type (문자열)(필수) - 여야 합니다OfferSet@1.0.
- DetailsDocument (객체)(필수) -이 작업의 객체가 비어 있습니다.

응답 구문

요청에 대한 변경 세트가 생성됩니다. 이 요청에 대한 응답은 변경 세트에 ChangeSetArn 대한 ChangeSetId 및를 제공하며 다음과 같습니다.

```
{
  "ChangeSetId": "example123456789012abcdef",
  "ChangeSetArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
ChangeSet/example123456789012abcdef"
}
```

변경 요청이 대기열에 추가되고 처리됩니다. 여기에는 AWS Marketplace 지침을 충족하는지 확인하기 위한 정보 검증이 포함됩니다. 검증 프로세스는 몇 분 정도 걸릴 수 있습니다.

AWS Marketplace 관리 포털을 통해 또는 API 작업을 사용하여 카탈로그 [DescribeChangeSet](#) API를 통해 직접 요청 상태를 확인할 수 있습니다.

비동기 오류

다음 오류는 AWS Marketplace 카탈로그 API의 ReleaseOfferSet 작업에만 해당됩니다. 이러한 오류는 변경 세트가 처리된 DescribeChangeSet 후를 호출할 때 반환됩니다. 를 사용하여 변경 요청의 상태를 가져오는 DescribeChangeSet 방법에 대한 자세한 내용은 섹션을 참조하세요 [변경 세트 작업](#).

오류 코드	오류 메시지
누락_구매자_노트	제안 세트를 릴리스하기 전에 BuyerNotes를 제공합니다.
누락_꺼짐	제안을 릴리스하기 전에 제안 세트에 두 개 이상의 제안을 연결합니다.
TOO_MANY_OFFERS_PER_PRODUCT	제품당 하나의 제안만 제안 세트에 연결합니다.
일관되지 않은_오퍼_통화_코드	연결된 모든 제안에 일치하는 CurrencyCode가 있는지 확인합니다.
INCONSISTENT_OFFER_AVAILABILITY_END_DATE	연결된 모든 제안에 일치하는 AvailabilityEndDate가 있는지 확인합니다.
일관되지 않은_제안_대상_규칙	연결된 모든 제안에 일치하는 TargetingRule이 있는지 확인합니다.
INCOMPATIBLE_OFFER_TARGETING_RULE	구매자 대상 제안만 제안 세트에 연결할 수 있습니다.

오류 코드	오류 메시지
EXPIRED_OFFERS	제안 세트에서 만료된 제안을 연결 해제하거나 AvailabilityEndDates를 확장합니다.
DRAFT_OFFERS	제안 세트를 릴리스하기 전에 모든 관련 제안을 릴리스합니다.
INCOMPATIBLE_SOLUTION_STATE	제한적 또는 퍼블릭 상태의 솔루션을 제공합니다.
INVALID_UPDATE_REQUEST	요청된 변경은 제안 세트가 릴리스된 후에는 수행할 수 없습니다.

제안 세트 수명 주기

제안 세트 수명 주기는 다음 단계로 구성됩니다.

생성 단계

CreateOfferSet 변경 유형을 사용하여 제안 세트 개체를 초안 상태로 생성합니다. 제안 세트는 고유 식별자를 수신하고 초안 상태로 시작됩니다.

구성 단계

구성 중에 다음을 수행합니다.

- UpdateInformation (이름, 구매자 메모)를 사용하여 메타데이터 추가 또는 업데이트
- 개별 비공개 제안 생성, 제안 생성 OfferSetId 중 지정
- 를 사용하여 제안을 제안 세트와 연결 AssociateOffers
- 선택적으로 향상된 마케팅 콘텐츠를 AssociateSolution 위해를 사용하여 솔루션 연결

Important

개별 제안OfferSetId의는 변경할 수 없으며 제안 생성 중에만 설정할 수 있습니다. 를 사용하여 제안을 제안 세트와 연결AssociateOffers하기 전에 먼저 개별 제안을 생성하고 제안 생성 OfferSetId 중예를 지정해야 합니다. 올바른가 없는 기존 제안을 포함해야 하는 경우 올바른가 OfferSetId 지정된 새 제안을 생성OfferSetId해야 합니다.

검증 단계

제안 세트를 릴리스하기 전에 다음을 확인하세요.

- 제안 세트에는 2~7개의 제안이 포함됩니다.
- 모든 관련 제안은 다음과 같아야 합니다.
 - 릴리스됨 상태
 - 활성
 - 동일한 통화 사용
 - 동일한 구매자 AWS 계정 ID(들)를 대상으로 지정
 - 만료 날짜가 동일함

릴리스 단계

준비가 되면 `ReleaseOfferSet` 변경 유형을 사용하여 제안 세트를 초안에서 릴리스됨 상태로 전환합니다. 이렇게 하면 구매자가 제안 세트를 검색할 수 있습니다 AWS Marketplace.

릴리스 후 관리

만료 관리:

- 제안 세트의 유효 만료는 모든 관련 제안 중 가장 빠른 만료 날짜로 계산됩니다.
- 기존 제안 관리 기능(`UpdateAvailability` 변경 유형)을 사용하여 개별 제안 만료 날짜를 수정할 수 있습니다.

제안 세트 수정 사항:

구매자가 릴리스된 제안 세트에 대한 변경을 요청하면 재생성 워크플로를 사용합니다.

1. 새 제안 세트 엔터티 생성
2. 변경이 필요하지 않은 제안의 경우 AWS Marketplace Management Portal을 통해 기존 제안을 복제합니다.
3. 수정이 필요한 제안의 경우 요청된 변경 사항이 포함된 새 제안을 생성하고 새를 지정합니다. `OfferSetId`
4. 를 사용하여 모든 제안(복제 및 신규)을 새 제안 세트와 연결 `AssociateOffers`
5. 를 사용하여 새 제안 세트 릴리스 `ReleaseOfferSet`

6. 관련 제안의 가용성 종료일을 설정하여 원래 제안 세트 만료

제안 세트에 대한 IAM 권한

AWS Marketplace 카탈로그 API를 사용하여 제안 세트를 사용하려면 특정 IAM 권한이 필요합니다. 이 섹션에서는 제안 세트 작업 및 교차 엔터티 권한 부여 요구 사항에 필요한 권한을 설명합니다.

필수 IAM 작업

제안 세트 작업에 필요한 IAM 작업은 다음과 같습니다.

- `aws-marketplace:StartChangeSet` - 생성, 업데이트, 연결 및 릴리스 작업을 포함한 모든 제안 세트 변경 유형에 필요합니다.
- `aws-marketplace:DescribeChangeSet` - 변경 세트 실행의 상태 및 결과를 확인하는 데 필요합니다.
- `aws-marketplace:ListEntities` - 계정의 제안 세트를 나열하는 데 필요합니다.
- `aws-marketplace:DescribeEntity` - 제안 세트에 대한 세부 정보를 검색하는 데 필요합니다.

리소스 권한

다음 ARN 패턴을 사용하여 특정 리소스에 대한 권한을 부여합니다.

- 제안 세트 - `arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/OfferSet/*`
- 개별 제안 - `arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Offer/*`
- 솔루션 - `arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/*`

보다 제한적인 권한을 얻으려면 와일드카드(*)를 특정 엔터티 식별자로 바꿉니다.

교차 엔터티 권한 부여

일부 제안 세트 작업에는 제안 세트와 연결 중인 엔터티 모두에 대한 권한이 필요합니다.

AssociateOffers

- `aws-marketplace:StartChangeSet` 제안 세트 엔터티에 대한 권한

- `aws-marketplace:StartChangeSet` 연결된 각 개별 제안 엔터티에 대한 `AssociateWithOfferSet` 변경 유형이 있는 권한

AssociateSolution

- `aws-marketplace:StartChangeSet` 제안 세트 엔터티에 대한 권한
- `aws-marketplace:StartChangeSet` 연결 중인 솔루션 엔터티에 대한 `AssociateWithOfferSet` 변경 유형이 있는 권한

Note

제안 또는 솔루션에 대한 `DescribeEntity` (읽기 전용) 권한만 있으면 연결 작업에 충분하지 않습니다. 제안 세트와 연결된 엔터티에 대한 `AssociateWithOfferSet` 변경 유형에 대한 `StartChangeSet` 권한이 있어야 합니다.

예제 IAM 정책

기본 제안 세트 관리

이 정책은 교차 엔터티 연결 없이 기본 제안 세트 작업에 대한 권한을 제공합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:StartChangeSet",
        "aws-marketplace:DescribeChangeSet",
        "aws-marketplace:ListEntities",
        "aws-marketplace:DescribeEntity"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/OfferSet/*"
      ]
    }
  ]
}
```

AssociateOffers에 대한 교차 엔터티 권한

이 정책은 AssociateOffers에 필요한 특정 권한 및 조건을 보여줍니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:StartChangeSet",
      "Resource": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
OfferSet/*",
      "Condition": {
        "StringEquals": {
          "catalog:ChangeType": ["AssociateOffers"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:DescribeEntity",
      "Resource": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Offer/
*"
    },
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:StartChangeSet",
      "Resource": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Offer/
*",
      "Condition": {
        "StringEquals": {
          "catalog:ChangeType": ["AssociateWithOfferSet"]
        }
      }
    }
  ]
}
```

보다 제한적인 권한을 얻으려면 와일드카드(*)를 특정 엔터티 식별자로 바꿉니다.

AssociateSolution에 대한 교차 엔터티 권한

이 정책은 AssociateSolution에 필요한 특정 권한 및 조건을 보여줍니다.


```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:StartChangeSet",
      "Resource": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
OfferSet/*",
      "Condition": {
        "StringEquals": {
          "catalog:ChangeType": ["AssociateSolution"]
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:DescribeEntity",
      "Resource": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
Solution/*"
    },
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:StartChangeSet",
      "Resource": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
Solution/*",
      "Condition": {
        "StringEquals": {
          "catalog:ChangeType": ["AssociateWithOfferSet"]
        }
      }
    }
  ]
}
```

보다 제한적인 권한을 얻으려면 와일드카드(*)를 특정 엔터티 식별자로 바꿉니다.

카탈로그 데이터를 사용하여 인터페이스 구축

AWS Marketplace Discovery API를 사용하여 프로그래밍 방식으로 카탈로그에 액세스하고 자체 플랫폼에 제품 및 요금 정보를 표시할 수 있습니다. 이 섹션에서는 Discovery API를 사용하여 매장, 파트너 포털, 고객이 탐색할 수 있는 통합 영업 경험과 같은 사용자 지정 인터페이스를 구축하는 방법을 설명합니다.

아키텍처 개요

일반적인 통합은 다음 패턴을 따릅니다.

1. 경험 찾아보기 - SearchListings를 호출하여 검색 텍스트 및 필터를 기반으로 목록 요약SearchFacets을 검색하고 범주 탐색 및 필터 옵션을 개수로 채웁니다.
2. 제품 세부 정보 페이지 - GetListing 또는 호출GetProduct하여 전체 제품 세부 정보, 미디어 및 리뷰를 표시합니다.
3. 요금 표시 - ListPurchaseOptions, GetOffer및 호출GetOfferTerms하여 요금 옵션을 표시합니다.

찾아보기 경험 구축

SearchListings를 사용하여 검색 텍스트 및 필터를 기반으로 목록 요약을 검색하고 동적 필터 탐색SearchFacets을 빌드합니다. 패킷에는 범주, 이행 유형, 요금 모델, 요금 단위, 게시자 및 배포 상태가 포함됩니다. 각 패킷 값에는 필터 옵션과 함께 표시할 수 있는 일치하는 목록 수가 포함됩니다.

```
import boto3

client = boto3.client('marketplace-discovery', region_name='us-east-1')

# Get facet values for building filter navigation
facets = client.search_facets(
    facetTypes=['CATEGORY', 'FULFILLMENT_OPTION_TYPE', 'PRICING_MODEL']
)

for facet_type, values in facets['listingFacets'].items():
    print(f"\n{facet_type}:")
    for facet in values:
        print(f"  {facet['displayName']} ({facet['count']})")

# Search listings with filters applied from user selections
response = client.search_listings(
    searchText='machine learning',
    filters=[
        {
            'filterType': 'CATEGORY',
            'filterValues': ['Machine Learning']
        },
        {
```

```

        'filterType': 'PRICING_MODEL',
        'filterValues': ['USAGE']
    }
],
sortBy='RELEVANCE',
maxResults=25
)

for listing in response['listingSummaries']:
    print(f"{listing['listingName']} - {listing['shortDescription']}")

```

빌드 제품 세부 정보 페이지

GetListing 또는 GetProduct를 사용하여 전체 제품 세부 정보, 미디어 및 리뷰를 표시합니다. 배지, 범주 및 요금 요약은 포함된 목록 개요가 필요할 GetListing 때 사용합니다. 설명, 하이라이트, 미디어와 같은 자세한 제품 정보가 필요한 GetProduct 경우를 사용합니다. 를 호출 ListFulfillmentOptions하여 특정 제품의 배포 옵션을 검색합니다.

```

def get_product_page_data(listing_id, product_id):
    """Retrieve data for a product detail page."""

    # Use GetListing for listing-level overview
    listing = client.get_listing(listingId=listing_id)

    # Or use GetProduct for detailed product information
    product = client.get_product(productId=product_id)

    # Get deployment options for the product
    fulfillment = client.list_fulfillment_options(productId=product_id)

    return {
        'listing': listing,
        'product': product,
        'fulfillmentOptions': fulfillment
    }

```

요금 정보 표시

ListPurchaseOptions를 사용하여 제품에 사용할 수 있는 제안을 찾는 다음 GetOfferGetOfferTerms를 사용하여 자세한 요금을 검색합니다. 고객이 옵션을 평가하는 데 도움이 되는 요금 모델, 요금 카드 및 기간 세부 정보를 표시합니다.

```
def get_pricing_for_product(product_id):
    """Retrieve pricing options for display."""
    options = client.list_purchase_options(
        filters=[{
            'filterType': 'PRODUCT_ID',
            'filterValues': [product_id]
        }]
    )

    pricing = []
    for option in options.get('purchaseOptions', []):
        for entity in option['associatedEntities']:
            offer = client.get_offer(
                offerId=entity['offer']['offerId']
            )
            terms = client.get_offer_terms(
                offerId=entity['offer']['offerId']
            )
            pricing.append({
                'option': option,
                'offer': offer,
                'terms': terms
            })

    return pricing
```

모범 사례

- 캐시 응답 - 제품 및 요금 데이터가 자주 변경되지 않습니다. API 응답을 15~60분 동안 캐시하여 API 호출을 줄이고 페이지 로드 시간을 개선합니다.
- 페이지 매김 처리 - 페이지 매김 작업에서 모든 결과를 검색nextToken하는 데 사용합니다. 페이지 매김 토큰은 24시간 후에 만료됩니다.
- 재시도 로직 구현 - ThrottlingException (HTTP 429) 응답에 지수 백오프를 사용합니다.
- 표시 속성 - 플랫폼에 AWS Marketplace 데이터를 표시할 때 [AWS 상표 지침](#)을 따릅니다.
- API 호출 최소화 - 요약 데이터에 SearchListings를 사용하고 사용자가 특정 제품으로 이동할 때만 세부 정보 APIs(GetProduct, GetOffer, GetOfferTerms)를 호출합니다.

 Note

Discovery API는 SaaS AWS Marketplace, AI 에이전트 및 도구, AMI, 컨테이너, 기계 학습 모델을 포함한 모든 제품 유형을 지원합니다. 인터페이스에 카탈로그의 전체 너비가 표시될 수 있습니다.

AWS Marketplace API를 구매자로 사용

다음 섹션에서는 API를 구매자로 사용하는 방법에 대한 정보를 제공합니다.

주제

- [제품 및 요금 알아보기](#)
- [구매자로서 계약 API 작업](#)
- [AWS MarketplaceAPI를 사용하여 프라이빗 마켓플레이스 작업](#)

제품 및 요금 알아보기

AWS Marketplace Discovery API를 사용하여 카탈로그를 프로그래밍 방식으로 검색하고, 자세한 제품 정보를 검색하고, 공개 및 비공개 제안을 포함한 요금 세부 정보에 액세스할 수 있습니다. 이 섹션에서는 제품 검색 및 요금 평가를 위한 일반적인 워크플로에 대해 설명합니다.

카탈로그 찾아보기

SearchFacets를 사용하여 사용 가능한 필터 값을 검색하고 브라우징 경험을 구축합니다. 패킷에는 범주, 요금 모델, 이행 유형, 게시자 및 배포 상태가 포함됩니다. 각 패킷 값에는 일치하는 목록 수가 포함됩니다.

```
response = client.search_facets(
    facetTypes=['CATEGORY', 'PRICING_MODEL', 'FULFILLMENT_OPTION_TYPE']
)

for facet_type, values in response['listingFacets'].items():
    print(f"\n{facet_type}:")
    for facet in values:
        print(f"  {facet['displayName']} ({facet['count']} listings)")
```

목록 검색

키워드 및 필터별로 목록을 찾으려면 SearchListings를 사용합니다. 텍스트 검색을 패킷 기반 필터와 결합하고 관련성 또는 고객 등급을 기준으로 결과를 정렬할 수 있습니다.

```
results = client.search_listings(
    searchText='computer vision',
```

```

    filters=[
        {'filterType': 'CATEGORY', 'filterValues': ['machine-learning']},
        {'filterType': 'FULFILLMENT_OPTION_TYPE', 'filterValues': ['SAAS']}
    ],
    sortBy='RELEVANCE',
    maxResults=25
)

print(f"Found {results['totalResults']} listings")
for listing in results['listingSummaries']:
    print(f" {listing['listingName']} -- {listing['publisher']['displayName']}")

```

목록 세부 정보 가져오기

GetListing를 사용하여 설명, 범주, 배지, 요금 모델 요약, 리뷰, 관련 제품 및 제안 등 목록의 전체 세부 정보를 검색합니다.

```

listing = client.get_listing(listingId='listing-saas-abc123')

print(f"Name: {listing['listingName']}")
print(f"Publisher: {listing['publisher']['displayName']}")
print(f"Pricing: {[p['displayName'] for p in listing['pricingModels']]}")
print(f"Rating: {listing['reviewSummary']['reviewSourceSummaries'][0]
['averageRating']}")

```

제품 세부 정보 및 이행 옵션 가져오기

GetProduct를 사용하여 설명, 하이라이트, 범주, 판매자 참여 등 자세한 제품 정보를 가져옵니다. ListFulfillmentOptions를 사용하여 제품을 배포하는 방법(AMI, SaaS, 컨테이너, Helm 등)을 확인합니다.

```

product = client.get_product(productId='prod-gitpod-enterprise')

print(f"Description: {product['shortDescription']}")
print(f"Deployed on AWS: {product['deployedOnAws']}")

fulfillment = client.list_fulfillment_options(productId='prod-gitpod-enterprise')
for option in fulfillment['fulfillmentOptions']:
    # Each option is a union -- check which variant is present
    for variant_name, variant_data in option.items():
        print(f" Deploy as: {variant_data['fulfillmentOptionType']}")

```

제품에 사용 가능한 제안 찾기

ListPurchaseOptions를 사용하여 제품에 사용할 수 있는 모든 구매 옵션(제안 및 제안 세트)을 찾습니다. 제품 ID, 판매자, 구매 옵션 유형, 가시성 범위 및 가용성 상태를 기준으로 필터링할 수 있습니다.

```
options = client.list_purchase_options(
    filters=[{
        'filterType': 'PRODUCT_ID',
        'filterValues': ['prod-gitpod-enterprise']
    }]
)

for option in options['purchaseOptions']:
    print(f" {option['purchaseOptionType']}: {option.get('purchaseOptionName', 'Public offer')}")
    print(f" Seller: {option['sellerOfRecord']['displayName']}")
```

자세한 요금 조건 가져오기

GetOfferTerms를 사용하여 제안의 전체 상용 구조를 검색합니다. 각 제안에는 요금 조건(사용 기준, 고정 선결제, 구성 가능한 선결제, BYOL, 무료 평가판), 법률 조건, 결제 일정, 유효 기간, 지원 조건 및 갱신 조건과 같은 하나 이상의 조건이 포함됩니다.

```
terms = client.get_offer_terms(offerId='offer-abc123')

for term in terms['offerTerms']:
    # Each term is a union -- check which type is present
    for term_type, term_data in term.items():
        print(f" Term: {term_data['type']}")
        if 'rateCards' in term_data:
            for card in term_data['rateCards']:
                for rate in card.get('rateCard', []):
                    print(f" {rate['displayName']}: ${rate['price']}/
{rate['unit']}")
```

제안 세트 세부 정보 가져오기

에서 반환한 구매 옵션purchaseOptionType의 ListPurchaseOptions가 OFFERSET인 경우 다중 제품 솔루션에 대한 비공개 제안의 그룹화된 컬렉션을 나타냅니다. GetOfferSet를 사용하여 관련 제

품 및 제안, 판매자 기록, 가용성 날짜, 구매자 메모 등 제안 세트의 세부 정보를 검색합니다. 그런 다음 세트 내의 각 개별 제안에 GetOfferTerms 대해 GetOffer 및를 호출할 수 있습니다.

```
# Find offer sets for a product
options = client.list_purchase_options(
    filters=[
        {'filterType': 'PRODUCT_ID', 'filterValues': ['prod-abc123']},
        {'filterType': 'PURCHASE_OPTION_TYPE', 'filterValues': ['OFFERSET']}
    ]
)

for option in options.get('purchaseOptions', []):
    offer_set_id = option['purchaseOptionId']
    offer_set = client.get_offer_set(offerSetId=offer_set_id)

    print(f"Offer set: {offer_set.get('offerSetName', offer_set_id)}")
    print(f"Seller: {offer_set['sellerOfRecord']['displayName']}")

# Review each product and offer in the set
for entity in offer_set['associatedEntities']:
    print(f" Product: {entity['product']['productName']}")
    print(f" Offer: {entity['offer']['offerId']}")

# Get terms for each offer in the set
terms = client.get_offer_terms(offerId=entity['offer']['offerId'])
for term in terms['offerTerms']:
    for term_type, term_data in term.items():
        print(f" Term: {term_data['type']}")
```

비공개 제안 요금에 액세스

비공개 제안 요금에 액세스하려면 호출 IAM 보안 주체에 적절한 권한이 있어야 합니다. 비공개 제안에는 협상된 요금, 사용자 지정 조건 및 조직별 제안이 포함됩니다. 값이 인 VISIBILITY_SCOPE 필터를 사용하여 계정PRIVATE에서 사용할 수 있는 비공개 제안을 ListPurchaseOptions 찾습니다.

```
private_options = client.list_purchase_options(
    filters=[
        {'filterType': 'PRODUCT_ID', 'filterValues': ['prod-gitpod-enterprise']},
        {'filterType': 'VISIBILITY_SCOPE', 'filterValues': ['PRIVATE']}
    ]
)
```

조달 도구와 통합

Discovery API는 엔터프라이즈 조달 플랫폼과 통합되도록 설계되었습니다. 일반적인 통합 패턴은 다음과 같습니다.

- AWS Marketplace 카탈로그를 조달 도구의 제품 카탈로그에 통합
- 구매 요청 워크플로 내에 제품 세부 정보 및 요금 표시
- AWS Marketplace 제안 간 비교 쇼핑 활성화

Note

Discovery API는 통합 조달 경험을 구축하는 데 필요한 제품 및 요금 데이터를 제공합니다. 기존 조달 워크플로와 결합하여 구매 결정을 간소화합니다.

구매자로서 계약 API 작업

계약은 제안자와 수락자(일반적으로 구매자)를 포함하여 두 당사자를 구속하고 이들 간에 적용되는 이용 약관을 정의하는 문서입니다.

구매자는 견적을 생성하는 `AgreementRequest`를 생성합니다. 이 견적에는 구매자의 구매 결정에 중요한 계약 기간 동안 발생할 예상 요금과 같은 모든 관련 정보가 포함되어 있습니다. 구매자가 견적에 만족하면 만료되기 전에 `AgreementRequest`를 수락할 수 있습니다. 계약은 `AgreementRequest`를 수락하면 생성됩니다. 이 프로세스 중에 구매자 관리자 또는 구매자 페르소나는 구매 주문을 요금에 연결할 수 있습니다. 마지막으로 유효한 계약의 이용 약관에 따라 구매자에게 인보이스가 발행되고 제품을 사용할 수 있는 라이선스가 부여됩니다.

사전 조건: 제품 및 제안 검색

계약을 생성하기 전에 AWS Marketplace Discovery API를 사용하여 제품을 검색하고 `agreementProposalId`를 구성하는 데 필요한 `pricingModel`, 및 기간 세부 정보를 가져옵니다 `CreateAgreementRequest`.

단계	Discovery API 작업	출력	사용 대상
1	ListPurchaseOptions	<code>offerId</code>	제품에 사용할 수 있는 제안 찾기

단계	Discovery API 작업	출력	사용 대상
2	GetOffer	agreementProposalId , pricingModel	구성 agreementProposalIdentifier
3	GetOfferTerms	기간 IDs 및 요금 세부 정보	구성 requestedTerms

제품 및 요금 검색에 대한 자세한 내용은 섹션을 참조하세요 [제품 및 요금 알아보기](#).

수락자는 이 API를 사용하여 다음 작업을 수행할 수 있습니다.

작업	설명	작업(들)	의도
다음표 생성	수락자는 구매자의 구매 결정에 중요한 계약 기간 동안 발생하는 요금과 같은 모든 관련 정보가 포함된 견적을 생성할 수 있습니다.	CreateAgreementRequest	신규
제안 수락	수락자는 새 계약을 생성하는 제안자가 제안한 조건을 수락할 수 있습니다. 이 수락에는 수량 또는 기간 선택, 구매 주문 추가 등과 같은 특정 조건에 대한 파라미터 전달이 포함될 수 있습니다. 인수자는 향후 제품 사용이 시작되는 새 계약을 생성할 수도 있습니다. 계약 서명 날짜는 제안이 수락되고 계약이 생성되는 시점입니다. 계약 시작 날짜는 제품 사용이 시작되는 미래 날짜입니다. 라이선스/권한이 활성화된 날짜입니다. 사용 권한의 최신 상태를 검색하려면 GetAgreementEntitlements API를 참조하세요.	CreateAgreementRequest , AcceptAgreementRequest	신규
기존 계약 교체	수락자는 계약에 대한 중기 업그레이드를 수행하여 더 유리한 조건으로 전환하거나 계약에 대한 판매자를 전환할 수 있습니다. 이 작업은 입력으로 전달된 기존	CreateAgreementRequest , AcceptAgr	REPLACE

작업	설명	작업(들)	의도
	계약을 종료하고 새 계약을 생성합니다. 이 작업은 논리적으로 CANCEL에 이어 새 계약과 동일하지만 수락자의 권한 연속성을 보장하며 수락자는 권한 없이 남지 않습니다.	reementRequest	
기존 계약 수정	수락자는 수락된 조건에 대한 구성만 수정할 수 있습니다. 예를 들어 수정 후 가격 변경으로 인해 환불이 이루어지지 않는 한 자동 갱신을 활성화 또는 비활성화하거나 구매한 수량을 수정할 수 있습니다. 계약이 시작되지 않은 경우에만 요금 수정을 지원합니다. 참고: 제안자는 pay-as-you-go 요금 기간의 가격을 수정할 수 있습니다. 가격이 인상되면 구매자에게 가격 인상에 대한 알림이 전송된 후 90 일이 걸립니다. 요금에 따른 모든 지불액 감소는 즉시 적용됩니다.	CreateAgreementRequest , AcceptAgreementRequest	수정
자동 갱신 활성화 또는 비활성화	판매자가 갱신 제안 조건을 활성화한 경우 수락자는 계약에서 자동 갱신 플래그를 켜거나 끌 수 있습니다. 이 활성화된 경우 갱신 계약은 계약 서비스에서 최신 제안 개정을 사용하여 원래 계약의 만료 날짜에 생성됩니다. 생성된 본 계약의 시작일은 원래 계약의 종료일과 동일합니다.	CreateAgreementRequest , AcceptAgreementRequest	수정
계약 취소	수락자는 사용 계약을 취소할 수 있습니다. 다른 모든 경우 구매자는 판매자에게 문의하여 취소를 시작해야 합니다. 계약을 취소하면 라이선스와 권한이 비활성화됩니다.	CancelAgreement	해당 사항 없음

작업	설명	작업(들)	의도
계약 검색	수락자는 수락자로 참여한 모든 계약에 대해 검색을 수행할 수 있습니다 AWS Marketplace. 검색은 기본 계약 정보가 포함된 계약 목록을 반환합니다.	SearchAgreements	해당 사항 없음
계약 설명	수락자는 제안자, 수락자, 시작 날짜 및 종료 날짜와 같은 계약에 대한 세부 정보를 볼 수 있습니다.	DescribeAgreement	해당 사항 없음
계약 조건 가져오기	수락자는 수락자로 참여한 계약의 조건에 대한 세부 정보를 얻을 수 있습니다.	GetAgreementTerms	해당 사항 없음
계약 권한 가져오기	수락자는 계약과 관련된 권한의 상태 및 세부 정보에 대한 계약 수준 보기를 얻을 수 있습니다. 예를 들어 권한 부여가 진행 중인지 또는 거부되었는지(그렇다면 어떤 이유로) 고객에게 부여되는 권한은 무엇입니까?	GetAgreementEntitlements	해당 사항 없음
등록 토큰 가져오기	등록 토큰은 수락자가 제안자로 계정을 설정하는 데 필요한 수명이 짧은 토큰입니다. 이 토큰은 외부에서 측정된 차원 유형과 권한이 부여된 차원 유형 모두에 사용됩니다. 토큰은 생성 후 30분 동안만 유효하며 현재 SaaS 구매 계약에만 적용됩니다.	GetAgreementEntitlements	해당 사항 없음
계약 요금 나열	수락자는 계약과 관련된 요금 및 구매 주문 세부 정보를 볼 수 있습니다.	ListAgreementCharges	해당 사항 없음
구매 주문 업데이트	수락자는 제품을 구독한 후 구매 주문 번호를 추가할 수 있습니다. 구매 주문이 요금과 연결되면 해당 요금에 대해 생성된 인보이스에 구매 주문 번호가 포함됩니다.	UpdatePurchaseOrders	해당 사항 없음

작업	설명	작업(들)	의도
취소 요청 나열	수락자는 자신이 참여하는 계약에 대한 모든 취소 요청을 나열할 수 있습니다. 목록은 계약, 상태 및 기타 기준에 따라 필터링할 수 있습니다.	ListAgreementCancellationRequests	해당 사항 없음
취소 요청 세부 정보 가져오기	수락자는 판매자(제안자)가 시작한 상태, 타임스탬프 및 사유 코드를 포함한 특정 취소 요청에 대한 세부 정보를 검색할 수 있습니다.	GetAgreementCancellationRequest	해당 사항 없음
취소 요청 수락	수락자는 판매자(제안자)가 활성 계약에 대해 시작한 취소 요청을 승인할 수 있습니다. 승인 후 계약 취소 워크플로가 비동기적으로 실행되고 계약 상태가 취소됨으로 변경됩니다. 참고: 취소 요청을 승인하면 계약이 취소되므로 사용자에게 CancelAgreement 권한도 필요합니다.	AcceptAgreementCancellationRequest, CancelAgreement	해당 사항 없음
취소 요청 거부	수락자는 판매자(제안자)가 시작한 취소 요청을 거부할 수 있습니다. 거부 후 계약은 활성 상태로 유지되고 취소 요청은 터미널 상태가 됩니다. 판매자는 필요한 경우 새 취소 요청을 생성할 수 있습니다.	RejectAgreementCancellationRequest	해당 사항 없음
결제 요청 나열	수락자는 자신이 참여하는 계약에 대한 모든 결제 요청을 나열할 수 있습니다. 목록은 계약, 상태 및 기타 기준에 따라 필터링할 수 있습니다.	ListAgreementPaymentRequests	해당 사항 없음
결제 요청 세부 정보 가져오기	수락자는 판매자(제안자)가 시작한 상태, 타임스탬프 및 관련 요금을 포함한 특정 결제 요청에 대한 세부 정보를 검색할 수 있습니다.	GetAgreementPaymentRequest	해당 사항 없음

작업	설명	작업(들)	의도
결제 요청 수락	수락자는 활성 계약에 대해 판매자(제안자)가 시작한 결제 요청을 승인할 수 있습니다.	AcceptAgreementPaymentRequest	해당 사항 없음
결제 요청 거부	수락자는 판매자(제안자)가 시작한 결제 요청을 거부할 수 있습니다. 거부 후 결제 요청은 터미널 상태가 됩니다. 판매자는 필요한 경우 새 결제 요청을 생성할 수 있습니다.	RejectAgreementPaymentRequest	해당 사항 없음

다음표 생성

를 사용하여 다음표를 CreateAgreementRequest 생성합니다. 다음을 제공해야 합니다.

- agreementProposalIdentifier - [GetOffer](#) 응답 agreementProposalId의 입니다.
- requestedTerms - [GetOfferTerms](#) 응답의 용어로 구성됩니다. 각 용어에는이 필요하며id, 일부 용어에는 추가이 필요합니다configuration. [requestedTerms 구성](#)을(를) 참조하세요.

```
response = client.create_agreement_request(
    agreementProposalIdentifier='at-edhtjnbilupjv3xqbphom77y',
    intent='NEW',
    requestedTerms=[
        {'id': 'term-legal-abc123'},
        {'id': 'term-validity-def456'},
        {
            'id': 'term-configurable-pricing-789',
            'configuration': {
                'configurableUpfrontPricingTermConfiguration': {
                    'selectorValue': 'P12M',
                    'dimensions': [
                        {'dimensionKey': 'Users', 'dimensionValue': 50}
                    ]
                }
            }
        },
        {
            'id': 'term-renewal-012',
```

```

        'configuration': {
            'renewalTermConfiguration': {'enableAutoRenew': True}
        }
    ]
)

print(f"Agreement Request ID: {response['agreementRequestId']}")
for charge in response['chargeSummary']['expectedCharges']:
    print(f"  Charge ID: {charge['id']}, Amount: ${charge['amount']}")

```

requestedTerms 구성

를 구성하려면 다음 두 가지 사항을 결정requestedTerms해야 합니다.

1. 포함할 용어 - 제안의 요금 모델에 따라 결정됩니다. [요금 모델별 필수 조건](#)을(를) 참조하세요.
2. 구성이 필요한 용어 - 대부분의 용어에는 만 필요하지id만 세 가지 용어 유형에는 추가 구매자 제공 구성이 필요합니다. [용어 구성](#)을(를) 참조하세요.

요금 모델별 필수 조건

[GetOffer](#) 응답의 pricingModel.[PricingModel](#) 필드에 따라 포함해야 하는 용어가 결정됩니다CreateAgreementRequest.

제안에 있는 경우 모든 계약에 LegalTerm, SupportTerm, 및 조건을 포함해야 합니다ValidityTermRenewalTerm.

Note

FreeTrialPricingTerm는 제품당 한 번만 수락할 수 있습니다. 구매자가 이미 무료 평가판을 사용한 경우에서 오류를 CreateAgreementRequest 반환합니다.

계약

[GetOfferTerms](#)에서 반환하는 모든 용어는 단일에서 필요합니다CreateAgreementRequest.

사용량

계약 생성은 제안에 있는 조건에 따라 달라집니다.

1. 첫 번째 계약 - 필수 조건과 함께 UsageBasedPricingTerm 및/또는 RecurringPaymentTerm (제안에 포함된 내용에 따라 다름)를 수락합니다. 이렇게 하면 제품 사용 권한이 부여됩니다.
2. 후속 계약(들)(선택 사항) - 제안에 ConfigurableUpfrontPricingTerm, FixedUpfrontPricingTerm 또는 포함된 경우 선택적으로 연간 구매에 대한 별도의 계약을 생성하여 할인 요금을 받을 PaymentScheduleTerm 수 있습니다.

Note

두 계약을 모두 생성하고 나중에 취소하려는 경우 먼저 후속 계약()을 취소한 다음 첫 번째 계약(UsageBasedPricingTerm 및/또는 ConfigurableUpfrontPricingTerm)을 취소합니다. RecurringPaymentTerm.

BYOL

필수 조건과 ByolPricingTerm 함께 수락합니다.

용어 구성

대부분의 용어에는 [GetOfferTerms](#)id의 만 필요합니다. 다음 용어 유형은 configuration 외에도 구매자가 제공해야 합니다. id

용어 유형	구성	구매자가 제공하는 내용
ConfigurableUpfrontPricingTerm	ConfigurableUpfrontPricingTermConfiguration	계약 기간 및 차원당 단위 수
RenewalTerm	RenewalTermConfiguration	계약 만료 시 자동 갱신 여부
VariablePaymentTerm	VariablePaymentTermConfiguration	판매자의 결제 요청이 승인되는 방법

ConfigurableUpfrontPricingTermConfiguration

아래 예제를 사용하여 필요한 값을 찾습니다. `ConfigurableUpfrontPricingTermConfiguration`.

GetOfferTerms 응답 예:

```

{
  "offerTerms": [
    {
      "configurableUpfrontPricingTerm": {
        "id": "term-34dfb665ebd38eb2d0269a4af454c4f33cd90292ecc7b0f71058175594f67394",
        "currencyCode": "USD",
        "rateCards": [
          {
            "constraints": {
              "multipleDimensionSelection": "Allowed",
              "quantityConfiguration": "Allowed"
            },
            "selector": {
              "type": "Duration",
              "value": "P1M"
            },
            "rateCard": [
              {
                "dimensionKey": "BasicService",
                "displayName": "Basic Service",
                "price": "0",
                "unit": "Units"
              },
              {
                "dimensionKey": "PremiumService",
                "displayName": "Premium Service",
                "price": "0",
                "unit": "Units"
              }
            ]
          }
        ],
        "type": "ConfigurableUpfrontPricingTerm"
      }
    ]
  }
}

```

Field	값/소스
ConfigurableUpfrontPricingTermConfiguration.selectorValue	offerTerms[].configurableUpfrontPricingTerm.rateCards[].selector.value
Dimension.dimensionKey	offerTerms[].configurableUpfrontPricingTerm.rateCards[].rateCard[].dimensionKey
Dimension.dimensionValue	구매자 제공 정수(해당 차원에 대해 구매할 단위 수)

Note

요금 카드의 constraints 필드는

ConfigurableUpfrontPricingTermConfiguration 예를 들어 여러 차원을 선택할 수 있는지 아니면 하나만 선택할 수 있는지와 같이 만드는 방법을 제어합니다. 자세한 내용은 [제약 데이터](#) 유형을 참조하세요.

에 대한 RequestedTerm페이로드 예제ConfigurableUpfrontPricingTerm:

```
{
  "id": "term-43ab7a2445f89dfcb7e1a6a81c92fe08c124125ffad126bea5e5b7915c065166",
  "configuration": {
    "configurableUpfrontPricingTermConfiguration": {
      "selectorValue": "P12M",
      "dimensions": [
        {
          "dimensionKey": "AdminUsers",
          "dimensionValue": 5
        },
        {
          "dimensionKey": "ReadOnlyUsers",
          "dimensionValue": 10
        }
      ]
    }
  }
}
```

}

RenewalTermConfiguration

Field	값/소스
enableAutoRenew	구매자 제공 부울(true 또는 false) - 필수

예 대한 RequestedTerm페이로드 예제RenewalTerm:

```
{
  "id": "term-ffcc0100ce0c5468426dc07d0b05cff09ab8447640108d9c95ce1591f0949738",
  "configuration": {
    "renewalTermConfiguration": {
      "enableAutoRenew": false
    }
  }
}
```

VariablePaymentTermConfiguration

Field	값/소스
RequestedTerm.configuration.variablePaymentTermConfiguration.paymentRequestApprovalStrategy	구매자 제공: AUTO_APPROVE_ON_EXPIRATION 또는 WAIT_FOR_APPROVAL - 필수
RequestedTerm.configuration.variablePaymentTermConfiguration.expirationDuration	구매자 제공 기간(예: P10D), 전략이 인 경우에 만 필요 AUTO_APPROVE_ON_EXPIRATION

예 대한 RequestedTerm페이로드 예제VariablePaymentTerm:

```
{
  "id": "term-9ab34376c51cd5b4dc995aa0f410b657d58bd577b3f09b34d38afb6964aaaf12",
  "configuration": {
```

```

    "variablePaymentTermConfiguration": {
      "paymentRequestApprovalStrategy": "AUTO_APPROVE_ON_EXPIRATION",
      "expirationDuration": "P10D"
    }
  }
}

```

제안 수락

먼저에 설명된 대로 따옴표를 생성합니다 [따옴표 생성](#). 그런 다음 AcceptAgreementRequest를 사용하여 견적을 수락하고 계약을 생성합니다. 계약 요청이 완료AcceptAgreementRequest되기 전에 호출해야 합니다. 수락 중에 구매 주문 번호를 선택적으로 연결할 수 있습니다.

```

# Use chargeId from CreateAgreementRequest response
response = client.accept_agreement_request(
    agreementRequestId='agrq-abc123',
    purchaseOrders=[
        {
            'chargeId': 'charge-xyz', # From chargeSummary.expectedCharges[].id
            'purchaseOrderReference': 'PO-2024-001'
        }
    ]
)

print(f"Agreement ID: {response['agreementId']}")

```

기존 계약 교체

에 설명된 CreateAgreementRequest 대로를 구성하되 [따옴표 생성](#) intent를 로 설정하고 대체할 계약을 REPLACE sourceAgreementIdentifier 제공합니다. 그런 다음 호출AcceptAgreementRequest하여 교체를 완료합니다.

```

response = client.create_agreement_request(
    agreementProposalIdentifier='at-edhtjnbilupjv3xqbphtom77y',
    intent='REPLACE',
    sourceAgreementIdentifier='agmt-existing123'
)

print(f"Agreement Request ID: {response['agreementRequestId']}")

```

기존 계약 수정

기존 계약의 기간 구성을 수정하려면 intent 로 설정하고 AMEND 제공합니

다 sourceAgreementIdentifier. 새 계약과 달리 agreementProposalIdentifier는 필요하지 않습니다. 구성을 변경하려는 용어만 포함합니다. 그런 다음 AcceptAgreementRequest하여 수정 사항을 적용합니다.

```
response = client.create_agreement_request(
    intent='AMEND',
    sourceAgreementIdentifier='agmt-existing123',
    requestedTerms=[
        {
            'id': 'term-configurable-pricing-789',
            'configuration': {
                'configurableUpfrontPricingTermConfiguration': {
                    'selectorValue': 'P12M',
                    'dimensions': [
                        {'dimensionKey': 'Users', 'dimensionValue': 100}
                    ]
                }
            }
        }
    ]
)

print(f"Agreement Request ID: {response['agreementRequestId']}")
```

자동 갱신 활성화 또는 비활성화

AMEND 의도와 CreateAgreementRequest 함께 사용하여 자동 갱신을 전환합니다.

```
response = client.create_agreement_request(
    intent='AMEND',
    sourceAgreementIdentifier='agmt-existing123',
    requestedTerms=[
        {
            'id': 'term-renewal-456',
            'configuration': {
                'renewalTermConfiguration': {'enableAutoRenew': True}
            }
        }
    ]
)
```

```
)  
  
print(f"Agreement Request ID: {response['agreementRequestId']}")
```

계약 취소

CancelAgreement를 사용하여 사용 계약을 취소합니다. 다른 계약 유형의 경우 구매자는 판매자에게 연락하여 취소를 시작해야 합니다.

```
client.cancel_agreement(  
    agreementId='agmt-abc123'  
)
```

계약 검색

SearchAgreements를 사용하여 수락자로 참여하는 모든 계약을 검색합니다. 검색은 기본 계약 정보가 포함된 계약 목록을 반환합니다.

```
response = client.search_agreements(  
    catalog='AWSMarketplace',  
    filters=[  
        {'name': 'PartyType', 'values': ['Acceptor']},  
        {'name': 'AgreementType', 'values': ['PurchaseAgreement']},  
        {'name': 'Status', 'values': ['ACTIVE']}  
    ]  
)  
  
for agmt in response['agreementViewSummaries']:  
    print(f" {agmt['agreementId']}: {agmt['status']}")
```

계약 설명

제안자, 수락자, 시작 날짜 및 종료 날짜와 같은 계약에 대한 세부 정보를 보는 DescribeAgreement를 사용합니다.

```
response = client.describe_agreement(  
    agreementId='agmt-abc123'  
)  
  
print(f"Status: {response['status']}")
```

```
print(f"Start: {response['startTime']}")
print(f"End: {response['endTime']}")
```

계약 조건 가져오기

GetAgreementTerms를 사용하여 수락자로 참여하는 계약의 조건에 대한 세부 정보를 얻습니다.

```
response = client.get_agreement_terms(
    agreementId='agmt-abc123'
)

for term in response['acceptedTerms']:
    print(f"  Term: {term}")
```

계약 권한 가져오기

GetAgreementEntitlements를 사용하여 계약과 연결된 권한의 상태 및 세부 정보를 계약 수준에서 확인할 수 있습니다.

```
response = client.get_agreement_entitlements(
    agreementId='agmt-abc123'
)

for ent in response['agreementEntitlements']:
    print(f"  {ent['resource']}: {ent['status']}")
```

등록 토큰 가져오기

GetAgreementEntitlements를 사용하여 등록 토큰을 가져옵니다. 이 수명이 짧은 토큰(30분 동안 유효)은 수락자가 제안자가 있는 계정을 설정하는 데 필요합니다. 현재 SaaS 구매 계약에만 적용됩니다.

```
response = client.get_agreement_entitlements(
    agreementId='agmt-abc123'
)

for ent in response['agreementEntitlements']:
    if 'registrationToken' in ent:
        print(f"Registration Token: {ent['registrationToken']}")
        # Token is valid for 30 minutes
```


계약 요금 나열

ListAgreementCharges를 사용하여 계약과 관련된 요금 및 구매 주문 세부 정보를 봅니다.

```
response = client.list_agreement_charges(
    agreementId='agmt-abc123'
)

for charge in response['items']:
    print(f" {charge['id']} (rev {charge['revision']}): ${charge['amount']}")
    print(f" P0: {charge.get('purchaseOrderReference', 'N/A')}")
```

구매 주문 업데이트

제품을 구독한 후 구매 주문 번호를 추가하는 UpdatePurchaseOrders 데 사용합니다. [계약 요금 나열](#) 응답에서 chargeId 및 chargeRevision를 사용합니다. 구매 주문이 요금과 연결되면 해당 요금에 대해 생성된 인보이스에 구매 주문 번호가 포함됩니다.

```
client.update_purchase_orders(
    agreementId='agmt-abc123',
    purchaseOrders=[
        {
            'chargeId': 'charge-xyz',
            'chargeRevision': 1,
            'purchaseOrderReference': 'PO-2024-001'
        }
    ]
)
```

취소 요청 나열

ListAgreementCancellationRequests를 사용하여 참여하는 계약에 대한 모든 취소 요청을 나열합니다. 계약, 상태 및 기타 기준을 기준으로 필터링할 수 있습니다.

```
response = client.list_agreement_cancellation_requests(
    partyType='Acceptor',
    agreementId='agmt-abc123'
)

for req in response['agreementCancellationRequests']:
```

```
print(f" {req['agreementCancellationRequestId']}: {req['status']}")
```

취소 요청 세부 정보 가져오기

GetAgreementCancellationRequest를 사용하여 상태, 타임스탬프 및 사유 코드를 포함하여 특정 취소 요청에 대한 세부 정보를 검색합니다.

```
response = client.get_agreement_cancellation_request(
    agreementId='agmt-abc123',
    agreementCancellationRequestId='acr-abc123'
)

print(f"Status: {response['status']}")
print(f"Reason: {response['reasonCode']}")
```

취소 요청 수락

판매자가 시작한 취소 요청을 승인AcceptAgreementCancellationRequest하는 데 사용합니다. 승인 후 계약 취소 워크플로는 비동기적으로 실행됩니다. 취소 요청을 승인하면 계약이 취소되므로 CancelAgreement 권한도 필요합니다.

```
client.accept_agreement_cancellation_request(
    agreementId='agmt-abc123',
    agreementCancellationRequestId='acr-abc123'
)
```

취소 요청 거부

판매자가 시작한 취소 요청을 거부RejectAgreementCancellationRequest하는 데 사용합니다. 거부 후에도 계약은 활성 상태로 유지되며 판매자는 필요한 경우 새 취소 요청을 생성할 수 있습니다.

```
client.reject_agreement_cancellation_request(
    agreementId='agmt-abc123',
    agreementCancellationRequestId='acr-abc123'
)
```

결제 요청 나열

ListAgreementPaymentRequests를 사용하여 참여하는 계약에 대한 모든 결제 요청을 나열합니다. 계약, 상태 및 기타 기준을 기준으로 필터링할 수 있습니다.

```
response = client.list_agreement_payment_requests(
    partyType='Acceptor',
    agreementId='agmt-abc123'
)

for req in response['agreementPaymentRequests']:
    print(f" {req['agreementPaymentRequestId']}: {req['status']}")
```

결제 요청 세부 정보 가져오기

상태, 타임스탬프 및 관련 요금을 포함하여 특정 결제 요청에 대한 세부 정보를 검색하는 `GetAgreementPaymentRequest` 데 사용합니다.

```
response = client.get_agreement_payment_request(
    paymentRequestId='apr-abc123',
    agreementId='agmt-abc123'
)

print(f"Status: {response['status']}")
print(f"Amount: ${response['amount']}")
```

결제 요청 수락

활성 계약에 대해 판매자가 시작한 결제 요청을 승인 `AcceptAgreementPaymentRequest` 하는 데 사용합니다.

```
client.accept_agreement_payment_request(
    paymentRequestId='apr-abc123',
    agreementId='agmt-abc123'
)
```

결제 요청 거부

판매자가 시작한 결제 요청을 거부 `RejectAgreementPaymentRequest` 하는 데 사용합니다. 거부 후 판매자는 필요한 경우 새 결제 요청을 생성할 수 있습니다.

```
client.reject_agreement_payment_request(
    paymentRequestId='apr-abc123',
    agreementId='agmt-abc123'
)
```

AWS MarketplaceAPI를 사용하여 프라이빗 마켓플레이스 작업

AWS Marketplace Catalog API를 사용하여 AWS 계정 또는 [조직의](#) 프라이빗 마켓플레이스를 관리할 수 있습니다.

모든 변경 유형은 조직의 관리 계정 또는 프라이빗 마켓플레이스의 위임된 관리자인 멤버 계정에서만 호출할 수 있습니다. 프라이빗 마켓플레이스에 대한 AWS Organizations 통합이 없는 현재 프라이빗 마켓플레이스 고객인 경우 AWSPrivateMarketplaceAdminFullAccess IAM 정책이 있는 조직의 모든 계정으로 프라이빗 마켓플레이스를 생성하고 관리할 수 있습니다.

프라이빗 마켓플레이스에 대한 자세한 내용은 AWS Marketplace 구매자 안내서의 [프라이빗 마켓플레이스](#)를 참조하세요.

다음 표에는 프라이빗 마켓플레이스를 관리하기 위한 작업 세트와 각 작업에 적용되는 변경 유형이 자세히 나와 있습니다.

Task	작업	변경 유형
the section called “프라이빗 마켓플레이스 생성”	StartChangeSet	CreateExperience CreateProcurementPolicy
the section called “프라이빗 마켓플레이스 경험의 브랜딩 변경”	StartChangeSet	CreateBrandingSettings UpdateBrandingSettings
the section called “프라이빗 마켓플레이스 경험 활성화 또는 비활성화”	StartChangeSet	UpdateExperience
the section called “사용자 요청 활성화 또는 비활성화”	StartChangeSet	UpdateProcurementPolicy
the section called “프라이빗 마켓플레이스 경험에서 제품 목록 가져오기”	DescribeEntity	해당 사항 없음

Task	작업	변경 유형
the section called “프라이빗 마켓플레이스에서 제품 추가 또는 제거”	StartChangeSet	AllowProductProcurement DenyProductProcurement
the section called “제품 찾기”	해당 사항 없음	해당 사항 없음
the section called “용 프라이빗 마켓플레이스 작업AWS Organizations”	해당 사항 없음	해당 사항 없음
the section called “보안 주체를 경험에 연결”	StartChangeSet	AssociateAudience DisassociateAudience
프라이빗 마켓플레이스 경험 보관 및 재활성화	StartChangeSet	RestrictExperience ReviveExperience

프라이빗 마켓플레이스 생성

의 프라이빗 마켓플레이스는 사용자가 해당 계정에서 구매할 AWS 계정수 있는 제품 목록과 마켓플레이스의 브랜딩으로 생각할 수 있습니다. 계정이 여러 개인 조직에서는 [OU\(조직 단위\)](#)라는 AWS Organizations 그룹화를 사용하여 경험에 연결할 수 있습니다. 예를 들어 조직의 모든 계정이 조달할 수 있는 제품 세트가 하나 있거나 조직의 각 OU에 대해 다른 제품 목록이 있을 수 있습니다. 조직의 개별 계정에 대해 다른 제품 목록을 가질 수도 있습니다. 승인된 제품 및 브랜딩의 각 목록을 조달 경험이라고 합니다.

에서 AWS Marketplace Catalog API 네 개의 엔터티는 경험을 나타냅니다.

- Experience 엔터티 - 이 엔터티는 경험의 최고 수준에 있으며 두 개의 하위 엔터티를 포함합니다.
- ProcurementPolicy 개체 - 이 개체는 프라이빗 마켓플레이스에서 허용 및 거부된 제품을 나타냅니다.
- BrandingSettings 개체 - BrandingSettings 개체를 생성하여 프라이빗 마켓플레이스가 사용자에게 어떻게 보이는지 정의할 수도 있습니다.

- Audience 엔터티 - 경험이 적용되는 AWS 계정OU 또는 조직의 집합을 정의하는 하나 이상의 Audience 엔터OU도 연결해야 합니다.

조달 경험을 생성하는 단계는 다음과 같습니다.

1. Experience 개체를 생성합니다.
2. ProcurementPolicy 엔터티를 생성하여 경험에 허용되거나 거부된 제품 목록을 저장합니다.
3. (선택 사항) BrandingSettings 엔터티를 생성하여 마켓플레이스 경험의 모양을 사용자 지정합니다.
4. 보안 주체를 사용자 경험과 연결합니다. 보안 주체는 AWS 계정, OU 또는 조직일 수 있습니다.
5. 환경을 활성화합니다.

Note

계정이 조직에 속해 있는 경우 섹션을 AWS Organizations 참조하세요 [용 프라이빗 마켓플레이스 작업AWS Organizations](#).

Experience 개체 생성

Experience 개체를 생성하려면 ChangeType 파라미터 CreateExperience 값과 함께 StartChangeSet 작업을 사용하여 경험을 생성하도록 요청합니다 AWS Marketplace. 다음 코드 예제를 참조하세요.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "CreateExperience",
      "DetailsDocument":
      {
        "Name": "ExamplePrivateMarketplace"
      },
      "Entity":
```

```
{
  "Type": "Experience@1.0"
}
],
"ChangeSetName": "Create Private Marketplace Example"
}
```

이 작업에서 Entity는 생성하려는 엔터티의 템플릿입니다. 생성 EntityId 시이 할당됩니다.는 나중에 찾을 수 있도록 변경 사항을 ChangeSetName 식별합니다.

그 응답은 다음과 같습니다.

```
{
  "ChangeSetArn": "arn:...:AWSMarketplace/ChangeSet/abcd1234example5678frjzkz",
  "ChangeSetId": "abcd1234example5678frjzkz"
}
```

응답에는 로 처리ChangeSetId되는 변경 요청의 상태를 가져오는 데 사용할 수 있는가 포함됩니다DescribeChangeSet. ListEntities를 사용하여 없이 Experience개체를 찾을 수도 있습니다ChangeSetId. 변경 세트에 대한 자세한 내용은 [변경 세트 작업](#)의 내용을 참조하세요.

새로 생성된 Experience엔터티에는 기본적으로 조달 정책이 없습니다. 또한 브랜딩에 대한 기본 설정으로 생성됩니다. 사용자 지정 방법을 포함하여 브랜딩 설정에 대한 자세한 내용은 섹션을 참조하세요 [프라이빗 마켓플레이스 경험의 브랜딩 변경](#).

ProcurementPolicy 개체 생성

ProcurementPolicy 개체를 생성해야 합니다. 기본적으로 새 Experience엔터티는 비활성화되어 있으므로 활성화하기 전에 조달 정책을 생성할 수 있습니다.

Note

조달 정책(null)이 없는 Experience엔터티는 프라이빗 마켓플레이스에서 모든 제품을 조달할 수 있도록 허용합니다. 빈 조달 정책이 있는 Experience엔터티에는 사용자가 조달할 수 있는 제품이 없습니다.

프라이빗 마켓플레이스에서 제품을 허용하고 거부하려면 조달 정책을 생성해야 합니다. 이렇게 하려면 다시 호출StartChangeSet하지만 이번에는 ChangeType의를 사용합니다CreateProcurementPolicy. 다음 코드 예제에서는 빈 조달 정책을 생성합니다.

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "CreateProcurementPolicy",
      "DetailsDocument":
      {
        "Name": "ExampleProcurementPolicy"
      },
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example@1"
      }
    }
  ]
}

```

이 작업에서 Entity 제공하는는 조달 정책을 생성하려는 Experience 엔터티이므로 이전에 생성한 엔터티의 식별자를 포함해야 합니다. ListEntities를 사용하여 Experience 개체를 찾습니다. CreateExperience 작업의 변경 세트 식별자 DescribeChangeSet와 함께 사용하여 식별자를 반환할 수도 있습니다.

Note

이 예제는 개정이 인 식별자를 보여줍니다¹. 식별자의 개정에 대한 자세한 내용은 섹션을 참조하세요 [식별자](#).

CreateProcurementPolicy 변경 유형에 DescribeChangeSet를 다시 사용하여 요청 처리를 따를 수 있습니다.

Note

Experience 및 ProcurementPolicy 객체에 지정한 이름은에 표시되지 않습니다AWS Marketplace. 이름은 API에서 엔터티를 쉽게 찾을 수 있도록 하기 위한 것입니다.

조달 정책을 생성하면 프라이빗 마켓플레이스가에 표시됩니다AWS Management Console. ([프라이빗 마켓플레이스 페이지](#)로 이동하여 볼 수 있습니다.) 이 단계를 완료하면 프라이빗 마켓플레이스가 비활성화되고, 기본 브랜딩이 있으며, 빈 조달 정책이 있고, 조직의 보안 주체와 연결되지 않습니다. 브랜딩을 업데이트하고 원하는 제품을 추가하고 경험을 하나 이상의 계정과 연결한 다음 프라이빗 마켓플레이스를 활성화할 수 있습니다.

다음 섹션에서는를 사용하여 프라이빗 마켓플레이스를 관리하는 방법을 설명합니다AWS Marketplace Catalog API.

프라이빗 마켓플레이스 경험의 브랜딩 변경

사용자에 맞게 프라이빗 마켓플레이스의 모양을 사용자 지정할 수 있습니다. 사용자 지정이 없으면 프라이빗 마켓플레이스의 기본 브랜딩 설정이 아래에 설명되어 있습니다. 프라이빗 마켓플레이스에서 변경할 수 있는 브랜딩의 측면은 다음과 같습니다.

- **Title** - 프라이빗 마켓플레이스에 표시되는 이름입니다. 이는 프라이빗 마켓플레이스 프로파일 설정 화면의 이름 필드와 동일합니다. 를 Title로 설정하면 표시된 텍스트**Example**는 Example Private Marketplace입니다. 기본값은 Private Marketplace입니다.
- **Information** - 프라이빗 마켓플레이스의 이름 아래에 표시되는 단락입니다. 프로필 설정의 설명 필드와 동일합니다. 기본값은 정보가 아니며,이 경우 프라이빗 마켓플레이스에 대한 일반적인 설명이 표시됩니다.
- **ThemeColor** - 프라이빗 마켓플레이스의 배너에 표시되는 색상입니다. RGB 16진수 형식의 색상입니다. 이 값은 프로파일 설정의 테마 색상 필드와 동일합니다. 기본값은 #232F3E입니다.
- **LogoUrl** - 프라이빗 마켓플레이스에서 로고로 사용할 이미지 파일을 가리키는 URL입니다. URL은 공개적으로 사용할 수 있어야 합니다(예: 서명된 Amazon S3 URL). 파일은 .png 또는 .svg 파일이어야 하며 500kb 미만이어야 합니다. 필요한 경우 이미지 파일의 크기를 최대 높이 30픽셀, 최대 너비 100픽셀로 조정합니다. 프로필 설정의 로고 선택과 동일한 값입니다. 기본값은 로고를 표시하지 않는 것입니다.

이러한 값을 설정하려면 먼저 CreateBrandingSettings 변경 유형으로 BrandingSettings객체를 생성해야 합니다. 그런 다음 UpdateBrandingSettings 변경을 요청하여 브랜딩을 설정하거나 변경할 수 있습니다. BrandingSettings 객체는 한 번만 생성하면 됩니다. 이 객체를 생성하려면 다음 코드 예제와 같이 CreateBrandingSettings 변경 유형StartChangeSet으로호출합니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json
```

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "CreateBrandingSettings",
      "DetailsDocument":
      {
        "Name": "ExampleBrandingSettingsName"
      },
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example@2"
      }
    }
  ]
}
```

이 예제에서는 BrandingSettings 객체를 추가하여 Experience객체를 수정합니다. 개체 식별자의 개정이 2로 증가했습니다². 식별자의 개정에 대한 자세한 내용은 섹션을 참조하세요 [식별자](#).

Note

호출에서 브랜딩 설정의 모든 세부 정보를 지정하여 브랜딩 설정 개체를 생성할 수 있습니다. 세부 정보 패킷은 CreateBrandingSettings 및 UpdateBrandingSettings에 대해 동일합니다.

UpdateBrandingSettings 변경 유형 StartChangeSet으로 호출하여 설정을 수정합니다. 설정은 DetailsDocument 객체 Configuration의 일부입니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateBrandingSettings",
      "DetailsDocument":
```

```

{
  "Name": "ExampleBrandingSettingsName",
  "Description": "Example description",
  "Configuration":
  {
    "Title": "ExampleName",
    "Information": "Example description.",
    "ThemeColor": "#0e7f74",
    "LogoUrl": "https://example.com/path/mylogo.png"
  }
},
"Entity":
{
  "Type": "Experience@1.0",
  "Identifier": "exp-1234example@3"
}
}
]
}

```

Note

로고의 URL은 업데이트 변경 중에 복사하는 데 사용됩니다. 변경이 완료된 후 해당 경로에서 URL을 제거하거나 변경하면 다시 요청하지 않는 한 프라이빗 마켓플레이스에 영향을 주지 않습니다 UpdateBrandingSettings.

프라이빗 마켓플레이스 경험 활성화 또는 비활성화

프라이빗 마켓플레이스가 활성화되면(및 조달 정책이 있는 경우) 연결된 계정의 사용자는 사용자가 승인한 제품만 구매할 수 있습니다. 계정에 프라이빗 마켓플레이스 환경이 활성화되지 않은 경우 사용자는 전체 AWS Marketplace 카탈로그에서 제품을 구매할 수 있습니다.

프라이빗 마켓플레이스를 활성화하려면 UpdateExperience 변경 유형과 함께 StartChangeRequest 작업을 사용합니다.

```

POST /StartChangeSet HTTP/1.1
Content-type: application/json

```

```

{
  "Catalog": "AWSMarketplace",

```

```

"ChangeSet":
[
  {
    "ChangeType": "UpdateExperience",
    "DetailsDocument":
    {
      "Status": "Enabled"
    },
    "Entity":
    {
      "Type": "Experience@1.0",
      "Identifier": "exp-1234example@4"
    }
  }
]
}

```

마찬가지로 및와 동일한 작업을 사용할 수 있지만 ChangeTypeStatus의를 DetailsDocument로 변경Disabled하여 프라이빗 마켓플레이스를 비활성화할 수 있습니다.

Note

프라이빗 마켓플레이스를 비활성화하면 허용 및 거부된 제품 목록과 브랜딩과 같은 사용자 지정이 모두 유지됩니다. 프라이빗 마켓플레이스가 비활성화되면 사용자는 더 이상 프라이빗 마켓플레이스를 볼 수 없습니다(조직의 기본 환경에 따라 관리될 수 있음). 계정에 대해 활성화된 프라이빗 마켓플레이스 경험이 없는 경우 모든 제한이 제거되고 사용자는 퍼블릭 제품을 조달할 수 있습니다AWS Marketplace.

사용자 요청 활성화 또는 비활성화

조직의 사용자는 전체 퍼블릭을 볼 수 AWS Marketplace있지만 허용한 제품만 구독할 수 있습니다. 기본적으로 프라이빗 마켓플레이스에 없는 제품을 추가하도록 요청할 수 있습니다. 이러한 요청은 프라이빗 마켓플레이스 관리자 페이지([프라이빗 마켓플레이스](#))에 표시되며, 여기서 요청을 수락 또는 거부할지 여부(및 동일한 제품에 대한 추가 요청을 차단할지 여부)를 결정할 수 있습니다. 카탈로그 API를 사용하여 요청을 보거나 응답할 수 없습니다.

사용자가 프라이빗 마켓플레이스 경험에 대한 요청을 생성할 수 있는 기능을 활성화하거나 비활성화할 수 있습니다. 를 UpdateProcurementPolicy 변경 유형StartChangeSet과 함께 사용합니다. 다음 코드 예제에서는 요청 기능이 비활성화되어 있습니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateProcurementPolicy",
      "DetailsDocument":
      {
        "Configuration":
        {
          "PolicyResourceRequests": "Deny"
        }
      },
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example@5"
      }
    }
  ]
}
```

사용자에 대한 변경 요청 기능을 활성화하려면 Deny에서 Allow 대신을 사용합니
다PolicyResourceRequests.

이 설정의 현재 상태를 가져오는 방법을 알아보려면 다음 섹션인 단원을 참조하십시오 [프라이빗 마켓플레이스 경험에서 제품 목록 가져오기](#).

프라이빗 마켓플레이스 경험에서 제품 목록 가져오기

프라이빗 마켓플레이스에서 허용(및 거부)된 제품은 Experience엔터티의 조달 정책의 일부입니다. 프라이빗 마켓플레이스의 조달 정책에 대한 세부 정보를 가져오려면 먼저 Experience엔터티에서 조달 정책 식별자를 가져온 다음 해당 식별자를 DescribeEntity 사용하여 호출합니다.

조달 정책 식별자를 가져오려면 다음 명령과 같이 관심 있는 Experience엔터티DescribeEntity에
를 사용합니다.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=exp-example01
```

다음은 응답의 예입니다.

```
{
  "Details": "{\\"Name\\":\\"New Private Marketplace\\", \\"Status\\":\\"Enabled\\",
    \\"ProcurementPolicies\\":[\\"procpolicy-123example456\\"], \\"BrandingSettings\\":
    [\\"brandsettings-456example123\\"]}",
  "DetailsDocument":
  {
    "Name": "New Private Marketplace",
    "Status": "Enabled",
    "ProcurementPolicies":
    [
      "procpolicy-123example456"
    ],
    "BrandingSettings":
    [
      "brandsettings-456example123"
    ]
  },
  "EntityArn": "arn:<...>:AWSMarketplace/Experience/exp-example-01",
  "EntityIdentifier": "exp-example01@6",
  "EntityType": "Experience@1.0",
  "LastModifiedDate": "2021-01-13T20:31:36Z"
}
```

Note

DetailsDocument 속성에는 개체 세부 정보가 JSON 객체로 포함됩니다. 레거시 Details 속성에는 문자열과 동일한 JSON 객체가 포함됩니다.

다음 명령과 같이 조달 정책에 EntityId 대해 반환된를 사용하여 세부 정보를 가져올 수 있습니다.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=procpolicy-123example456
```

이렇게 하면 허용된 제품과 거부된 제품을 모두 포함하여 정책의 전체 세부 정보가 반환됩니다. 다음은 응답의 예입니다.

```
{
  "Details": "{\\"Name\\":\\"ExampleProcurementPolicy\\", \\"Statements\\":[{\\"Effect
    \":\\"Allow\\",\\"Resources\\":[{\\"Type\\":\\"Product\\",\\"Ids\\":[\\"example1-1234-
```

```

abcd-5678-90abcdef1234\"]), {\\"Type\\": \\"Product\\", \\"Ids\\": [\\"example2-2345-
bcde-6789-01bcdea2345\\"]]}], {\\"Effect\\": \\"Deny\\", \\"Resources\\": [{\\"Type\\": \\"Product
\\", \\"Ids\\": [\\"example3-3456-cdef-7890-12defabc5678\\"]}]}], \\"Configuration\\":
{\\"PolicyResourceRequests\\": \\"Allow\\"}},
  \\"DetailsDocument\\":
  {
    \\"Name\\": \\"ExampleProcurementPolicy\\",
    \\"Statements\\":
    [
      {
        \\"Effect\\": \\"Allow\\",
        \\"Resources\\":
        [
          {
            \\"Type\\": \\"Product\\",
            \\"Ids\\":
            [
              \\"example1-1234-abcd-5678-90abcdef1234\\"
            ]
          },
          {
            \\"Type\\": \\"Product\\",
            \\"Ids\\":
            [
              \\"example2-2345-bcde-6789-01bcdea2345\\"
            ]
          }
        ]
      },
      {
        \\"Effect\\": \\"Deny\\",
        \\"Resources\\":
        [
          {
            \\"Type\\": \\"Product\\",
            \\"Ids\\":
            [
              \\"example3-3456-cdef-7890-12defabc5678\\"
            ]
          }
        ]
      }
    ]
  },
  \\"Configuration\\":

```

```
{
  "PolicyResourceRequests": "Allow"
},
"EntityArn": "arn:<...>AWSMarketplace/ProcurementPolicy/procpolicy-123example456",
"EntityIdentifier": "procpolicy-123example456@4",
"EntityType": "ProcurementPolicy@1.0",
"LastModifiedDate": "2020-10-01T12:00:00Z"
}
```

이 예제에서 조달 정책에는 두 개의 허용된 제품과 한 개의 거부된 제품이 있습니다. 정책은 사용자 리소스 요청을 허용합니다.

프라이빗 마켓플레이스에서 제품 추가 또는 제거

기본적으로 프라이빗 마켓플레이스에는 승인된 제품이 없습니다. 변경 요청을 사용하여 제품을 추가하거나 제거합니다. 제품을 추가하려면 AllowProductProcurement 변경 유형을 사용합니다. 제품을 제거하려면 DenyProductProcurement 변경 유형을 사용합니다.

다음 코드 예제에서는 프라이빗 마켓플레이스에 제품을 추가하는 StartChangeSet 작업이 포함된 AllowProductProcurement 변경 유형을 보여줍니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AllowProductProcurement",
      "DetailsDocument": {
        "Products": [
          {
            "Ids": [
              "example-1234-abcd-5678-90abcded1234"
            ],
            "Notes": "Useful product"
          }
        ]
      }
    ]
  }
}
```



```

    },
    "Entity":
    {
        "Identifier": "exp-1234example@6",
        "Type": "Experience@1.0"
    }
  }
]
}

```

를 사용하여 프라이빗 마켓플레이스의 Experience 개체에 제품을 추가합니다. `AllowProductProcurement`. 프라이빗 마켓플레이스에서 제품을 제거하는 구문은 동일하지만 `DenyProductProcurement` `ChangeType` 대신를 사용하는 경우는 예외입니다. `AllowProductProcurement`. 제품은 `ProcurementPolicy` 개체에 포함된 Experience 개체의 허용(또는 거부) 목록에 추가됩니다.

Note

`DetailsDocument` 변경 사항의에 있는 제품 목록은의 `배열Ids`이므로 제품 식별자 목록을 포함하여 한 번의 호출로 여러 제품을 추가(또는 제거)할 수 있습니다. 한 번의 요청으로 50개의 제품이 제한됩니다.

의 목록에 대한 `Notes` 필드는 필요하지 `Ids` 않습니다. 그러나 이를 사용하여 제품 세트를 허용하거나 거부하기로 결정한 이유를 기록할 수 있습니다.

제품 찾기

조달 정책의 세부 정보를 가져오면 이미 프라이빗 마켓플레이스에 있는 제품의 IDs를 찾을 수 있습니다. 그러나 AWS Marketplace Catalog API는 다른 제품의 제품 IDs를 찾는 방법을 제공하지 않습니다. 카탈로그 API 서비스와 함께 사용할 제품 IDs를 가져오는 방법에는 두 가지가 있습니다.

- 퍼블릭 마켓플레이스 - 퍼블릭 마켓플레이스에서 제품을 찾은 후 구독 계속을 선택하여 제품에 대한 세부 정보 페이지를 확인합니다(제품을 구독하지 않음). URL에는 제품 ID가 파라미터로 포함됩니다. 예를 들어 URL에서 `https://aws.amazon.com/marketplace/fulfillment?productId=ab1234cd-1234-abcd-5678-90abcdef1234&ref_=dtl_psb_continue`**`ab1234cd-1234-abcd-5678-90abcdef1234`**는 제품 ID입니다.
- AWS MarketplaceDiscovery API - 프로그래밍 방식으로 Discovery API를 AWS Marketplace 사용하여의 전체 제품 목록에 액세스할 수 있습니다. Discovery API는 프라이빗 API입니다. 액세스를 사용

하려면 액세스를 요청해야 합니다. 자세한 내용은 [AWS Marketplace Discovery API에 대한 액세스 제어](#) 단원을 참조하십시오.

용 프라이빗 마켓플레이스 작업AWS Organizations

계정의 프라이빗 마켓플레이스로 작업하든 조직의 프라이빗 마켓플레이스로 작업하든 동일한 API를 사용합니다. 그러나 조직 내에서 작업할 때는 다음과 같은 차이점이 있습니다.

- 조직에서 프라이빗 마켓플레이스 기능을 사용하려면 먼저 신뢰할 수 [있는 액세스를 할성화](#)하여 AWS Organizations데이터에 대한 프라이빗 마켓플레이스 서비스(private-marketplace.marketplace.amazonaws.com) 액세스를 제공해야 합니다. 또한 관리 계정에서 [프라이빗 마켓플레이스 서비스 연결 역할을 생성](#)해야 합니다. 이 역할에는 프라이빗 마켓플레이스가 사용자를 대신하여 프라이빗 AWS Organizations마켓플레이스 리소스를 설명하고 업데이트하는 데 필요한 모든 권한이 포함됩니다. 이러한 작업은 관리 계정에서만 수행할 수 있습니다. 프라이빗 마켓플레이스 관리자 페이지를 사용하여이 활성화를 수행하는 것이 좋습니다. 신규 고객인 경우 AWS Marketplace구매자 안내서의 [프라이빗 마켓플레이스를 참조하세요](#). 기존 고객인 경우 AWS Marketplace구매자 안내서의 [프라이빗 마켓플레이스 생성 및 관리](#)를 참조하세요.
- 조직의 프라이빗 마켓플레이스 리소스는 관리 계정에서 생성되고 프라이빗 마켓플레이스의 위임된 관리자인 멤버 계정과 공유됩니다.
- 프라이빗 마켓플레이스의 위임된 관리자인 멤버 계정에서 프라이빗 마켓플레이스의 객체를 나열할 때는 SharedWithMe 필터를 사용하여 객체를 구체적으로 요청해야 합니다. 이는 ListEntities 및 ListChangeSets 작업 모두에 적용됩니다.

자체 계정의 Experience 객체를 나열하려면 다음 코드 예제와 ListEntities 같이 호출합니다.

```
POST /ListEntities HTTP/1.1
Content-Type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "Experience"
}
```

그러나 공유된 엔터티를 나열하려면 다음 코드 예제FilterList와 SharedWithMe같이 Scope인를 추가해야 합니다. 따라서 사용자 계정 외부에서 AWS Marketplace검색하여 사용자와 공유되는 엔터티를 찾습니다.

```
POST /ListEntities HTTP/1.1
Content-Type: application/json
```

```
{"Catalog": "AWSMarketplace",
  "EntityType": "Experience",
  "FilterList":
    [{
      "Name": "Scope",
      "ValueList":
        ["SharedWithMe"]
    }]}
```

이 경우 계정 외부의 엔터티(조직의 엔터티)만 반환됩니다.

마찬가지로 호출하려면 다음 코드 예제와 같이 범위를 설정해야 ListChangeSets합니다.

```
POST /ListChangeSets HTTP/1.1
Content-Type: application/json
```

```
{"Catalog": "AWSMarketplace",
  "FilterList":
    [{
      "Name": "Scope",
      "ValueList":
        ["SharedWithMe"]
    }]}
```

그러면 조직의 공유 프라이빗 마켓플레이스에 적용되는 변경 세트가 반환됩니다.

보안 주체를 경험에 연결

조직에 영향을 미치려면 프라이빗 마켓플레이스 환경에 하나 이상의 보안 주체가 연결되어 있어야 합니다. 단일의 경우 프라이빗 마켓플레이스를 사용하려면 계정을 경험과 연결AWS 계정해야 합니다. 조직에서는 여러 보안 주체에 여러 경험을 적용할 수 있습니다.

Note

- 조직과 연결된 환경은 조직의 다른 모든 계정에 대한 기본값입니다. 멤버 계정 또는 OU를 다른 환경과 연결하면 OU의 멤버 계정 또는 하위 계정에 대한 다른 환경이 직접 설정됩니다.

- 프라이빗 마켓플레이스에 AWS Organizations 대한 통합이 없는 현재 프라이빗 마켓플레이스 고객인 경우 관리 계정과 연결된 환경은 조직의 다른 모든 계정에 대한 기본값입니다.

보안 주체를 환경에 연결하려면 다음 코드 예제와 같이 AssociateAudience 변경 유형을 StartChangeSet 작업과 함께 사용합니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier": "exp-example01@1"
      },
      "ChangeType": "AssociateAudience",
      "DetailsDocument": {
        "Name": "AudienceName",
        "Description": "Audience example.",
        "Principals": [
          "012345678901",
          "ou-abcd-01234567",
          "o-0123456789"
        ]
      }
    }
  ],
  "ChangeSetName": "Set Audience for experience 01"
}
```

대상은와 연결된 보안 주체 목록입니다Experience. 보안 주체는 ID로 정의된 , AWS 계정조직 단위 또는 조직입니다. Principals는 목록이므로 경험과 연결할 여러 보안 주체를 포함할 수 있습니다. 첫 번째 호출 후 AssociateAudience 변경 유형에 대한 후속 호출은 경험을 위해 연결에 보안 주체를 추가합니다.

경험에서 계정을 제거할 수도 있습니다. 다음 코드 예제와 같이 DisassociateAudience 변경 유형을 사용하여이 작업을 수행합니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-example01@02"
      },
      "ChangeType": "DisassociateAudience",
      "DetailsDocument":
      {
        "Principals":
        [
          "012345678901",
          "ou-abcd-01234567",
          "o-0123456789"
        ]
      }
    }
  ],
  "ChangeSetName": "Disassociate audience example"
}
```

Note

보안 주체는 하나의 환경에만 직접 연결할 수 있습니다. 보안 주체가 한 경험과 직접 연결되지 않도록 다른 경험으로 이동하려면 보안 주체를 초기 경험과 연결 해제한 다음 두 번째 경험과 연결해야 합니다.

프라이빗 마켓플레이스 경험 보관 및 재활성화

프라이빗 마켓플레이스 경험을 보관하여 제거할 수 있습니다. 보관된 경험은 업데이트하거나 조직의 계정을 관리하는 데 사용할 수 없습니다. 보관된 경험과 연결된 대상이 있는 경우 해당 대상을 다른 경험과 연결할 수 있습니다. 나중에 언제든지 보관된 경험을 다시 활성화할 수 있습니다. 관리 계정 또는 프라이빗 마켓플레이스의 위임된 관리자인 멤버 계정의 관리자는 경험을 아카이브하고 다시 활성화할 수 있는 권한이 있습니다. 프라이빗 마켓플레이스에 대한 AWS Organizations 통합이 없는 현재 프라이빗 마켓플레이스 고객인 경우 경험을 생성한 계정의 관리자는 경험을 아카이브하고 다시 활성화할 수 있는 권한이 있습니다.

Note

경험을 보관하기 전에 반드시 비활성화해야 합니다. 경험 비활성화에 대한 자세한 내용은 구매자 안내서의 [프라이빗 마켓플레이스 구성을 참조하세요](#). AWS Marketplace

경험을 아카이브하려면 다음 코드 예제와 같이 RestrictExperience 변경 유형을 StartChangeSet 작업과 함께 사용합니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "RestrictExperience",
      "DetailsDocument":
      {},
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example"
      }
    }
  ]
}
```

경험을 다시 활성화하려면 다음 코드 예제와 같이 `ReviveExperience` 변경 유형을 `StartChangeSet` 작업과 함께 사용합니다.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReviveExperience",
      "DetailsDocument": {},
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example"
      }
    }
  ]
}
```

프라이빗 마켓플레이스 API의 오류

다음 오류는 카탈로그 API의 프라이빗 AWS Marketplace마켓플레이스 작업에만 해당됩니다.

유형 변경	오류 코드	오류 메시지	설명
Errors returned directly by the StartChangeSet action			
모두	422	문서가 잘못된 JSON 형식임	잘못된 JSON 입력이 사용되었습니다. 구문을 확인하세요.
AllowProductProcurement, DenyProductProcurement	422	Ids 배열의 값은 고유해야 합니다.	단일 변경 요청에 동일한 제품을 여러 번 포함할 수 없습니다.

유형 변경	오류 코드	오류 메시지	설명
AllowProductProcurement, DenyProductProcurement	422	Ids 배열의 누적 값 수는 50보다 작거나 같아야 합니다.	단일 변경 요청에서 최대 50개의 제품을 허용하거나 거부할 수 있습니다.

Errors found by calling the DescribeChangeSet action

CreateBrandingSettings, UpdateBrandingSettings	INVALID_URL	입력 URL에서 이미지를 가져올 수 없음	의 로고 필드에 대해 유효하고 연결 가능한 URL을 지정해야 합니다BrandingSettings .
CreateBrandingSettings, UpdateBrandingSettings	INVALID_IMAGE	유형, 콘텐츠 또는 파일 크기에 대한 이미지 확인이 실패했습니다. 크기가 500KB 이하인 .png 및 .svg 파일 유형만 지원됩니다.	이미지 파일은 브랜딩 설정에 대한 로고 요구 사항과 일치해야 합니다.
AllowProductProcurement, DenyProductProcurement	ENTITY_NOT_FOUND	Experience에서 조달 정책 누락	제품을 허용하거나 거부ProcurementPolicy 하기 전에 생성해야 합니다.
CreateProcurementPolicy	ENTITY_ALREADY_EXISTS	Experience에 대한 조달 정책이 있습니다.	프라이빗 마켓플레이스에 대한 단일 조달 정책만 가질 수 있습니다.
UpdateProcurementPolicy	ENTITY_NOT_FOUND	Experience에서 조달 정책 누락	조달 정책을 업데이트ProcurementPolicy 하기 전에 생성해야 합니다.

유형 변경	오류 코드	오류 메시지	설명
CreateBrandingSettings	ENTITY_ALREADY_EXISTS	Experience에 대한 브랜딩 설정이 존재합니다.	프라이빗 마켓플레이스에 대한 단일 브랜딩 설정만 가질 수 있습니다.
UpdateBrandingSettings	ENTITY_NOT_FOUND	Experience에서 누락된 브랜딩 설정	브랜딩 설정을 업데이트하기 전에 BrandingSettings 개체를 생성해야 합니다.
AssociateAudience	CALLER_NOT_AUTHORIZED	발신자가 작업을 실행할 권한이 없음	작업을 호출할 권한이 있어야 합니다. 추가되는 계정은 동일한 조직에 있어야 합니다.
CreateExperience	CALLER_NOT_AUTHORIZED	발신자가 경험을 생성할 권한이 없습니다.	경험을 생성할 수 있는 권한이 있어야 합니다.
AssociateAudience	ENTITY_ALREADY_EXISTS	경험이 이미 {accountId} 계정과 연결되어 있습니다. 업데이트하기 전에 이전 경험 연결 해제	단일 환경만 계정과 연결할 수 있습니다. 새 환경을 연결하기 전에 현재 환경의 연결을 해제합니다.
AssociateAudience, DisassociateAudience	ENTITY_IN_USE	선택한 계정에 대해 이미 충돌하는 변경 사항이 진행 중입니다. 나중에 다시 시도	연결을 변경하기 위한 다른 변경 요청이 이미 진행 중인 동안에는 계정과의 연결을 변경할 수 없습니다.

프라이빗 마켓플레이스에서 정의한 개체 유형

다음 표에는 각각을 지정할 수 있는 프라이빗 마켓플레이스 엔터티 유형, 용도 및 작업이 나열되어 있습니다. 각 엔터티 유형을 사용하여 (IAM) 정책에서 사용할 수 있는 리소스 Amazon 리소스 이름AWS

Identity and Access Management(ARN)을 지정할 수 있습니다. ARN 형식에 대한 자세한 내용은 섹션을 참조하세요 [the section called “카탈로그 API 엔터티”](#).

개체	용도	작업
Experience	프라이빗 마켓플레이스에 대한 최상위 설정을 저장합니다.	StartChangeSet DescribeEntity
BrandingSettings	프라이빗 마켓플레이스의 브랜딩 설정을 저장합니다.	DescribeEntity
ProcurementPolicy	조달 설정 및 제품 목록을 프라이빗 마켓플레이스에 저장합니다.	DescribeEntity
Audience	프라이빗 마켓플레이스와 연결된 보안 주체의 세부 정보를 저장합니다.	DescribeEntity

AWS Marketplace API를 사용하여 리소스 공유

AWS Marketplace Catalog API 는 AWS Resource Access Manager (AWS RAM)와 통합되어 리소스 공유를 활성화합니다. 리소스는 제품 AWS Marketplace, 제안 또는 재판매 권한 부여와 같이 사용자가 작업할 수 있는 엔터티입니다. AWS RAM를 사용하면 일부 AWS Marketplace Catalog API 리소스를 다른 리소스와 공유할 수 있습니다 AWS 계정. 리소스 공유를 생성하여 소유한 리소스를 공유할 수 있습니다. 리소스 공유는 공유할 리소스와 공유할 소비자를 지정합니다.

내용

- [AWS Marketplace 엔터티를 공유하기 위한 사전 조건](#)
- [AWS Marketplace 개체 공유](#)

AWS Marketplace 엔터티를 공유하기 위한 사전 조건

에서 엔터티를 공유 AWS Marketplace Catalog API하기 전에 다음 사전 조건을 충족해야 합니다.

- AWS Marketplace 엔터티에 하나의 리소스 정책만 연결할 수 있습니다.
- 개체를 공유하려면 해당 AWS Marketplace 개체를 소유해야 합니다 AWS 계정. 이 요구 사항은 계정에 엔터티를 할당하거나 프로비저닝해야 함을 의미합니다. 공유된 AWS Marketplace 엔터티는 공유할 수 없습니다.

AWS Marketplace 개체 공유

AWS Marketplace 리소스 공유를 통해 엔터티 소유자는 AWS 계정 의 다른 엔터티와 엔터티를 공유할 수 있습니다 AWS Marketplace. 개체 소유자는 ISVs 및 채널 파트너일 수 있습니다. 공유할 수 있는 엔터티는 제품, 제안 및 재판매 권한 부여입니다.

Note

현재 엔터티만 공유할 수 있습니다. 의 개체에는 AmiProduct, Audience, BrandingSettings, ContainerProduct, 및 Experience가 AWS Marketplace 포함됩니다ProcurementPolicy.

에 대한 자세한 내용은 [AWS RAM 사용 설명서를](#) AWS RAM참조하세요. 공유 리소스 관리에 대한 자세한 내용은 AWS RAM 사용 설명서의 [공유 AWS 리소스 사용](#)을 참조하세요.

공유 계정은 공유하려는 리소스에 대해 읽기 전용 또는 읽기/쓰기를 모두 설정할 수 있습니다. 이러한 권한은 소비 계정이 공유되는 리소스에 대해 수행할 수 있는 작업을 결정합니다.

- 공유 계정 - 공유되고 AWS RAM 관리자가 사용하여 리소스 공유를 생성하는 AWS 리소스입니다 AWS RAM.
- 계정 사용 - 리소스가 공유되는 AWS 계정입니다. 리소스 공유에는 계정 전체를 보안 주체로 지정하거나, 일부 리소스 유형의 경우 계정의 개별 역할 또는 사용자를 주체로 지정할 수 있습니다.

AWS Marketplace 개체를 공유하려면 리소스 공유에 추가해야 합니다. 리소스 공유는 AWS 계정 전반에서 리소스를 공유할 수 있게 해주는 AWS RAM 리소스입니다. 리소스 공유는 공유할 리소스와 공유 대상 소비자를 지정합니다. AWS Marketplace 콘솔을 사용하여 개체를 공유할 때 기존 리소스 공유에 추가합니다. 새 리소스 공유에 AWS Marketplace 개체를 추가하려면 먼저 [AWS RAM 콘솔](#)을 사용하여 리소스 공유를 생성해야 합니다.

AWS Marketplace 콘솔, AWS RAM 콘솔 또는 AWS Command Line Interface ()를 사용하여 소유한 AWS Marketplace 개체를 공유할 수 있습니다 AWS CLI.

AWS RAM 콘솔을 사용하여 소유한 AWS Marketplace 엔터티를 공유하려면

AWS RAM 사용 설명서의 [리소스 공유 생성](#)을 참조하세요.

를 사용하여 소유한 AWS Marketplace 엔터티를 공유하려면 AWS CLI

[create-resource-share](#) 명령을 사용합니다.

Note

리소스 기반 정책을 지원하는 엔터티와 같은 리소스 유형의 경우 AWS RAM 를 사용하여 리소스를 공유하여 추가 AWS RAM 기능을 사용할 수 있습니다. 자세한 내용은 AWS RAM 사용 설명서의 [리소스 기반 정책](#)을 참조하세요. AWS RAM 는 AWS Marketplace Catalog API 를 사용하여 리소스 공유의 권한에서 리소스 정책을 자동으로 구성하고 해당 리소스 정책을 관리합니다.

를 통해 AWS Marketplace 엔터티에 대한 AWS 리소스 기반 정책을 설정, 확인 또는 삭제하는 방법에 대한 자세한 내용은 AWS RAM 사용 설명서의 [모든 리소스에 대한 작업 허용](#)을 AWS RAM참조하세요.

AWS RAM 와를 통한 엔터티 공유의 차이점 AWS Marketplace Catalog API

를 통해 엔터티를 공유하는 것 외에도 AWS RAM를 통해 엔터티에 대한 AWS 리소스 기반 정책을 설정, 확인 또는 삭제할 수도 있습니다 AWS Marketplace Catalog API. 그러나를 통해 개체를 공유하는 AWS RAM 것과를 통해 개체를 공유하는 것에는 몇 가지 차이점이 있습니다 AWS Marketplace Catalog API.

다음은 통해 개체를 공유하는 경우 AWS RAM:

- 엔터티를 외부에 있는 계정과 공유하는 경우 AWS Organizations소비 계정은 먼저 공유 요청을 수락해야 엔터티가 공유됩니다.
- 소비 계정은가 ListEntities로 OwnershipType 설정된를 통해 공유 엔터티를 검색할 수 있습니다 SHARED.
- 여러 리소스 할당량을 준수해야 합니다. 자세한 내용은 AWS RAM 사용 안내서의 [Service quotas for AWS RAM](#)를 참조하세요.

를 통해 개체를 공유하는 경우 AWS Marketplace Catalog API:

- PutResourcePolicy 요청이 성공하는 즉시 소비 계정의 입력 없이 엔터티가 공유됩니다.
- 소비 계정은가 ListEntities로 OwnershipType 설정된를 통해 공유 엔터티를 검색할 수 없습니다 SHARED. 대신 공유 계정 소유자는 소비 계정에 공유 엔터티 ID를 알려야 합니다.

Note

사용 사례에 AWS RAM 서비스 할당량을 초과할 수 있는 리소스 공유가 필요하거나 소비 계정의 직접 입력 없이 리소스를 공유하려는 경우를 통해 공유하는 것이 좋습니다 AWS Marketplace Catalog API. 다른 모든 사용 사례의 경우를 사용하여 AWS Marketplace 리소스를 공유하는 AWS RAM 것이 좋습니다.

다음 섹션에서는를 통해 엔터티에 대한 리소스 기반 정책을 설정, 확인 또는 삭제 AWS 하는 방법을 자세히 설명합니다 AWS Marketplace Catalog API.

주제

- [리소스에 읽기 전용 정책 연결](#)
- [리소스에 읽기 및 쓰기 리소스 정책 연결](#)

- [리소스에 설정된 리소스 정책 보기](#)
- [리소스에서 리소스 정책 삭제](#)
- [사용자가 소유하고 공유한 모든 리소스 보기](#)

리소스에 읽기 전용 정책 연결

공유 계정을 사용하여 공유 리소스에 대한 읽기 전용 리소스 기반 정책을 생성할 수 있습니다. 이 정책을 사용하면 보안 주체는 자신과 공유되는 리소스의 세부 정보만 볼 수 있습니다.

요청

```
POST /PutResourcePolicy HTTP/1.1
Content-type: application/json

{
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6",
  "Policy": {
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222233334444:root"
      },
      "Action": [
        "aws-marketplace:DescribeEntity"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6"
      ]
    }
  }
}
```

응답

```
HTTP/1.1 200
Content-type: application/json

{}
```

리소스에 읽기 및 쓰기 리소스 정책 연결

공유 계정은 공유 리소스에 대한 읽기 및 쓰기 리소스 기반 정책을 생성할 수 있습니다. 이 정책을 사용하면 보안 주체가 세부 정보를 보고 공유된 리소스에 대한 쓰기 작업을 수행할 수 있습니다.

요청

```
POST /PutResourcePolicy HTTP/1.1
Content-type: application/json

{
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6",
  "Policy": {
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222233334444:root"
      },
      "Action": [
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6"
      ]
    }
  }
}
```

응답

```
HTTP/1.1 200
Content-type: application/json

{}
```

리소스에 설정된 리소스 정책 보기

공유 계정은 공유 리소스에 설정된 리소스 정책을 볼 수 있습니다.

요청

```
POST /GetResourcePolicy HTTP/1.1
Content-type: application/json
```

```
{
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
  AmiProduct/example2-abcd-1234-5ef6"
}
```

응답

```
HTTP/1.1 200
Content-type: application/json
```

```
{
  "Policy": {
    "Version": "2012-10-17",
    "Statement": {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222233334444:root"
      },
      "Action": [
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
        AmiProduct/example2-abcd-1234-5ef6"
      ]
    }
  }
}
```

리소스에서 리소스 정책 삭제

공유 계정은 공유 리소스에 설정된 리소스 정책을 삭제할 수 있습니다.

요청


```
POST /DeleteResourcePolicy HTTP/1.1
Content-type: application/json

{
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6"
}
```

응답

```
HTTP/1.1 200
Content-type: application/json

{}
```

사용자가 소유하고 공유한 모든 리소스 보기

소비 계정은 공유된 리소스를 볼 수 있습니다.

Note

리소스를 공유한 경우에만 공유된 리소스를 볼 수 있습니다 AWS RAM.

요청

```
POST /ListEntities HTTP/1.1
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "FilterList": [
    {
      "Name": "EntityId",
      "ValueList": [ "example2-abcd-1234-5ef6" ]
    }
  ],
  "OwnershipType": "SHARED"
}
```

응답

```
HTTP/1.1 200
Content-type: application/json

{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
AmiProduct/example2-abcd-1234-5ef6",
      "EntityId": "example2-abcd-1234-5ef6",
      "EntityType": "AmiProduct",
      "LastModifiedDate": "2018-02-27T13:45:22Z",
      "Name": "TestProduct",
      "Visibility": "public"
    }
  ],
  "NextToken": ""
}
```

AWSSDK에서이 서비스 사용

AWS소프트웨어 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 CLI코드 예제
AWS SDK for Go	AWS SDK for Go코드 예제
AWS SDK for Java	AWS SDK for Java코드 예제
AWS SDK for JavaScript	AWS SDK for JavaScript코드 예제
AWS SDK for Kotlin	AWS SDK for Kotlin코드 예제
AWS SDK for .NET	AWS SDK for .NET코드 예제
AWS SDK for PHP	AWS SDK for PHP코드 예제
AWS Tools for PowerShell	AWS Tools for PowerShell코드 예제
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3)코드 예제
AWS SDK for Ruby	AWS SDK for Ruby코드 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 사용 가능 여부

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

AWSSDKs AWS Marketplace사용에 대한 코드 예제

다음 코드 예제에서는 AWS소프트웨어 개발 키트(SDK)와 AWS Marketplace함께를 사용하는 방법을 보여줍니다.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

코드 예시

- [AWSSDKs를 사용한 AWS Marketplace카탈로그 API의 코드 예제](#)
 - [AWS Marketplace카탈로그 API용 AMI 제품](#)
 - [AWSSDK를 사용하여 기존 AMI 제품에 차원을 추가하고 제안 요금 조건을 업데이트합니다.](#)
 - [AWSSDK를 사용하여 AMI 제품이 배포되는 리전 추가](#)
 - [AWSSDK를 사용하여 시간당 연간 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 시간당 월별 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 시간당 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 공개 제안 초안을 사용하여 AMI 제품 초안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품이 배포되는 리전 제한](#)
 - [AWSSDK를 사용하여 제품 가시성 제한](#)
 - [AWSSDK를 사용하여 AMI 자산을 새 리전에 배포할지 여부를 지정합니다.](#)
 - [AWS Marketplace카탈로그 API에 대한 채널 파트너 제안](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 초안 CPPO 생성](#)
 - [AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 재판매 권한 부여 대체 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 채널 파트너가 생성한 모든 CPPOs 나열](#)
 - [AWSSDK를 사용하여 채널 파트너가 사용할 수 있는 모든 공유 재판매 권한 나열](#)
 - [AWSSDK를 사용하여 CPPO 게시 및 구매자 EULA 추가](#)
 - [일회성 재판매 권한 부여를 사용하여 CPPO 게시 및 AWSSDK를 사용하여 가격 인상 업데이트](#)
 - [AWSSDK를 사용하여 CPPO 초안 게시 및 가격 인상 업데이트](#)
 - [AWSSDK를 사용하여 CPPO의 만료 날짜 업데이트](#)
 - [AWS Marketplace카탈로그 API용 컨테이너 제품](#)

- [AWSSDK를 사용하여 공개 제안 초안을 사용하여 컨테이너 제품 초안 생성](#)
- [AWSSDK를 사용하여 공개 제안, 계약 요금으로 제한된 컨테이너 제품 생성](#)
- [AWS Marketplace카탈로그 API의 개체](#)
 - [AWSSDK를 사용하여 단일 호출의 모든 엔터티 설명](#)
 - [AWSSDK를 사용하여 제품과 관련된 모든 제안 나열 및 설명](#)
- [AWS Marketplace카탈로그 API에 대한 제안](#)
 - [AWSSDK를 사용하여 SaaS 제품의 사용자 지정 차원 생성 및 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 또는 SaaS 제품에 대한 비공개 제안 초안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품에 대한 계약 및 Pay-As-You-Go 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품에 대한 계약 요금 및 유연한 결제 일정으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 컨테이너 제품의 계약 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품에 대한 계약 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품에 대한 시간당 연간 요금 및 유연한 결제 일정으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품의 시간당 연간 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품의 시간당 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품에 대한 계층형 계약 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 공개 무료 평가판 제안 생성](#)
 - [AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 대체 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 공개 제안 설명](#)
 - [AWSSDK를 사용하여 비공개 제안 만료](#)
 - [AWSSDK를 사용하여 모든 비공개 제안 나열](#)
 - [AWSSDK를 사용하여 특정 제품 ID에 대해 릴리스된 공개 및 비공개 제안 나열](#)
 - [AWSSDK를 사용하여 Pay-As-You-Go 요금 계약을 적용하도록 제안 업데이트](#)
 - [AWSSDK를 사용하여 시간당 연간 요금을 적용하도록 제안 업데이트](#)
 - [AWSSDK를 사용하여 특정 지리적 리전에 대상 지정을 적용하도록 제안 업데이트](#)
 - [AWSSDK를 사용하여 공개 제안의 이름 및 설명 업데이트](#)
 - [AWSSDK를 사용하여 제안의 EULA 업데이트](#)
 - [AWSSDK를 사용하여 비공개 제안의 만료 날짜를 미래 날짜로 업데이트](#)

- [AWSSDK를 사용하여 SaaS 제품에 대한 공개 무료 평가판 제안의 무료 평가판 기간 업데이트](#)
- [AWSSDK를 사용하여 제안의 환불 정책 업데이트](#)
- [AWS Marketplace 카탈로그 API용 제품](#)
 - [AWSSDK를 사용하여 AMI, SaaS 또는 컨테이너 제품 설명](#)
 - [AWSSDK를 사용하여 모든 AMI, SaaS 또는 컨테이너 제품 및 관련 공개 제안 나열](#)
- [AWS Marketplace 카탈로그 API에 대한 재판매 권한 부여](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 초안 재판매 권한 부여 생성](#)
 - [AWSSDK를 사용하여 재판매 권한 부여 설명](#)
 - [AWSSDK를 사용하여 비공개 제안으로 일회성 재판매 권한 부여 게시](#)
 - [AWSSDK를 사용하여 만료 날짜로 다중 사용 재판매 권한 부여 게시](#)
 - [AWSSDK를 사용하여 구매자에게 보낼 만료 날짜 및 EULA와 함께 다중 사용 재판매 권한 부여 게시](#)
 - [만료 날짜가 포함된 다중 사용 재판매 권한 부여 게시 및 AWSSDK를 사용하여 리셀러 계약 설명서 추가](#)
 - [만료와 함께 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.](#)
 - [AWSSDK를 사용하여 만료 날짜 없이 다중 사용 재판매 권한 부여 게시](#)
 - [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 사용자 지정 EULA 추가](#)
 - [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 리셀러 계약 문서 추가](#)
 - [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.](#)
 - [AWSSDK를 사용하여 일회성 재판매 권한 부여 게시 및 유연한 결제 일정 추가](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 EULA 추가](#)
 - [SDK를 사용하여 일회성 재판매 권한 부여를 게시하고 재판매에 대한 특정 구매자 계정을 추가합니다.](#)[AWS](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 리셀러 계약 문서 추가](#)
 - [에 대한 일회성 재판매 권한 부여 게시 및 AWSSDK를 사용한 갱신 여부 추가](#)
 - [AWSSDK를 사용하여 재판매 권한 부여 제한](#)

- [AWSSDK를 사용하여 일회성 또는 다목적 재판매 권한 부여의 이름 및 설명 업데이트](#)
- [AWS Marketplace카탈로그 API용 SaaS 제품](#)
 - [AWSSDK를 사용하여 공개 제안 초안을 사용하여 SaaS 제품 초안 생성](#)
 - [AWSSDK를 사용하여 계약 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 Pay-As-You-Go 요금 계약으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 구독 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품 및 관련 공개 제안 게시](#)
 - [AWSSDK를 사용하여 기존 초안에서 SaaS 제품 및 관련 공개 제안 게시](#)
 - [AWSSDK를 사용하여 AMI 또는 SaaS 제품의 차원 업데이트](#)
- [AWS Marketplace카탈로그 API의 유틸리티](#)
 - [AWSSDK를 사용하여 변경 세트를 시작하는 유틸리티](#)
- [AWSSDKs를 사용한 AWS Marketplace계약 API의 코드 예제](#)
- [계약 API에 대한 AWS Marketplace계약](#)
 - [AWSSDK를 사용하여 계약 취소 요청 수락](#)
 - [AWSSDK를 사용하여 계약 결제 요청 수락](#)
 - [AWSSDK를 사용하여 SaaS 계약 갱신 기간 수정](#)
 - [AWSSDK를 사용하여 사용량 요금 모델에 대한 AMI ConfigurableUpfrontPricingTerm 수정](#)
 - [AWSSDK를 사용하여 새 AMI 무료 평가판 계약 생성](#)
 - [AWSSDK를 사용하여 선결제로 새 SaaS 계약 생성](#)
 - [AWSSDK를 사용하여 모든 계약 IDs 가져오기](#)
 - [AWSSDK를 사용하여 모든 계약 가져오기](#)
 - [AWSSDK를 사용하여 계약에서 고객 ID 가져오기](#)
 - [AWSSDK를 사용하여 계약 취소 요청 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약 결제 요청 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약에서 재무 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약에서 무료 평가판 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약에 대한 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약에서 제품 및 제안 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약의 EULA 가져오기](#)

- [AWSSDK를 사용하여 계약의 자동 갱신 조건 가져오기](#)
- [AWSSDK를 사용하여 계약에서 구매한 차원 가져오기](#)
- [AWSSDK를 사용하여 계약에서 구매한 각 차원의 인스턴스 가져오기](#)
- [AWSSDK를 사용하여 계약의 결제 일정 가져오기](#)
- [AWSSDK를 사용하여 계약에서 차원당 요금 가져오기](#)
- [AWSSDK를 사용하여 계약의 요금 유형 가져오기](#)
- [AWSSDK를 사용하여 계약의 제품 유형 가져오기](#)
- [AWSSDK를 사용하여 계약 상태 가져오기](#)
- [AWSSDK를 사용하여 계약의 지원 조건 가져오기](#)
- [AWSSDK를 사용하여 계약 조건 가져오기](#)
- [AWSSDK를 사용하여 계약 취소 요청 나열](#)
- [AWSSDK를 사용하여 계약 결제 요청 나열](#)
- [AWSSDK를 사용하여 계약 취소 요청 거부](#)
- [AWSSDK를 사용하여 계약 결제 요청 거부](#)
- [AWSSDK를 사용하여 SaaS 계약 계약을 계약 기반 제안으로 대체](#)
- [AWSSDK를 사용하여 SaaS 무료 평가판을 소비 요금으로 계약으로 교체](#)
- [AWSSDK를 사용하여 SaaS 사용량 기반 요금 기간 대체 및 이전 계약 취소](#)
- [AWSSDK를 사용하여 AMI 사용량 기반 요금 기간을 바꾸고 기존 ConfigurableUpfrontPricingTerm 유지](#)
- [AWSSDK를 사용하여 계정 ID별로 계약 검색](#)
- [AWSSDK를 사용하여 계약 ID로 계약 검색](#)
- [AWSSDK를 사용하여 종료 날짜별로 계약 검색](#)
- [AWSSDK를 사용하여 제안 ID로 계약 검색](#)
- [AWSSDK를 사용하여 제품 ID별로 계약 검색](#)
- [AWSSDK를 사용하여 상태별로 계약 검색](#)
- [AWSSDK를 사용하여 하나의 사용자 지정 필터가 있는 계약 검색](#)
- [AWSSDK를 사용하여 두 개의 사용자 지정 필터가 있는 계약 검색](#)
- [AWSSDK를 사용하여 계약 수락 후 구매 주문 업데이트](#)

AWSSDKs를 사용한 AWS Marketplace카탈로그 API의 코드 예제

다음 코드 예제에서는 AWS Marketplace카탈로그 API를 AWS소프트웨어 개발 키트(SDK)와 함께 사용하는 방법을 보여줍니다.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace카탈로그 API

- [AWS Marketplace카탈로그 API용 AMI 제품](#)
 - [AWSSDK를 사용하여 기존 AMI 제품에 차원을 추가하고 제안 요금 조건을 업데이트합니다.](#)
 - [AWSSDK를 사용하여 AMI 제품이 배포되는 리전 추가](#)
 - [AWSSDK를 사용하여 시간당 연간 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 시간당 월별 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 시간당 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
 - [AWSSDK를 사용하여 공개 제안 초안을 사용하여 AMI 제품 초안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품이 배포되는 리전 제한](#)
 - [AWSSDK를 사용하여 제품 가시성 제한](#)
 - [AWSSDK를 사용하여 AMI 자산을 새 리전에 배포할지 여부를 지정합니다.](#)
- [AWS Marketplace카탈로그 API에 대한 채널 파트너 제안](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 초안 CPPO 생성](#)
 - [AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 재판매 권한 부여 대체 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 채널 파트너가 생성한 모든 CPPOs 나열](#)
 - [AWSSDK를 사용하여 채널 파트너가 사용할 수 있는 모든 공유 재판매 권한 나열](#)
 - [AWSSDK를 사용하여 CPPO 게시 및 구매자 EULA 추가](#)
 - [일회성 재판매 권한 부여를 사용하여 CPPO 게시 및 AWSSDK를 사용하여 가격 인상 업데이트](#)
 - [AWSSDK를 사용하여 CPPO 초안 게시 및 가격 인상 업데이트](#)
 - [AWSSDK를 사용하여 CPPO의 만료 날짜 업데이트](#)
- [AWS Marketplace카탈로그 API용 컨테이너 제품](#)
 - [AWSSDK를 사용하여 공개 제안 초안을 사용하여 컨테이너 제품 초안 생성](#)
 - [AWSSDK를 사용하여 공개 제안, 계약 요금으로 제한된 컨테이너 제품 생성](#)

- [AWSSDK를 사용하여 단일 호출의 모든 엔터티 설명](#)
- [AWSSDK를 사용하여 제품과 관련된 모든 제안 나열 및 설명](#)
- [AWS Marketplace 카탈로그 API에 대한 제안](#)
 - [AWSSDK를 사용하여 SaaS 제품의 사용자 지정 차원 생성 및 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 또는 SaaS 제품에 대한 비공개 제안 초안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품에 대한 계약 및 Pay-As-You-Go 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품에 대한 계약 요금 및 유연한 결제 일정으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 컨테이너 제품의 계약 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품에 대한 계약 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품에 대한 시간당 연간 요금 및 유연한 결제 일정으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품의 시간당 연간 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 AMI 제품의 시간당 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품에 대한 계층형 계약 요금으로 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 공개 무료 평가판 제안 생성](#)
 - [AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 대체 비공개 제안 생성](#)
 - [AWSSDK를 사용하여 공개 제안 설명](#)
 - [AWSSDK를 사용하여 비공개 제안 만료](#)
 - [AWSSDK를 사용하여 모든 비공개 제안 나열](#)
 - [AWSSDK를 사용하여 특정 제품 ID에 대해 릴리스된 공개 및 비공개 제안 나열](#)
 - [AWSSDK를 사용하여 Pay-As-You-Go 요금 계약을 적용하도록 제안 업데이트](#)
 - [AWSSDK를 사용하여 시간당 연간 요금을 적용하도록 제안 업데이트](#)
 - [AWSSDK를 사용하여 특정 지리적 리전에 대상 지정을 적용하도록 제안 업데이트](#)
 - [AWSSDK를 사용하여 공개 제안의 이름 및 설명 업데이트](#)
 - [AWSSDK를 사용하여 제안의 EULA 업데이트](#)
 - [AWSSDK를 사용하여 비공개 제안의 만료 날짜를 미래 날짜로 업데이트](#)
 - [AWSSDK를 사용하여 SaaS 제품에 대한 공개 무료 평가판 제안의 무료 평가판 기간 업데이트](#)
 - [AWSSDK를 사용하여 제안의 환불 정책 업데이트](#)

- [AWSSDK를 사용하여 AMI, SaaS 또는 컨테이너 제품 설명](#)
- [AWSSDK를 사용하여 모든 AMI, SaaS 또는 컨테이너 제품 및 관련 공개 제안 나열](#)
- [AWS Marketplace 카탈로그 API에 대한 재판매 권한 부여](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 초안 재판매 권한 부여 생성](#)
 - [AWSSDK를 사용하여 재판매 권한 부여 설명](#)
 - [AWSSDK를 사용하여 비공개 제안으로 일회성 재판매 권한 부여 게시](#)
 - [AWSSDK를 사용하여 만료 날짜로 다중 사용 재판매 권한 부여 게시](#)
 - [AWSSDK를 사용하여 구매자에게 보낼 만료 날짜 및 EULA와 함께 다중 사용 재판매 권한 부여 게시](#)
 - [만료 날짜가 포함된 다중 사용 재판매 권한 부여 게시 및 AWSSDK를 사용하여 리셀러 계약 설명서 추가](#)
 - [만료와 함께 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.](#)
 - [AWSSDK를 사용하여 만료 날짜 없이 다중 사용 재판매 권한 부여 게시](#)
 - [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 사용자 지정 EULA 추가](#)
 - [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 리셀러 계약 문서 추가](#)
 - [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.](#)
 - [AWSSDK를 사용하여 일회성 재판매 권한 부여 게시 및 유연한 결제 일정 추가](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 EULA 추가](#)
 - [SDK를 사용하여 일회성 재판매 권한 부여를 게시하고 재판매에 대한 특정 구매자 계정을 추가합니다.](#)
 - [AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 리셀러 계약 문서 추가](#)
 - [에 대한 일회성 재판매 권한 부여 게시 및 AWSSDK를 사용한 갱신 여부 추가](#)
 - [AWSSDK를 사용하여 재판매 권한 부여 제한](#)
 - [AWSSDK를 사용하여 일회성 또는 다목적 재판매 권한 부여의 이름 및 설명 업데이트](#)
- [AWS Marketplace 카탈로그 API용 SaaS 제품](#)
 - [AWSSDK를 사용하여 공개 제안 초안을 사용하여 SaaS 제품 초안 생성](#)
 - [AWSSDK를 사용하여 계약 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)

- [AWSSDK를 사용하여 Pay-As-You-Go 요금 계약으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 구독 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 SaaS 제품 및 관련 공개 제안 게시](#)
- [AWSSDK를 사용하여 기존 초안에서 SaaS 제품 및 관련 공개 제안 게시](#)
- [AWSSDK를 사용하여 AMI 또는 SaaS 제품의 차원 업데이트](#)
- [AWS Marketplace 카탈로그 API의 유틸리티](#)
- [AWSSDK를 사용하여 변경 세트를 시작하는 유틸리티](#)

AWS Marketplace 카탈로그 API용 AMI 제품

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 기존 AMI 제품에 차원을 추가하고 제안 요금 조건을 업데이트합니다.](#)
- [AWSSDK를 사용하여 AMI 제품이 배포되는 리전 추가](#)
- [AWSSDK를 사용하여 시간당 연간 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 시간당 월별 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 시간당 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 공개 제안 초안을 사용하여 AMI 제품 초안 생성](#)
- [AWSSDK를 사용하여 AMI 제품이 배포되는 리전 제한](#)
- [AWSSDK를 사용하여 제품 가시성 제한](#)
- [AWSSDK를 사용하여 AMI 자산을 새 리전에 배포할지 여부를 지정합니다.](#)

AWSSDK를 사용하여 기존 AMI 제품에 차원을 추가하고 제안 요금 조건을 업데이트합니다.

다음 코드 예제에서는 기존 AMI 제품에 차원을 추가하고 제안 요금 조건을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddDimensions",
      "Entity": {
        "Identifier": "prod-111111111111",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": [
        {
          "Key": "m7g.8xlarge",
          "Description": "m7g.8xlarge",
          "Name": "m7g.8xlarge",
          "Types": [
            "Metered"
          ],
          "Unit": "Hrs"
        }
      ]
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",

```

```

    "Terms": [
      {
        "Type": "UsageBasedPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
          {
            "RateCard": [
              {
                "DimensionKey": "m5.large",
                "Price": "0.15"
              },
              {
                "DimensionKey": "m7g.4xlarge",
                "Price": "0.45"
              },
              {
                "DimensionKey": "m7g.2xlarge",
                "Price": "0.45"
              },
              {
                "DimensionKey": "m7g.8xlarge",
                "Price": "0.55"
              }
            ]
          }
        ]
      }
    ]
  }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddDimensions",
      "Entity": {
        "Identifier": "prod-111111111111",
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": [
        {
          "Key": "m7g.8xlarge",
          "Description": "m7g.8xlarge",
          "Name": "m7g.8xlarge",
          "Types": [
            "Metered"
          ],
          "Unit": "Hrs"
        }
      ]
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "UsageBasedPricingTerm",
```



```

        "CurrencyCode": "USD",
        "RateCards": [
            {
                "RateCard": [
                    {
                        "DimensionKey": "m5.large",
                        "Price": "0.15"
                    },
                    {
                        "DimensionKey": "m7g.4xlarge",
                        "Price": "0.45"
                    },
                    {
                        "DimensionKey": "m7g.2xlarge",
                        "Price": "0.45"
                    },
                    {
                        "DimensionKey": "m7g.8xlarge",
                        "Price": "0.55"
                    }
                ]
            }
        ]
    }
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to add a dimension to an existing
AMI product and update the offer pricing terms.
CAPI-23
"""

import os

```

```
import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Add dimension for AMI product")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 제품이 배포되는 리전 추가

다음 코드 예제에서는 AMI 제품이 배포되는 리전을 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
```

```

    "Catalog": "AWSMarketplace",
    "ChangeSet": [
      {
        "ChangeType": "AddRegions",
        "Entity": {
          "Type": "AmiProduct@1.0",
          "Identifier": "prod-11111111111111"
        },
        "DetailsDocument": {
          "Regions": [
            "us-east-2",
            "us-west-2"
          ]
        }
      }
    ]
  }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-11111111111111"
      },
      "DetailsDocument": {

```

```

        "Regions": [
            "us-east-2",
            "us-west-2"
        ]
    }
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to add a region where my
AMI product is deployed
CAPI-25A
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Add a region where my AMI product is deployed",
    )

```

```

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 시간당 연간 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성

다음 코드 예제에서는 공개 또는 제한된 AMI 제품과 시간 단위 연간 요금이 청구되는 공개 제안을 생성하는 방법을 보여줍니다. 이 예제에서는 표준 또는 사용자 지정 EULA를 생성합니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      }
    },

```

```

        "DetailsDocument": {}
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "AmiProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "ProductTitle": "Sample product",
            "ShortDescription": "Brief description",
            "LongDescription": "Detailed description",
            "Highlights": [
                "Sample highlight"
            ],
            "SearchKeywords": [
                "Sample keyword"
            ],
            "Categories": [
                "Operating Systems"
            ],
            "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
            "VideoUrls": [
                "https://sample.amazonaws.com/awsmvp-video-1"
            ],
            "AdditionalResources": []
        }
    },
    {
        "ChangeType": "AddRegions",
        "Entity": {
            "Type": "AmiProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Regions": [
                "us-east-1"
            ]
        }
    },
    {
        "ChangeType": "AddInstanceTypes",
        "Entity": {
            "Type": "AmiProduct@1.0",

```

```

        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "InstanceTypes": [
            "t2.micro"
        ]
    }
},
{
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Version": {
            "VersionTitle": "Test AMI Version1.0",
            "ReleaseNotes": "Test AMI Version"
        },
        "DeliveryOptions": [
            {
                "Details": {
                    "AmiDeliveryOptionDetails": {
                        "AmiSource": {
                            "AmiId": "ami-1111111111111111",
                            "AccessRoleArn":
"arn:aws:iam::111111111111:role/AWSMarketplaceAmiIngestion",
                            "UserName": "ec2-user",
                            "OperatingSystemName": "AMAZONLINUX",
                            "OperatingSystemVersion": "10.0.14393",
                            "ScanningPort": 22
                        },
                        "UsageInstructions": "Test AMI Version",
                        "RecommendedInstanceType": "t2.micro",
                        "SecurityGroups": [
                            {
                                "IpProtocol": "tcp",
                                "IpRanges": [
                                    "0.0.0.0/0"
                                ],
                                "FromPort": 10,
                                "ToPort": 22
                            }
                        ]
                    }
                }
            }
        ]
    }
}

```

```

    }
  }
}
},
{
  "ChangeType": "AddDimensions",
  "Entity": {
    "Type": "AmiProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": [
    {
      "Key": "t2.micro",
      "Description": "t2.micro",
      "Name": "t2.micro",
      "Types": [
        "Metered"
      ],
      "Unit": "Hrs"
    }
  ],
},
{
  "ChangeType": "UpdateTargeting",
  "Entity": {
    "Type": "AmiProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "PositiveTargeting": {
      "BuyerAccounts": [
        "111111111111",
        "222222222222"
      ]
    }
  }
},
{
  "ChangeType": "ReleaseProduct",
  "Entity": {
    "Type": "AmiProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  }
}

```



```

    },
    "DetailsDocument": {}
  },
  {
    "ChangeType": "CreateOffer",
    "ChangeName": "CreateOfferChange",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "DetailsDocument": {
      "ProductId": "$CreateProductChange.Entity.Identifier"
    }
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Name": "Test public offer for AmiProduct using AWS Marketplace
API Reference Code",
      "Description": "Test public offer with hourly-annual pricing for
AmiProduct using AWS Marketplace API Reference Code"
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Usage",
      "Terms": [
        {
          "Type": "UsageBasedPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "RateCard": [
                {
                  "DimensionKey": "t2.micro",
                  "Price": "0.15"
                }
              ]
            }
          ]
        }
      ]
    }
  }
]

```

```

        }
    ]
}
},
{
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
        {
            "Selector": {
                "Type": "Duration",
                "Value": "P365D"
            },
            "RateCard": [
                {
                    "DimensionKey": "t2.micro",
                    "Price": "150"
                }
            ],
            "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
            }
        }
    ]
}
],
}
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
}
]
}
}

```

```

    }
  ]
}

},
{
  "ChangeType": "UpdateSupportTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "SupportTerm",
        "RefundPolicy": "Absolutely no refund, period."
      }
    ]
  }
},
{
  "ChangeType": "ReleaseOffer",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {}
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [
          "Operating Systems"
        ],
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [],
        "AdditionalResources": [],
        "SupportDescription": "For support, contact support@example.com"
      }
    },
    {
      "ChangeType": "AddRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {

```

```

        "Regions": [
            "us-east-1"
        ]
    },
    {
        "ChangeType": "AddInstanceTypes",
        "Entity": {
            "Type": "AmiProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "InstanceTypes": [
                "t2.micro"
            ]
        }
    },
    {
        "ChangeType": "AddDeliveryOptions",
        "Entity": {
            "Type": "AmiProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Version": {
                "VersionTitle": "Test AMI Version1.0",
                "ReleaseNotes": "Test AMI Version"
            },
            "DeliveryOptions": [
                {
                    "Details": {
                        "AmiDeliveryOptionDetails": {
                            "AmiSource": {
                                "AmiId": "ami-1111111111111111",
                                "AccessRoleArn":
"arn:aws:iam::111111111111:role/AWSMarketplaceAmiIngestion",
                                "UserName": "ec2-user",
                                "OperatingSystemName": "AMAZONLINUX",
                                "OperatingSystemVersion": "10.0.14393",
                                "ScanningPort": 22
                            },
                            "UsageInstructions": "Test AMI Version",
                            "RecommendedInstanceType": "t2.micro",
                            "SecurityGroups": [

```

```

        {
            "IpProtocol": "tcp",
            "IpRanges": [
                "0.0.0.0/0"
            ],
            "FromPort": 10,
            "ToPort": 22
        }
    ]
}

},
{
    "ChangeType": "AddDimensions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
        {
            "Key": "t2.micro",
            "Description": "t2.micro",
            "Name": "t2.micro",
            "Types": [
                "Metered"
            ],
            "Unit": "Hrs"
        }
    ]
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111",
                "222222222222"
            ]
        }
    }
}

```

```

        ]
      }
    },
    {
      "ChangeType": "ReleaseProduct",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test public offer for AmiProduct using AWS Marketplace
API Reference Code",
        "Description": "Test public offer with hourly-annual pricing for
AmiProduct using AWS Marketplace API Reference Code"
      }
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [

```

```

        {
            "Type": "UsageBasedPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
                {
                    "RateCard": [
                        {
                            "DimensionKey": "t2.micro",
                            "Price": "0.15"
                        }
                    ]
                }
            ]
        },
        {
            "Type": "ConfigurableUpfrontPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
                {
                    "Selector": {
                        "Type": "Duration",
                        "Value": "P365D"
                    },
                    "RateCard": [
                        {
                            "DimensionKey": "t2.micro",
                            "Price": "150"
                        }
                    ],
                    "Constraints": {
                        "MultipleDimensionSelection": "Allowed",
                        "QuantityConfiguration": "Allowed"
                    }
                }
            ]
        }
    ],
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        }
    }
}

```



```

    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "StandardEula",
              "Version": "2022-07-14"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "SupportTerm",
          "RefundPolicy": "Absolutely no refund, period."
        }
      ]
    }
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a public or limited AMI
product and public offer with hourly-annual pricing and standard or custom EULA
CAPI-06
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create limited AMI product and public offer with hourly-annual pricing
and standard EULA",
    )

    return response

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 시간당 월별 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성

다음 코드 예제에서는 공개 또는 제한된 AMI 제품과 시간 단위 월간 요금이 청구되는 공개 제안을 생성하는 방법을 보여줍니다. 이 예제에서는 표준 또는 사용자 지정 EULA를 생성합니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ]
      }
    }
  ]
}
```

```

        ],
        "SearchKeywords": [
            "Sample keyword"
        ],
        "Categories": [
            "Operating Systems"
        ],
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [
            "https://sample.amazonaws.com/awsmvp-video-1"
        ],
        "AdditionalResources": []
    }
},
{
    "ChangeType": "AddRegions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Regions": [
            "us-east-1"
        ]
    }
},
{
    "ChangeType": "AddInstanceTypes",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "InstanceTypes": [
            "t2.micro"
        ]
    }
},
{
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },

```

```

    "DetailsDocument": {
      "Version": {
        "VersionTitle": "Test AMI Version1.0",
        "ReleaseNotes": "Test AMI Version"
      },
      "DeliveryOptions": [
        {
          "Details": {
            "AmiDeliveryOptionDetails": {
              "AmiSource": {
                "AmiId": "ami-1111111111111111",
                "AccessRoleArn":
"arn:aws:iam::111111111111:role/AWSMarketplaceAmiIngestion",
                "UserName": "ec2-user",
                "OperatingSystemName": "AMAZONLINUX",
                "OperatingSystemVersion": "10.0.14393",
                "ScanningPort": 22
              },
              "UsageInstructions": "Test AMI Version",
              "RecommendedInstanceType": "t2.micro",
              "SecurityGroups": [
                {
                  "IpProtocol": "tcp",
                  "IpRanges": [
                    "0.0.0.0/0"
                  ],
                  "FromPort": 10,
                  "ToPort": 22
                }
              ]
            }
          }
        }
      ],
    },
    {
      "ChangeType": "AddDimensions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": [
        {

```

```

        "Key": "t2.micro",
        "Description": "t2.micro",
        "Name": "t2.micro",
        "Types": [
            "Metered"
        ],
        "Unit": "Hrs"
    }
]
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111",
                "222222222222"
            ]
        }
    }
},
{
    "ChangeType": "ReleaseProduct",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
},
{
    "ChangeType": "CreateOffer",
    "ChangeName": "CreateOfferChange",
    "Entity": {
        "Type": "Offer@1.0"
    },
    "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier"
    }
},
{

```

```

        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test public offer for AmiProduct using AWS Marketplace
API Reference Code",
            "Description": "Test public offer with hourly-monthly pricing for
AmiProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Usage",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.micro",
                                    "Price": "0.15"
                                }
                            ]
                        }
                    ]
                },
                {
                    "Type": "RecurringPaymentTerm",
                    "CurrencyCode": "USD",
                    "BillingPeriod": "Monthly",
                    "Price": "15.0"
                }
            ]
        }
    },
    },

```

```

    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "StandardEula",
                "Version": "2022-07-14"
              }
            ]
          }
        ]
      }
    },
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "SupportTerm",
            "RefundPolicy": "Absolutely no refund, period."
          }
        ]
      }
    },
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {}
    }
  ]

```



```
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [

```

```

        "Operating Systems"
    ],
    "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
    "VideoUrls": [
        "https://sample.amazonaws.com/awsmvp-video-1"
    ],
    "AdditionalResources": []
},
{
    "ChangeType": "AddRegions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Regions": [
            "us-east-1"
        ]
    }
},
{
    "ChangeType": "AddInstanceTypes",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "InstanceTypes": [
            "t2.micro"
        ]
    }
},
{
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Version": {
            "VersionTitle": "Test AMI Version1.0",
            "ReleaseNotes": "Test AMI Version"
        }
    }
},

```

```

        "DeliveryOptions": [
            {
                "Details": {
                    "AmiDeliveryOptionDetails": {
                        "AmiSource": {
                            "AmiId": "ami-1111111111111111",
                            "AccessRoleArn":
"arn:aws:iam::111111111111:role/AWSMarketplaceAmiIngestion",
                            "UserName": "ec2-user",
                            "OperatingSystemName": "AMAZONLINUX",
                            "OperatingSystemVersion": "10.0.14393",
                            "ScanningPort": 22
                        },
                        "UsageInstructions": "Test AMI Version",
                        "RecommendedInstanceType": "t2.micro",
                        "SecurityGroups": [
                            {
                                "IpProtocol": "tcp",
                                "IpRanges": [
                                    "0.0.0.0/0"
                                ],
                                "FromPort": 10,
                                "ToPort": 22
                            }
                        ]
                    }
                }
            }
        ],
        {
            "ChangeType": "AddDimensions",
            "Entity": {
                "Type": "AmiProduct@1.0",
                "Identifier": "$CreateProductChange.Entity.Identifier"
            },
            "DetailsDocument": [
                {
                    "Key": "t2.micro",
                    "Description": "t2.micro",
                    "Name": "t2.micro",
                    "Types": [
                        "Metered"
                    ]
                }
            ]
        }
    ]
}

```

```

        ],
        "Unit": "Hrs"
    }
}
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111",
                "222222222222"
            ]
        }
    }
},
{
    "ChangeType": "ReleaseProduct",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
},
{
    "ChangeType": "CreateOffer",
    "ChangeName": "CreateOfferChange",
    "Entity": {
        "Type": "Offer@1.0"
    },
    "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier"
    }
},
{
    "ChangeType": "UpdateInformation",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },

```

```

        "DetailsDocument": {
            "Name": "Test public offer for AmiProduct using AWS Marketplace
API Reference Code",
            "Description": "Test public offer with hourly-monthly pricing for
AmiProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Usage",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.micro",
                                    "Price": "0.15"
                                }
                            ]
                        }
                    ]
                },
                {
                    "Type": "RecurringPaymentTerm",
                    "CurrencyCode": "USD",
                    "BillingPeriod": "Monthly",
                    "Price": "15.0"
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        }
    }
]

```

```

    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "StandardEula",
              "Version": "2022-07-14"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "SupportTerm",
          "RefundPolicy": "Absolutely no refund, period."
        }
      ]
    }
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a public or limited AMI
product and public offer with hourly-monthly pricing and standard or custom EULA
CAPI-08
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set,
        "create limited AMI product and public offer with hourly-monthly pricing
and standard EULA",
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 시간당 요금으로 공개 또는 제한된 AMI 제품 및 공개 제안 생성

다음 코드 예제에서는 공개 또는 제한된 AMI 제품과 시간 단위 요금이 청구되는 공개 제안을 생성하는 방법을 보여줍니다. 이 예제에서는 표준 또는 사용자 지정 EULA를 생성합니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [
          "Operating Systems"
        ]
      }
    }
  ]
}
```



```

        ],
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [
            "https://sample.amazonaws.com/awsmvp-video-1"
        ],
        "AdditionalResources": []
    }
},
{
    "ChangeType": "AddRegions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Regions": [
            "us-east-1"
        ]
    }
},
{
    "ChangeType": "AddInstanceTypes",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "InstanceTypes": [
            "t2.micro"
        ]
    }
},
{
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Version": {
            "VersionTitle": "Test AMI Version1.0",
            "ReleaseNotes": "Test AMI Version"
        },
        "DeliveryOptions": [

```

```

        {
            "Details": {
                "AmiDeliveryOptionDetails": {
                    "AmiSource": {
                        "AmiId": "ami-1111111111111111",
                        "AccessRoleArn":
"arn:aws:iam::111111111111:role/AWSMarketplaceAmiIngestion",
                        "UserName": "ec2-user",
                        "OperatingSystemName": "AMAZONLINUX",
                        "OperatingSystemVersion": "10.0.14393",
                        "ScanningPort": 22
                    },
                    "UsageInstructions": "Test AMI Version",
                    "RecommendedInstanceType": "t2.micro",
                    "SecurityGroups": [
                        {
                            "IpProtocol": "tcp",
                            "IpRanges": [
                                "0.0.0.0/0"
                            ],
                            "FromPort": 10,
                            "ToPort": 22
                        }
                    ]
                }
            }
        },
        {
            "ChangeType": "AddDimensions",
            "Entity": {
                "Type": "AmiProduct@1.0",
                "Identifier": "$CreateProductChange.Entity.Identifier"
            },
            "DetailsDocument": [
                {
                    "Key": "t2.micro",
                    "Description": "t2.micro",
                    "Name": "t2.micro",
                    "Types": [
                        "Metered"
                    ]
                }
            ],

```

```

        "Unit": "Hrs"
      }
    ]
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "AmiProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111",
          "222222222222"
        ]
      }
    }
  },
  {
    "ChangeType": "ReleaseProduct",
    "Entity": {
      "Type": "AmiProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  },
  {
    "ChangeType": "CreateOffer",
    "ChangeName": "CreateOfferChange",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "DetailsDocument": {
      "ProductId": "$CreateProductChange.Entity.Identifier"
    }
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {

```

```

        "Name": "Test public offer for AmiProduct using AWS Marketplace
API Reference Code",
        "Description": "Test public offer with hourly pricing for
AmiProduct using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
            {
                "Type": "UsageBasedPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "RateCard": [
                            {
                                "DimensionKey": "t2.micro",
                                "Price": "0.15"
                            }
                        ]
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {

```

```

        "Type": "StandardEula",
        "Version": "2022-07-14"
      }
    ]
  },
  {
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "SupportTerm",
          "RefundPolicy": "Absolutely no refund, period."
        }
      ]
    }
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [
          "Operating Systems"
        ],
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [
```

```

        "https://sample.amazonaws.com/awssmp-video-1"
    ],
    "AdditionalResources": []
  }
},
{
  "ChangeType": "AddRegions",
  "Entity": {
    "Type": "AmiProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Regions": [
      "us-east-1"
    ]
  }
},
{
  "ChangeType": "AddInstanceTypes",
  "Entity": {
    "Type": "AmiProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "InstanceTypes": [
      "t2.micro"
    ]
  }
},
{
  "ChangeType": "AddDeliveryOptions",
  "Entity": {
    "Type": "AmiProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Version": {
      "VersionTitle": "Test AMI Version1.0",
      "ReleaseNotes": "Test AMI Version"
    },
    "DeliveryOptions": [
      {
        "Details": {
          "AmiDeliveryOptionDetails": {

```

```

        "AmiSource": {
            "AmiId": "ami-1111111111111111",
            "AccessRoleArn":
"arn:aws:iam::111111111111:role/AWSMarketplaceAmiIngestion",
            "UserName": "ec2-user",
            "OperatingSystemName": "AMAZONLINUX",
            "OperatingSystemVersion": "10.0.14393",
            "ScanningPort": 22
        },
        "UsageInstructions": "Test AMI Version",
        "RecommendedInstanceType": "t2.micro",
        "SecurityGroups": [
            {
                "IpProtocol": "tcp",
                "IpRanges": [
                    "0.0.0.0/0"
                ],
                "FromPort": 10,
                "ToPort": 22
            }
        ]
    }
},
{
    "ChangeType": "AddDimensions",
    "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
        {
            "Key": "t2.micro",
            "Description": "t2.micro",
            "Name": "t2.micro",
            "Types": [
                "Metered"
            ],
            "Unit": "Hrs"
        }
    ]
}
]

```



```

    },
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "BuyerAccounts": [
            "111111111111",
            "222222222222"
          ]
        }
      }
    },
    {
      "ChangeType": "ReleaseProduct",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test public offer for AmiProduct using AWS Marketplace
API Reference Code",

```

```

        "Description": "Test public offer with hourly pricing for
        AmiProduct using AWS Marketplace API Reference Code"
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Usage",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.micro",
                                    "Price": "0.15"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "LegalTerm",
                    "Documents": [
                        {
                            "Type": "StandardEula",
                            "Version": "2022-07-14"
                        }
                    ]
                }
            ]
        }
    }
]

```

```

    }
  ]
}
},
{
  "ChangeType": "UpdateSupportTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "SupportTerm",
        "RefundPolicy": "Absolutely no refund, period."
      }
    ]
  },
},
{
  "ChangeType": "ReleaseOffer",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to Create a public or limited AMI
product
and public offer with hourly pricing and standard or custom EULA
CAPI-07

```

```

"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create limited AMI product and public offer with hourly pricing and
standard EULA",
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 공개 제안 초안을 사용하여 AMI 제품 초안 생성

다음 코드 예제에서는 공개 제안 초안으로 AMI 제품 초안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product"
      }
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier",
        "Name": "Test Offer"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "AmiProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier",
        "Name": "Test Offer"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
```

Shows how to use the AWS SDK for Python (Boto3) to create an AMI draft product with a draft public offer.

CAPI-02

"""

```
import os
```

```
import utils.start_changeset as sc
```

```
import utils.stringify_details as sd
```

```
def main(change_set=None):
```

```
    if change_set is None:
```

```
        fname = "changeset.json"
```

```
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
```

```
        stringified_change_set = sd.stringify_changeset(change_set_file)
```

```
    else:
```

```
        stringified_change_set = change_set
```

```
    response = sc.usage_demo(
```

```
        stringified_change_set,
```

```
        "AMI draft product with draft public offer",
```

```
    )
```

```
    return response
```

```
if __name__ == "__main__":
```

```
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 제품이 배포되는 리전 제한

다음 코드 예제에서는 AMI 제품이 배포되는 리전을 제한하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "Regions": [
          "us-west-2"
        ]
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "Regions": [
          "us-west-2"
        ]
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to restrict a region where my
AMI product is deployed
CAPI-25B
"""
```

```

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Add a region where my AMI product is deployed",
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 제품 가시성 제한

다음 코드 예제에서는 제품 가시성을 제한하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateVisibility",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-1111111111111"
      },
      "DetailsDocument": {
        "TargetVisibility": "Restricted"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateVisibility",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-11111111111111"
      },
      "DetailsDocument": {
        "TargetVisibility": "Restricted"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to change a product visibility to
restricted
CAPI-17
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Restrict existing AMI")
```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 자산을 새 리전에 배포할지 여부를 지정합니다.

다음 코드 예제에서는 향후 리전을 지원하기 AWS위해에서 빌드한 새 리전에 AMI 자산을 배포할지 여부를 지정하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateFutureRegionSupport",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "FutureRegionSupport": {
          "SupportedRegions": [
            "All"
          ]
        }
      }
    }
  ]
}
```

```

    }
  }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateFutureRegionSupport",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "FutureRegionSupport": {
          "SupportedRegions": [
            "All"
          ]
        }
      }
    }
  ]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to modify a product to support
all future regions
CAPI-26
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Update future region support")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API에 대한 채널 파트너 제안

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 모든 제품 유형에 대한 초안 CPPO 생성](#)
- [AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 재판매 권한 부여 대체 비공개 제안 생성](#)

- [AWSSDK를 사용하여 채널 파트너가 생성한 모든 CPPOs 나열](#)
- [AWSSDK를 사용하여 채널 파트너가 사용할 수 있는 모든 공유 재판매 권한 나열](#)
- [AWSSDK를 사용하여 CPPO 게시 및 구매자 EULA 추가](#)
- [일회성 재판매 권한 부여를 사용하여 CPPO 게시 및 AWSSDK를 사용하여 가격 인상 업데이트](#)
- [AWSSDK를 사용하여 CPPO 초안 게시 및 가격 인상 업데이트](#)
- [AWSSDK를 사용하여 CPPO의 만료 날짜 업데이트](#)

AWSSDK를 사용하여 모든 제품 유형에 대한 초안 CPPO 생성

다음 코드 예제에서는 구매자를 대상으로 게시하기 전에 내부적으로 검토할 수 있도록 모든 제품 유형에 대한 CPPO 초안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "11111111-1111-1111-1111-111111111111",
        "Name": "Test Offer",
        "Description": "Test product"
      }
    }
  ]
}
```



```
]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "11111111-1111-1111-1111-111111111111",
        "Name": "Test Offer name"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create "draft" CPP0
for any product type (AMI/SaaS/Container) that can be reviewed internally
before publishing to buyers
```

```

CAPI-60
"""
import os

import utils.start_changeset as sc # noqa: E402
import utils.stringify_details as sd # noqa: E402

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Create a draft CPP0 offer for a product")

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 재판매 권한 부여 대체 비공개 제안 생성

다음 코드 예제에서는 계약 요금이 적용되는 기존 계약에서 재판매 권한 부여 대체 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateReplacementOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateReplacementOfferResaleAuth",
      "DetailsDocument": {
        "AgreementId": "agmt-11111111111111111111111111111111",
        "ResaleAuthorizationId": "resaleauthz-1111111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test replacement offer for SaaSProduct using AWS
Marketplace API Reference Codes",
        "Description": "Test private resale replacement offer with
contract pricing for SaaSProduct"
      }
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
      },
      "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
          {
            "Type": "FixedUpfrontPricingTerm",
```

```

        "CurrencyCode": "USD",
        "Price": "0.0",
        "Duration": "P12M",
        "Grants": [
            {
                "DimensionKey": "BasicService",
                "MaxQuantity": 2
            }
        ]
    },
    ],
},
{
    "ChangeType": "UpdateValidityTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "ValidityTerm",
                "AgreementEndDate": "2024-01-30"
            }
        ]
    }
},
{
    "ChangeType": "UpdatePaymentScheduleTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "PaymentScheduleTerm",
                "CurrencyCode": "USD",
                "Schedule": [
                    {
                        "ChargeDate": "2024-01-01",

```

```

        "ChargeAmount": "0"
      }
    ]
  }
},
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "StandardEula",
            "Version": "2022-07-14"
          }
        ]
      }
    ]
  }
},
{
  "ChangeType": "UpdateAvailability",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
  },
  "DetailsDocument": {
    "AvailabilityEndDate": "2023-12-31"
  }
},
{
  "ChangeType": "ReleaseOffer",
  "Entity": {
    "Type": "Offer@1.0",

```

```

        "Identifier":
        "$CreateReplacementOfferResaleAuth.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType" : "CreateReplacementOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateReplacementOfferResaleAuth",
      "DetailsDocument": {
        "AgreementId": "agmt-111111111111111111111111",
        "ResaleAuthorizationId": "resaleauthz-1111111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
        "$CreateReplacementOfferResaleAuth.Entity.Identifier"
      },
      "DetailsDocument": {

```

```

        "Name": "Test replacement offer for SaaSProduct using AWS
Marketplace API Reference Codes",
        "Description": "Test private resale replacement offer with
contract pricing for SaaSProduct"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "FixedUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "Price": "0.0",
                "Duration": "P12M",
                "Grants": [
                    {
                        "DimensionKey": "BasicService",
                        "MaxQuantity": 2
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateValidityTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "ValidityTerm",
                "AgreementEndDate": "2024-01-30"
            }
        ]
    }
}

```

```

        ]
      }
    },
    {
      "ChangeType": "UpdatePaymentScheduleTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "PaymentScheduleTerm",
            "CurrencyCode": "USD",
            "Schedule": [
              {
                "ChargeDate": "2024-01-01",
                "ChargeAmount": "0"
              }
            ]
          }
        ]
      }
    },
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "StandardEula",
                "Version": "2022-07-14"
              }
            ]
          }
        ]
      }
    }
  ]
}

```



```

        }
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-12-31"
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier":
"$CreateReplacementOfferResaleAuth.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a resale authorization
replacement private offer
from an existing agreement with contract pricing
CAPI-96
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

```

```
def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create resale authorization replacement private offer with contract pricing",
    )

    return response

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 채널 파트너가 생성한 모든 CPPOs 나열

다음 코드 예제에서는 채널 파트너가 생성한 모든 CPPO를 나열하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import java.util.ArrayList;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.document.Document;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;
import software.amazon.awssdk.services.marketplacecatalog.model.EntitySummary;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesResponse;

public class ListAllCppoOffers {

    /*
     * List all CPPOs created by a channel partner
     */
    public static void main(String[] args) {

        List<String> cppoOfferIds = getAllCppoOfferIds();

        ReferenceCodesUtils.formatOutput(cppoOfferIds);
    }

    public static List<String> getAllCppoOfferIds() {
        MarketplaceCatalogClient marketplaceCatalogClient =
            MarketplaceCatalogClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        // get all offer entity ids
    }
```

```
List<String> entityIdList = new ArrayList<String>();

ListEntitiesRequest listEntitiesRequest =
    ListEntitiesRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .entityType(ENTITY_TYPE_OFFER)
        .maxResults(10)
        .nextToken(null)
        .build();

ListEntitiesResponse listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);

for (EntitySummary entitySummary : listEntitiesResponse.entitySummaryList()) {
    entityIdList.add(entitySummary.entityId());
}

while (listEntitiesResponse.nextToken() != null) {
    listEntitiesRequest =
        ListEntitiesRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityType(ENTITY_TYPE_OFFER)
            .maxResults(10)
            .nextToken(listEntitiesResponse.nextToken())
            .build();
    listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);

    for (EntitySummary entitySummary : listEntitiesResponse.entitySummaryList()) {
        entityIdList.add(entitySummary.entityId());
    }
}

// filter for CPP0 offers: ResaleAuthorizationId exists in Details

List<String> cppoOfferIds = new ArrayList<String>();

for (String entityId : entityIdList) {
    DescribeEntityRequest describeEntityRequest =
        DescribeEntityRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityId(entityId)
            .build();
```

```

DescribeEntityResponse describeEntityResponse =
marketplaceCatalogClient.describeEntity(describeEntityRequest);

Document resaleAuthorizationDocument =
describeEntityResponse.detailsDocument().asMap().get(ATTRIBUTE_RESALE_AUTHORIZATION_ID);
String resaleAuthorizationId = resaleAuthorizationDocument != null ?
resaleAuthorizationDocument.asString() : "";

if (!resaleAuthorizationId.isEmpty()) {
    cppoOfferIds.add(resaleAuthorizationId);
}
}
return cppoOfferIds;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ListEntities](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to list all Channel Partner
Offers
in an account

Program executed with no arguments:
ie. python3 list_all_cppo_offers.py

CAPI-93

```

```
"""

import json
import logging

import boto3
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-catalog")

def get_offer_entities():
    """
    Returns a list of all offers in the account
    """

    next_token = "" # nsec: B105
    response_list = []

    try:
        response = mp_client.list_entities(Catalog="AWSMarketplace",
                                           EntityType="Offer")
    except ClientError as e:
        logging.exception(f"Couldn't list entities. {e}")
        raise

    response_list.append(response)

    # Results are paginated depending on number of entities returned
    while "NextToken" in response:
        next_token = response["NextToken"]

        try:
            response = mp_client.list_entities(
                Catalog="AWSMarketplace",
                EntityType="Offer",
                NextToken=next_token,
            )
        except ClientError as e:
            logging.exception(f"Couldn't list entities. {e}")
            raise

        if "NextToken" in response:
            response_list.append(response)
```

```
    return response_list

def build_offer_list(response_list):
    """
    Cleans up list_entities response list with just list of offer IDs
    """
    offer_list = []

    for response in response_list:
        for entity in response["EntitySummaryList"]:
            offer_list.append(entity["EntityId"])

    return offer_list

def check_offer_resaleauth(offer_id):
    """
    Checks to see if an offer is based on a resale authorization
    """
    offer_response = describe_entity(offer_id)
    offer_details = json.loads(offer_response["Details"])
    if offer_details is None:
        offer_details = offer_response["DetailsDocument"]
    if "ResaleAuthorizationId" in offer_details and
    offer_details["ResaleAuthorizationId"] is not None:
        return offer_id
    else:
        return None

def describe_entity(entity_id):
    """
    General purpose describe entity call
    """
    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace",
            EntityId=entity_id,
        )
    except ClientError as e:
        logging.exception(f"Couldn't describe entity. {e}")
        raise
```

```

    return response

def get_resaleauth_offers():
    """
    Returns a list of all offers in the account that are
    based on a resale authorization
    """
    resale_offer_list = []

    response_list = get_offer_entities()
    offer_list = build_offer_list(response_list)
    for offer in offer_list:
        print ("offer id " + offer)
        offer_info = check_offer_resaleauth(offer)
        if offer_info is not None:
            resale_offer_list.append(offer_info)

    return resale_offer_list

if __name__ == "__main__":
    print(get_resaleauth_offers())

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [ListEntities](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 채널 파트너가 사용할 수 있는 모든 공유 재판매 권한 나열

다음 코드 예제에서는 채널 파트너가 사용할 수 있는 모든 공유 재판매 권한을 나열하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import java.util.ArrayList;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesResponse;

public class ListAllSharedResaleAuthorizations {

    /*
     * list all resale authorizations shared to an account
     */
    public static void main(String[] args) {

        List<ListEntitiesResponse> responseList = getListEntityResponseList();
        ReferenceCodesUtils.formatOutput(responseList);
    }

    public static List<ListEntitiesResponse> getListEntityResponseList() {
        MarketplaceCatalogClient marketplaceCatalogClient =
            MarketplaceCatalogClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
```

```
.credentialsProvider(ProfileCredentialsProvider.create())
    .build();

List<ListEntitiesResponse> responseList = new
ArrayList<ListEntitiesResponse>();

ListEntitiesRequest listEntitiesRequest =
    ListEntitiesRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .entityType(ENTITY_TYPE_RESALE_AUTHORIZATION)
        .maxResults(10)
        .ownershipType(OWNERSHIP_TYPE_SHARED)
        .nextToken(null)
        .build();

ListEntitiesResponse listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);

responseList.add(listEntitiesResponse);

while (listEntitiesResponse.nextToken() != null) {
    listEntitiesRequest = ListEntitiesRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .entityType(ENTITY_TYPE_RESALE_AUTHORIZATION)
        .maxResults(10)
        .ownershipType(OWNERSHIP_TYPE_SHARED)
        .nextToken(listEntitiesResponse.nextToken())
        .build();

    listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);

    responseList.add(listEntitiesResponse);
}
return responseList;
}
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [ListEntities](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to list all resale authorizations
shared to an account

Program executed with no arguments:
ie. python3 list_all_resale_authorizations.py

CAPI-94
"""

import logging

import boto3
import utils.helpers as hlp # noqa: E402
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-catalog")

def get_shared_entities():
    next_token = "" # nosec: B105
    response_list = []

    try:
        response = mp_client.list_entities(
            Catalog="AWSMarketplace",
            EntityType="ResaleAuthorization",
            OwnershipType="SHARED",
        )
```

```

except ClientError as e:
    logging.exception(f"Couldn't list entities. {e}")
    raise

response_list.append(response)

# Results can be paginated depending on number of entities returned
while "NextToken" in response:
    next_token = response["NextToken"]

    try:
        response = mp_client.list_entities(
            Catalog="AWSMarketplace",
            EntityType="ResaleAuthorization",
            OwnershipType="SHARED",
            NextToken=next_token,
        )
    except ClientError as e:
        logging.exception(f"Couldn't list entities. {e}")
        raise

    if "NextToken" in response:
        response_list.append(response)

return response_list

if __name__ == "__main__":
    response_list = get_shared_entities()
    hlp.pretty_print_datetime(response_list)

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [ListEntities](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 CPPO 게시 및 구매자 EULA 추가

다음 코드 예제에서는 CPPO를 게시하고 구매자 EULA를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType" : "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateCPP0offer",
      "DetailsDocument": {
        "ResaleAuthorizationId": "resaleauthz-111111111111",
        "Name": "Test Offer",
        "Description": "Test product"
      }
    },
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/sample-bucket/custom-eula.pdf"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

```

        }
    ]
}
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": ["222222222222"]
        }
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-07-31"
    }
},
{
    "ChangeType": "UpdateValidityTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "ValidityTerm",
                "AgreementDuration": "P450D"
            }
        ]
    }
},
{

```

```

        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateCPP0offer.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "CreateOfferUsingResaleAuthorization",
            "Entity": {
                "Type": "Offer@1.0"
            },
            "ChangeName": "CreateCPP0offer",
            "DetailsDocument": {
                "ResaleAuthorizationId": "resaleauthz-111111111111",
                "Name": "Test Offer",
                "Description": "Test product"
            }
        },
        {
            "ChangeType": "UpdateLegalTerms",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "$CreateCPP0offer.Entity.Identifier"
            }
        }
    ]
}

```

```

    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/custom-eula.pdf"
            }
          ]
        }
      ]
    }
  ]
},
{
  "ChangeType": "UpdateTargeting",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0offer.Entity.Identifier"
  },
  "DetailsDocument": {
    "PositiveTargeting": {
      "BuyerAccounts": ["222222222222"]
    }
  }
},
{
  "ChangeType": "UpdateAvailability",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0offer.Entity.Identifier"
  },
  "DetailsDocument": {
    "AvailabilityEndDate": "2023-07-31"
  }
},
{
  "ChangeType": "UpdateValidityTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0offer.Entity.Identifier"
  },
  "DetailsDocument": {

```



```

        "Terms": [
            {
                "Type": "ValidityTerm",
                "AgreementDuration": "P450D"
            }
        ]
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateCPP0offer.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to publish CPP0
for any product type (AMI/SaaS/Container) and append buyer EULA
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

```

```

else:
    stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Publish CPPO for any product type and append buyer EULA",
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

일회성 재판매 권한 부여를 사용하여 CPPO 게시 및 AWSSDK를 사용하여 가격 인상 업데이트

다음 코드 예제에서는 AMI, SaaS 또는 컨테이너 제품에 대한 일회성 재판매 권한 부여를 사용하여 CPPO를 게시하고 가격 인상을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```

{
    "Catalog": "AWSMarketplace",

```

```

"ChangeSet": [
  {
    "ChangeType" : "CreateOfferUsingResaleAuthorization",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "ChangeName": "CreateCPP0offer",
    "DetailsDocument": {
      "ResaleAuthorizationId": "resaleauthz-111111111111",
      "Name": "Test Offer",
      "Description": "Test product"
    }
  },
  {
    "ChangeType": "UpdateMarkup",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
      "Percentage" : "5.0"
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": ["222222222222"]
      }
    }
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-07-31"
    }
  }
]

```

```

    },
    {
      "ChangeType": "UpdateValidityTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "ValidityTerm",
            "AgreementDuration": "P450D"
          }
        ]
      }
    },
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
      },
      "DetailsDocument": {}
    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [

```

```

{
  "ChangeType" : "CreateOfferUsingResaleAuthorization",
  "Entity": {
    "Type": "Offer@1.0"
  },
  "ChangeName": "CreateCPP0",
  "DetailsDocument": {
    "ResaleAuthorizationId": "resaleauthz-111111111111"
  }
},
{
  "ChangeType": "UpdateInformation",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0.Entity.Identifier"
  },
  "DetailsDocument": {
    "Name": "Test Offer name",
    "Description": "Test Offer description"
  }
},
{
  "ChangeType": "UpdateMarkup",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0.Entity.Identifier"
  },
  "DetailsDocument": {
    "Percentage" : "5.0"
  }
},
{
  "ChangeType": "UpdateTargeting",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0.Entity.Identifier"
  },
  "DetailsDocument": {
    "PositiveTargeting": {
      "BuyerAccounts": ["111111111111"]
    }
  }
},
{

```

```

        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateCPP0.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-12-31"
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateCPP0.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to Create CPP0 using one-time
resale
authorization on AMI, SaaS or Container products and update price markup
CAPI-63
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"

```

```

        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create CPP0 using one-time resale authorization and update price markup"
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 CPPO 초안 게시 및 가격 인상 업데이트

다음 코드 예제에서는 초안 CPPO를 게시하고 가격 인상을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```

{
    "Catalog": "AWSMarketplace",

```

```

"ChangeSet": [
  {
    "ChangeType" : "CreateOfferUsingResaleAuthorization",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "ChangeName": "CreateCPP0offer",
    "DetailsDocument": {
      "ResaleAuthorizationId": "resaleauthz-111111111111",
      "Name": "Test Offer",
      "Description": "Test product"
    }
  },
  {
    "ChangeType": "UpdateMarkup",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
      "Percentage" : "5.0"
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": ["222222222222"]
      }
    }
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateCPP0offer.Entity.Identifier"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-07-31"
    }
  }
]

```



```

    },
    {
      "ChangeType": "UpdateValidityTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "ValidityTerm",
            "AgreementDuration": "P450D"
          }
        ]
      }
    },
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateCPP0offer.Entity.Identifier"
      },
      "DetailsDocument": {}
    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [

```

```

{
  "ChangeType" : "CreateOfferUsingResaleAuthorization",
  "Entity": {
    "Type": "Offer@1.0"
  },
  "ChangeName": "CreateCPP0",
  "DetailsDocument": {
    "ResaleAuthorizationId": "resaleauthz-111111111111"
  }
},
{
  "ChangeType": "UpdateInformation",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0.Entity.Identifier"
  },
  "DetailsDocument": {
    "Name": "Test Offer name",
    "Description": "Test Offer description"
  }
},
{
  "ChangeType": "UpdateMarkup",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0.Entity.Identifier"
  },
  "DetailsDocument": {
    "Percentage" : "5.0"
  }
},
{
  "ChangeType": "UpdateTargeting",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateCPP0.Entity.Identifier"
  },
  "DetailsDocument": {
    "PositiveTargeting": {
      "BuyerAccounts": ["111111111111"]
    }
  }
},
{

```

```

        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateCPP0.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-05-31"
        }
    },
    {
        "ChangeType": "UpdateValidityTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateCPP0.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "ValidityTerm",
                    "AgreementDuration": "P450D"
                }
            ]
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateCPP0.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to publish "draft" CPP0

```

```

for any product type (AMI/SaaS/Container) and update price markup
CAPI-72
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Publish draft CPPO for any product type adn update price markup"
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 CPPO의 만료 날짜 업데이트

다음 코드 예제에서는 구매자에게 제안을 평가하고 수락하는 데 필요한 시간을 더 줄 수 있도록 CPPO의 만료 날짜를 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2025-07-31"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2025-07-31"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update the expiry
date of a CPP0 offer
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set
```

```

    response = sc.usage_demo(
        stringified_change_set,
        "Update the expiry date of a CPP0 offer"
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API용 컨테이너 제품

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 공개 제안 초안을 사용하여 컨테이너 제품 초안 생성](#)
- [AWSSDK를 사용하여 공개 제안, 계약 요금으로 제한된 컨테이너 제품 생성](#)

AWSSDK를 사용하여 공개 제안 초안을 사용하여 컨테이너 제품 초안 생성

다음 코드 예제에서는 공개 제안 초안을 사용하여 컨테이너 제품 초안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "changeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product"
      }
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier",
        "Name": "Test Offer"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.


```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier",
        "Name": "Test Offer"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create an container draft
product
with a draft public offer.
CAPI-03
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd
```

```
def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create a draft container product with a draft public offer",
    )

    return response

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 공개 제안, 계약 요금으로 제한된 컨테이너 제품 생성

다음 코드 예제에서는 공개 제안, 계약 요금, 표준 EULA로 제한된 컨테이너 제품을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {},
      "ChangeName": "CreateProductChange"
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "ContainerProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "Categories": [
          "Streaming solutions"
        ],
        "ProductTitle": "ContainerProduct",
        "AdditionalResources": [],
        "LongDescription": "Long description goes here",
        "SearchKeywords": [
          "container streaming"
        ],
        "ShortDescription": "Description1",
        "Highlights": [
          "Highlight 1",
          "Highlight 2"
        ],
        "SupportDescription": "No support available",
        "VideoUrls": []
      }
    },
    {
      "ChangeType": "AddDimensions",
      "Entity": {
```

```

        "Type": "ContainerProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
        {
            "Key": "Cores",
            "Description": "Cores per cluster",
            "Name": "Cores",
            "Types": [
                "Entitled"
            ],
            "Unit": "Units"
        }
    ]
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "ContainerProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111"
            ]
        }
    }
},
{
    "ChangeType": "AddRepositories",
    "Entity": {
        "Type": "ContainerProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Repositories": [
            {
                "RepositoryName": "uniquerepositoryname",
                "RepositoryType": "ECR"
            }
        ]
    }
},

```

```

{
  "ChangeType": "ReleaseProduct",
  "Entity": {
    "Type": "ContainerProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {}
},
{
  "ChangeType": "CreateOffer",
  "Entity": {
    "Type": "Offer@1.0"
  },
  "DetailsDocument": {
    "ProductId": "$CreateProductChange.Entity.Identifier"
  },
  "ChangeName": "CreateOfferChange"
},
{
  "ChangeType": "UpdatePricingTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "PricingModel": "Contract",
    "Terms": [
      {
        "Type": "ConfigurableUpfrontPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
          {
            "Selector": {
              "Type": "Duration",
              "Value": "P12M"
            },
            "Constraints": {
              "MultipleDimensionSelection": "Disallowed",
              "QuantityConfiguration": "Disallowed"
            },
            "RateCard": [
              {
                "DimensionKey": "Cores",
                "Price": "0.25"
              }
            ]
          }
        ]
      }
    ]
  }
}

```

```

    }
  ]
}
},
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "StandardEula",
            "Version": "2022-07-14"
          }
        ]
      }
    ]
  }
},
{
  "ChangeType": "UpdateSupportTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "SupportTerm",
        "RefundPolicy": "No refunds"
      }
    ]
  }
},
{

```

```

        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Some container offer Name",
            "Description": "Some interesting container offer description"
        }
    },
    {
        "ChangeType": "UpdateRenewalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "RenewalTerm"
                }
            ]
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument": {},
      "ChangeName": "CreateProductChange"
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "ContainerProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "Categories": [
          "Streaming solutions"
        ],
        "ProductTitle": "ContainerProduct",
        "AdditionalResources": [],
        "LongDescription": "Long description goes here",
        "SearchKeywords": [
          "container streaming"
        ],
        "ShortDescription": "Description1",
        "Highlights": [
          "Highlight 1",
          "Highlight 2"
        ]
      }
    }
  ]
}
```



```

        ],
        "SupportDescription": "No support available",
        "VideoUrls": []
    },
    {
        "ChangeType": "AddDimensions",
        "Entity": {
            "Type": "ContainerProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": [
            {
                "Key": "Cores",
                "Description": "Cores per cluster",
                "Name": "Cores",
                "Types": [
                    "Entitled"
                ],
                "Unit": "Units"
            }
        ]
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "ContainerProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111"
                ]
            }
        }
    },
    {
        "ChangeType": "AddRepositories",
        "Entity": {
            "Type": "ContainerProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {

```

```

        "Repositories": [
            {
                "RepositoryName": "uniquerepositoryname",
                "RepositoryType": "ECR"
            }
        ]
    },
    {
        "ChangeType": "ReleaseProduct",
        "Entity": {
            "Type": "ContainerProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "CreateOffer",
        "Entity": {
            "Type": "Offer@1.0"
        },
        "DetailsDocument": {
            "ProductId": "$CreateProductChange.Entity.Identifier"
        },
        "ChangeName": "CreateOfferChange"
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",
                                "Value": "P12M"
                            }
                        }
                    ]
                }
            ]
        }
    }

```

```

        "Constraints": {
            "MultipleDimensionSelection": "Disallowed",
            "QuantityConfiguration": "Disallowed"
        },
        "RateCard": [
            {
                "DimensionKey": "Cores",
                "Price": "0.25"
            }
        ]
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [

```

```

        {
            "Type": "SupportTerm",
            "RefundPolicy": "No refunds"
        }
    ]
},
{
    "ChangeType": "UpdateInformation",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Name": "Some container offer Name",
        "Description": "Some interesting container offer description"
    }
},
{
    "ChangeType": "UpdateRenewalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "RenewalTerm"
            }
        ]
    }
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create limited container
product with public offer, contract pricing and standard EULA
CAPI-15
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create limited container product with public offer contract pricing and
standard EULA",
    )

    return response

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#) [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API의 개체

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 단일 호출의 모든 엔터티 설명](#)
- [AWSSDK를 사용하여 제품과 관련된 모든 제안 나열 및 설명](#)

AWSSDK를 사용하여 단일 호출의 모든 엔터티 설명

다음 코드 예제에서는 단일 직접 호출로 모든 엔터티를 설명하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.BatchDescribeEntitiesRequest;
import software.amazon.awssdk.services.marketplacecatalog.model.EntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.BatchDescribeEntitiesResponse;
```

```
import software.amazon.awssdk.services.marketplacecatalog.model.EntityDetail;
import
    software.amazon.awssdk.services.marketplacecatalog.model.BatchDescribeErrorDetail;

import java.util.Arrays;
import java.util.Map;

public class BatchDescribeEntities {

    /**
     * BatchDescribe my entities in a single call and
     * check if it contains all the information I need to know about the
     entities.
     */
    public static void main(String[] args) {

        MarketplaceCatalogClient marketplaceCatalogClient =
            MarketplaceCatalogClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        BatchDescribeEntitiesRequest batchDescribeEntitiesRequest =
            BatchDescribeEntitiesRequest.builder()
                .entityRequestList(Arrays.asList(
                    EntityRequest.builder()

.catalog(AWS_MP_CATALOG).entityId(OFFER_ID)
                        .build(),
                    EntityRequest.builder()

.catalog(AWS_MP_CATALOG).entityId(PRODUCT_ID)
                        .build()))
                .build();

        BatchDescribeEntitiesResponse batchDescribeEntitiesResponse =
marketplaceCatalogClient.batchDescribeEntities(batchDescribeEntitiesRequest);

        // Reading the successful entities response
        Map<String, EntityDetail> entityDetailsMap =
batchDescribeEntitiesResponse.entityDetails();
        for (Map.Entry<String, EntityDetail> entry : entityDetailsMap.entrySet())
        {
            System.out.println("EntityId: " + entry.getKey());
        }
    }
}
```

```

        ReferenceCodesUtils.formatOutput(entry.getValue());
    }

    // Logging the failed entities error details
    Map<String, BatchDescribeErrorDetail> entityErrorsMap =
batchDescribeEntitiesResponse.errors();
    for (Map.Entry<String, BatchDescribeErrorDetail> entry :
entityErrorsMap.entrySet()) {
        System.out.println(String.format("EntityId: %s, ErrorCode: %s,
ErrorMessage: %s", entry.getKey(),
            entry.getValue().errorCode(),
entry.getValue().errorMessage()));
    }
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [BatchDescribeEntities](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to describe for multiple entities
information in the AWS Marketplace Catalog
CAPI-98
"""

import json
import logging

import boto3
from botocore.exceptions import ClientError

```



```
logger = logging.getLogger(__name__)

PRODUCT_ID = "prod-111111111111"
OFFER_ID = "offer-111111111111"
MARKETPLACE_CATALOG = "AWSMarketplace"

def pretty_print(response):
    json_object = json.dumps(response, indent=4)
    print(json_object)

def get_entities_information(mp_client):
    """
    Returns information about a given product
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of product information
    """

    entity_request_list_param = [
        {'EntityId': PRODUCT_ID, 'Catalog': MARKETPLACE_CATALOG},
        {'EntityId': OFFER_ID, 'Catalog': MARKETPLACE_CATALOG}
    ]
    try:
        response = mp_client.batch_describe_entities(
            EntityRequestList=entity_request_list_param
        )

        return response

    except ClientError as e:
        logger.exception("Unexpected error: %s", e)
        raise

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for entities in the AWS Marketplace Catalog.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-catalog")
```

```

response = get_entities_information(mp_client)
print("Successful entities response -")
pretty_print(response["EntityDetails"])
print("Failed entities response -")
pretty_print(response["Errors"])

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [BatchDescribeEntities](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 제품과 관련된 모든 제안 나열 및 설명

다음 코드 예제에서는 제품과 관련된 모든 제안을 나열하고 설명하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.catalogapi;

import java.util.ArrayList;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;

```

```

import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;
import software.amazon.awssdk.services.marketplacecatalog.model.EntitySummary;
import
    software.amazon.awssdk.services.marketplacecatalog.model.EntityTypeFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesResponse;
import software.amazon.awssdk.services.marketplacecatalog.model.OfferFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferProductIdFilter;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferTargetingFilter;

public class ListProductPrivateOffers {

    private static MarketplaceCatalogClient marketplaceCatalogClient =
        MarketplaceCatalogClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();
    /*
     * retrieve all private offer information related to a single product
     */
    public static void main(String[] args) {

        List<EntitySummary> entitySummaryList = getEntitySummaryList();

        // for each offer id, output the offer detail using DescribeEntity API

        for (EntitySummary entitySummary : entitySummaryList) {
            DescribeEntityRequest describeEntityRequest =
                DescribeEntityRequest.builder()
                    .catalog(AWS_MP_CATALOG)
                    .entityId(entitySummary.entityId())
                    .build();
            DescribeEntityResponse describeEntityResponse =
                marketplaceCatalogClient.describeEntity(describeEntityRequest);
            ReferenceCodesUtils.formatOutput(describeEntityResponse);
        }
    }
}

```

```

    }
}
public static List<EntitySummary> getEntitySummaryList() {
    // define list entities filters

    EntityTypeFilters entityTypeFilters =
        EntityTypeFilters.builder()
            .offerFilters(OfferFilters.builder()
                .targeting(OfferTargetingFilter.builder()
                    .valueListWithStrings(OFFER_TARGETING_BUYERACCOUNTS)
                    .build())
                .productId(OfferProductIdFilter.builder()
                    .valueList(PRODUCT_ID)
                    .build())
                .build())
            .build();

    ListEntitiesRequest listEntitiesRequest =
        ListEntitiesRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityType(ENTITY_TYPE_OFFER).maxResults(50)
            .entityTypeFilters(entityTypeFilters)
            .nextToken(null)
            .build();

    ListEntitiesResponse listEntitiesResponse =
        marketplaceCatalogClient.listEntities(listEntitiesRequest);

    // save all entitySummary of the results into entitySummaryList

    List<EntitySummary> entitySummaryList = new ArrayList<EntitySummary>();

    entitySummaryList.addAll(listEntitiesResponse.entitySummaryList());

    while ( listEntitiesResponse.nextToken() != null &&
listEntitiesResponse.nextToken().length() > 0) {
        listEntitiesRequest =
            ListEntitiesRequest.builder()
                .catalog(AWS_MP_CATALOG)
                .entityType(ENTITY_TYPE_OFFER).maxResults(50)
                .entityTypeFilters(entityTypeFilters)
                .nextToken(listEntitiesResponse.nextToken())
                .build();
    }
}

```

```

    listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);
    entitySummaryList.addAll(listEntitiesResponse.entitySummaryList());
}
return entitySummaryList;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 다음 주제를 참조하세요.
 - [DescribeEntity](#)
 - [ListEntities](#)

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to retrieve all offer information
related to a single product
CAPI-97
"""

import argparse
import logging

import boto3
from botocore.exceptions import ClientError
from utils import helpers

logger = logging.getLogger(__name__)

mp_client = boto3.client("marketplace-catalog")

```

```

def get_entity_information(entity_id):
    """
    Returns information about a given entity
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of entity information
    """

    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace",
            EntityId=entity_id,
        )

        return response

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Entity with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def list_entity_details(entity_type, entity_id):
    """
    Returns details about a given entity and entity type
    """

    entity_summary_list = []

    # filter will return details for given entity_id with BuyerAccounts targeting
    filter_list_param = {
        'OfferFilters':{
            'ProductId':{
                'ValueList':[entity_id]
            },
            'Targeting': {
                'ValueList': ["BuyerAccounts"]
            }
        }
    }

    try:

```

```

        response = mp_client.list_entities(
            Catalog="AWSMarketplace",
            EntityType=entity_type,
            EntityTypeFilters = filter_list_param,
            MaxResults=10
        )

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Entity ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

    # add results to entity_summary_list
    entity_summary_list.extend(response["EntitySummaryList"])

    # if there are more than 10 offers, paginate through the results
    while "NextToken" in response and response["NextToken"] is not None:
        try:
            response = mp_client.list_entities(
                Catalog="AWSMarketplace",
                EntityType=entity_type,
                EntityTypeFilters = filter_list_param,
                NextToken=response["NextToken"],
                MaxResults=10
            )

        except ClientError as e:
            if e.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error("Entity ID %s not found.", entity_id)
            else:
                logger.error("Unexpected error: %s", e)

        # add results to entity_summary_list
        entity_summary_list.extend(response["EntitySummaryList"])

    return entity_summary_list

if __name__ == "__main__":
    parser = argparse.ArgumentParser()

    parser.add_argument(
        "--entity-id",
        "-eid",

```

```

        help="Provide Entity ID corresponding to a product to filter offers on",
        required=True,
    )

    args = parser.parse_args()

    # Gets a offers associated with the entity_id
    response = list_entity_details(
        "Offer",
        entity_id=args.entity_id
    )

    if response: # if response is not empty

        # list_entity_details returns a list of offers
        for offer in response:

            print("-"*128)
            print(f"Terms for Offer ID: {offer['EntityId']}")
            print("-"*128)

            #retrieve offer information for each offer
            entity_information = get_entity_information(offer["EntityId"])

            helpers.pretty_print_datetime(entity_information)

    else:
        print(f"No information found for Entity ID: {args.entity_id}")

```

- API 세부 정보는 AWSSDK for Python (Boto3) API 참조의 다음 주제를 참조하세요.
 - [DescribeEntity](#)
 - [ListEntities](#)

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API에 대한 제안

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 SaaS 제품의 사용자 지정 차원 생성 및 비공개 제안 생성](#)
- [AWSSDK를 사용하여 AMI 또는 SaaS 제품에 대한 비공개 제안 초안 생성](#)
- [AWSSDK를 사용하여 SaaS 제품에 대한 계약 및 Pay-As-You-Go 요금으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 SaaS 제품에 대한 계약 요금 및 유연한 결제 일정으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 컨테이너 제품의 계약 요금으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 AMI 제품에 대한 계약 요금으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 AMI 제품에 대한 시간당 연간 요금 및 유연한 결제 일정으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 AMI 제품의 시간당 연간 요금으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 AMI 제품의 시간당 요금으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 SaaS 제품에 대한 계층형 계약 요금으로 비공개 제안 생성](#)
- [AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 공개 무료 평가판 제안 생성](#)
- [AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 대체 비공개 제안 생성](#)
- [AWSSDK를 사용하여 공개 제안 설명](#)
- [AWSSDK를 사용하여 비공개 제안 만료](#)
- [AWSSDK를 사용하여 모든 비공개 제안 나열](#)
- [AWSSDK를 사용하여 특정 제품 ID에 대해 릴리스된 공개 및 비공개 제안 나열](#)
- [AWSSDK를 사용하여 Pay-As-You-Go 요금 계약을 적용하도록 제안 업데이트](#)
- [AWSSDK를 사용하여 시간당 연간 요금을 적용하도록 제안 업데이트](#)
- [AWSSDK를 사용하여 특정 지리적 리전에 대상 지정을 적용하도록 제안 업데이트](#)
- [AWSSDK를 사용하여 공개 제안의 이름 및 설명 업데이트](#)
- [AWSSDK를 사용하여 제안의 EULA 업데이트](#)
- [AWSSDK를 사용하여 비공개 제안의 만료 날짜를 미래 날짜로 업데이트](#)
- [AWSSDK를 사용하여 SaaS 제품에 대한 공개 무료 평가판 제안의 무료 평가판 기간 업데이트](#)
- [AWSSDK를 사용하여 제안의 환불 정책 업데이트](#)

AWSSDK를 사용하여 SaaS 제품의 사용자 지정 차원 생성 및 비공개 제안 생성

다음 코드 예제에서는 SaaS 제품에 대한 사용자 지정 차원을 생성하고 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddDimensions",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": [
        {
          "Types": [
            "Entitled"
          ],
          "Description": "Custom Pricing 4 w/ terms and coverage to be defined in Private Offer",
          "Unit": "Units",
          "Key": "Custom4",
          "Name": "Custom Pricing 4"
        }
      ]
    },
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    }
  ]
}
```

```

    },
    "ChangeName": "CreateOfferChange"
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Name": "Private Test Offer - SaaS Contract Product",
      "Description": "Private Test Offer - SaaS Contract Product"
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111"
        ]
      }
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "StandardEula",
              "Version": "2022-07-14"
            }
          ]
        }
      ]
    }
  }
]

```

```

        }
    ]
}
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Constraints": {
                            "MultipleDimensionSelection": "Allowed",
                            "QuantityConfiguration": "Allowed"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "Custom4",
                                "Price": "300.0"
                            }
                        ],
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P36M"
                        }
                    }
                ]
            }
        ]
    }
}
]

```

```

        }
    ]
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
],
"ChangeSetName": "PrivateOfferWithCustomDimension"
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "AddDimensions",
            "Entity": {
                "Type": "SaaSProduct@1.0",
                "Identifier": "prod-11111111111111"
            },
            "DetailsDocument": [
                {
                    "Types": [
                        "Entitled"
                    ],

```

```

        "Description": "Custom Pricing 4 w/ terms and coverage to be
defined in Private Offer",
        "Unit": "Units",
        "Key": "Custom4",
        "Name": "Custom Pricing 4"
    }
]
},
{
    "ChangeType": "CreateOffer",
    "Entity": {
        "Type": "Offer@1.0"
    },
    "DetailsDocument": {
        "ProductId": "prod-11111111111111"
    },
    "ChangeName": "CreateOfferChange"
},
{
    "ChangeType": "UpdateInformation",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Name": "Private Test Offer - SaaS Contract Product",
        "Description": "Private Test Offer - SaaS Contract Product"
    }
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "11111111111111"
            ]
        }
    }
},
{

```

```

    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "StandardEula",
              "Version": "2022-07-14"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-12-31"
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Constraints": {

```

```

        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
    },
    "RateCard": [
        {
            "DimensionKey": "Custom4",
            "Price": "300.0"
        }
    ],
    "Selector": {
        "Type": "Duration",
        "Value": "P36M"
    }
}
]
}
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
],
"ChangeSetName": "PrivateOfferWithCustomDimension"
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a SaaS product custom
dimension and private offer
CAPI-91
"""

```



```
import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set, "Create a SaaS product custom dimension and private offer"
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 또는 SaaS 제품에 대한 비공개 제안 초안 생성

다음 코드 예제에서는 구매자를 대상으로 게시하기 전에 내부적으로 검토할 수 있도록 AMI 또는 SaaS 제품에 대한 비공개 제안 초안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangeSets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "Test Private Offer"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
```

```

        "Name": "Test Private Offer"
    }
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create "draft" Private Offer
for any AMI or SAAS product type that can be reviewed internally
before publishing to buyers
CAPI-30
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Private offer for AMI product")

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 제품에 대한 계약 및 Pay-As-You-Go 요금으로 비공개 제안 생성

다음 코드 예제에서는 SaaS 제품에 대한 계약 및 종량제 요금이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for SaaSProduct using AWS Marketplace API Reference Code",

```

```

        "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [
                                {
                                    "DimensionKey": "WorkloadSmall",
                                    "Price": "0.15"
                                },
                                {
                                    "DimensionKey": "WorkloadMedium",
                                    "Price": "0.25"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    }
}

```

```

        ]
      },
      {
        "Type": "ConfigurableUpfrontPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
          {
            "Selector": {
              "Type": "Duration",
              "Value": "P12M"
            },
            "RateCard": [
              {
                "DimensionKey": "BasicService",
                "Price": "150"
              },
              {
                "DimensionKey": "PremiumService",
                "Price": "300"
              }
            ],
            "Constraints": {
              "MultipleDimensionSelection": "Allowed",
              "QuantityConfiguration": "Allowed"
            }
          }
        ]
      }
    ],
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",

```

```

        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
    }
}
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
```

```

"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType": "CreateOffer",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "ChangeName": "CreateOfferChange",
    "DetailsDocument": {
      "ProductId": "prod-111111111111"
    }
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Name": "Test private offer for SaaSProduct using AWS Marketplace
API Reference Code",
      "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111",
          "222222222222"
        ]
      }
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",

```



```

        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "UsageBasedPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "RateCard": [
                            {
                                "DimensionKey": "WorkloadSmall",
                                "Price": "0.15"
                            },
                            {
                                "DimensionKey": "WorkloadMedium",
                                "Price": "0.25"
                            }
                        ]
                    }
                ]
            },
            {
                "Type": "ConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "BasicService",
                                "Price": "150"
                            },
                            {
                                "DimensionKey": "PremiumService",
                                "Price": "300"
                            }
                        ],
                        "Constraints": {
                            "MultipleDimensionSelection": "Allowed",

```

```

        "QuantityConfiguration": "Allowed"
    }
}
]
}
]
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "CustomEula",
                        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    }
}
]
}

```

```

        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer with
contract with PAYG pricing for my SaaS product
CAPI-34
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set,
        "Create private offer with contract with PAYG pricing for my SaaS
product",
    )

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 제품에 대한 계약 요금 및 유연한 결제 일정으로 비공개 제안 생성

다음 코드 예제에서는 SaaS 제품에 대한 계약 요금 및 유연한 결제 일정이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
```

```

        "Name": "Test private offer for SaaSProduct using AWS Marketplace
API Reference Code",
        "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111",
                "222222222222"
            ]
        }
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "FixedUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "Price": "0.0",
                "Grants": [
                    {
                        "DimensionKey": "BasicService",
                        "MaxQuantity": 1
                    },
                    {
                        "DimensionKey": "PremiumService",
                        "MaxQuantity": 1
                    }
                ]
            }
        ]
    }
}

```

```

        }
    ]
}
},
{
    "ChangeType": "UpdateValidityTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "ValidityTerm",
                "AgreementDuration": "P12M"
            }
        ]
    }
},
{
    "ChangeType": "UpdatePaymentScheduleTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "PaymentScheduleTerm",
                "CurrencyCode": "USD",
                "Schedule": [
                    {
                        "ChargeDate": "2024-01-01",
                        "ChargeAmount": "200.00"
                    },
                    {
                        "ChargeDate": "2024-02-01",
                        "ChargeAmount": "170.00"
                    }
                ]
            }
        ]
    }
},

```

```

    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
              }
            ]
          }
        ]
      }
    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
      }
    },
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {}
    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for SaaSProduct using AWS Marketplace
API Reference Code",
        "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
      }
    },
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      }
    }
  ]
}
```



```

        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111"
                ]
            }
        },
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "FixedUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "Price": "0.0",
                    "Grants": [
                        {
                            "DimensionKey": "BasicService",
                            "MaxQuantity": 1
                        }
                    ]
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateValidityTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "ValidityTerm",
                    "AgreementDuration": "P12M"
                }
            ]
        }
    }
]

```

```

    }
  },
  {
    "ChangeType": "UpdatePaymentScheduleTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "PaymentScheduleTerm",
          "CurrencyCode": "USD",
          "Schedule": [
            {
              "ChargeDate": "2024-01-01",
              "ChargeAmount": "200.00"
            },
            {
              "ChargeDate": "2024-02-01",
              "ChargeAmount": "170.00"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/custom-eula.pdf"
            }
          ]
        }
      ]
    }
  }
]

```

```

        }
    ]
}
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "SupportTerm",
                "RefundPolicy": "Some kind of refund policy description"
            }
        ]
    }
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer
with contract pricing and flexible payment schedule for my SaaS product
CAPI-39
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create private offer with contract pricing and flexible payment schedule
for my SaaS product",
    )

    return response

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 컨테이너 제품의 계약 요금으로 비공개 제안 생성

다음 코드 예제에서는 컨테이너 제품에 대한 계약 요금이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
        "Name": "Test private offer for Container product using AWS
Marketplace API Reference Code",
        "Description": "Test private offer for Container product with
contract pricing using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111"
            ]
        }
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "Constraints": {
                            "MultipleDimensionSelection": "Disallowed",
                            "QuantityConfiguration": "Disallowed"
                        }
                    }
                ]
            }
        ]
    }
}

```

```

        "RateCard": [
            {
                "DimensionKey": "ReqPerHour",
                "Price": "0.25"
            }
        ]
    }
}
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "ReleaseOffer",

```

```

        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "CreateOffer",
            "Entity": {
                "Type": "Offer@1.0"
            },
            "ChangeName": "CreateOfferChange",
            "DetailsDocument": {
                "ProductId": "prod-111111111111"
            }
        },
        {
            "ChangeType": "UpdateInformation",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "$CreateOfferChange.Entity.Identifier"
            },
            "DetailsDocument": {

```



```

        "Name": "Test private offer for Container product using AWS
Marketplace API Reference Code",
        "Description": "Test private offer for Container product with
contract pricing using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111"
            ]
        }
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "Constraints": {
                            "MultipleDimensionSelection": "Disallowed",
                            "QuantityConfiguration": "Disallowed"
                        },
                        "RateCard": [

```

```

        "DimensionKey": "ReqPerHour",
        "Price": "0.25"
      }
    ]
  }
},
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "StandardEula",
            "Version": "2022-07-14"
          }
        ]
      }
    ]
  }
},
{
  "ChangeType": "UpdateAvailability",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "AvailabilityEndDate": "2023-12-31"
  }
},
{
  "ChangeType": "ReleaseOffer",
  "Entity": {
    "Type": "Offer@1.0",

```

```

        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer (target
buyers)
for my Container product with contract pricing
CAPI-36
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "publish a private offer for my Container product with contract pricing",
    )

    return response

```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 제품에 대한 계약 요금으로 비공개 제안 생성

다음 코드 예제에서는 AMI 제품에 대한 계약 요금이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    }
  ],
}
```

```

    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for AmiProduct using AWS Marketplace
API Reference Code",
        "Description": "Test private offer with hourly annual pricing for
AmiProduct using AWS Marketplace API Reference Code"
      }
    },
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "BuyerAccounts": [
            "111111111111",
            "222222222222"
          ]
        }
      }
    },
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
              }
            ]
          }
        ]
      }
    }
  ]
}

```

```

        ]
      }
    ]
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-12-31"
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "ReadOnlyUsers",
                  "Price": "220.00"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  }
}

```

```

        ]
      }
    ]
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {

```

```

        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test private offer for AmiProduct using AWS Marketplace
API Reference Code",
            "Description": "Test private offer with hourly annual pricing for
AmiProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "LegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",
                            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                        }
                    ]
                }
            ]
        }
    }
]

```



```

        }
      ]
    }
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-12-31"
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "ReadOnlyUsers",
                  "Price": "220.00"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  }
]

```

```

        }
    ]
}
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer with
contract pricing for my AMI product
CAPI-35
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set, "create private offer with contract pricing for my AMI
product"

```

```
)

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 제품에 대한 시간당 연간 요금 및 유연한 결제 일정으로 비공개 제안 생성

다음 코드 예제에서는 AMI 제품에 대한 시간 단위 연간 요금과 유연한 결제 일정이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
```

```

        "ProductId": "prod-111111111111"
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test private offer for AmiProduct using AWS Marketplace
API Reference Code",
            "Description": "Test private offer with hourly annual pricing for
AmiProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "LegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",

```

```

        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
    }
}
}
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
            {
                "Type": "UsageBasedPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "RateCard": [
                            {
                                "DimensionKey": "t2.micro",
                                "Price": "0.17"
                            }
                        ]
                    }
                ]
            }
        ],
        {
            "Type": "FixedUpfrontPricingTerm",
            "CurrencyCode": "USD",

```

```

        "Price": "0.0",
        "Duration": "P365D",
        "Grants": [
            {
                "DimensionKey": "t2.micro",
                "MaxQuantity": 1
            }
        ]
    },
    {
        "ChangeType": "UpdateValidityTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "ValidityTerm",
                    "AgreementDuration": "P650D"
                }
            ]
        }
    },
    {
        "ChangeType": "UpdatePaymentScheduleTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "PaymentScheduleTerm",
                    "CurrencyCode": "USD",
                    "Schedule": [
                        {
                            "ChargeDate": "2024-01-01",
                            "ChargeAmount": "200.00"
                        }
                    ]
                }
            ]
        }
    }
]

```

```

        "ChargeDate": "2024-02-01",
        "ChargeAmount": "170.00"
      }
    ]
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    }
  ]
}

```

```

    }
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Name": "Test private offer for AmiProduct using AWS Marketplace
API Reference Code",
      "Description": "Test private offer with hourly annual pricing for
AmiProduct using AWS Marketplace API Reference Code"
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111",
          "222222222222"
        ]
      }
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",

```



```

        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
    }
  ]
}
},
{
  "ChangeType": "UpdateAvailability",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "AvailabilityEndDate": "2023-12-31"
  }
},
{
  "ChangeType": "UpdatePricingTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "PricingModel": "Usage",
    "Terms": [
      {
        "Type": "UsageBasedPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
          {
            "RateCard": [
              {
                "DimensionKey": "t2.micro",
                "Price": "0.17"
              }
            ]
          }
        ]
      }
    ],
    "Type": "FixedUpfrontPricingTerm",
    "CurrencyCode": "USD",

```

```

        "Price": "0.0",
        "Duration": "P365D",
        "Grants": [
            {
                "DimensionKey": "t2.micro",
                "MaxQuantity": 1
            }
        ]
    },
    {
        "ChangeType": "UpdateValidityTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "ValidityTerm",
                    "AgreementDuration": "P650D"
                }
            ]
        }
    },
    {
        "ChangeType": "UpdatePaymentScheduleTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "PaymentScheduleTerm",
                    "CurrencyCode": "USD",
                    "Schedule": [
                        {
                            "ChargeDate": "2024-01-01",
                            "ChargeAmount": "200.00"
                        }
                    ]
                }
            ]
        }
    }
]

```

```

        "ChargeDate": "2024-02-01",
        "ChargeAmount": "170.00"
    }
    ]
}
    ]
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer with
hourly annual pricing and flexible payment schedule for my AMI product
CAPI-XX
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():

```

```

    sc.usage_demo(
        change_set,
        "Create private offer with hourly annual pricing and flexible payment
        schedule for my AMI product",
    )

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 제품의 시간당 연간 요금으로 비공개 제안 생성

다음 코드 예제에서는 AMI 제품에 대한 시간 단위 연간 요금이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {

```

```

        "Type": "Offer@1.0"
    },
    "DetailsDocument": {
        "ProductId": "prod-111111111111"
    }
},
{
    "ChangeType": "UpdateInformation",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Name": "Test private offer for AmiProduct using AWS Marketplace
API Reference Code",
        "Description": "Test private offer with hourly annual pricing for
AmiProduct using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdateTargeting",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111",
                "222222222222"
            ]
        }
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",

```

```

        "Documents": [
            {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
        ]
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-12-31"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Usage",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.micro",
                                    "Price": "0.17"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    },

```

```

        {
            "Type": "ConfigurableUpfrontPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
                {
                    "Selector": {
                        "Type": "Duration",
                        "Value": "P365D"
                    },
                    "RateCard": [
                        {
                            "DimensionKey": "t2.micro",
                            "Price": "220.00"
                        }
                    ],
                    "Constraints": {
                        "MultipleDimensionSelection": "Allowed",
                        "QuantityConfiguration": "Allowed"
                    }
                }
            ]
        }
    ],
    {
        "ChangeType": "UpdateValidityTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "ValidityTerm",
                    "AgreementDuration": "P650D"
                }
            ]
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",

```

```

        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for AmiProduct using AWS Marketplace
API Reference Code",

```



```

        "Description": "Test private offer with hourly annual pricing for
        AmiProduct using AWS Marketplace API Reference Code"
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "LegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",
                            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                        }
                    ]
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",

```

```

        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
            {
                "Type": "UsageBasedPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "RateCard": [
                            {
                                "DimensionKey": "t2.micro",
                                "Price": "0.17"
                            }
                        ]
                    }
                ]
            },
            {
                "Type": "ConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P365D"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.micro",
                                "Price": "220.00"
                            }
                        ]
                    }
                ]
            }
        ]
    }
}

```

```

        "Constraints": {
            "MultipleDimensionSelection": "Allowed",
            "QuantityConfiguration": "Allowed"
        }
    }
]
}
},
{
    "ChangeType": "UpdateValidityTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "ValidityTerm",
                "AgreementDuration": "P650D"
            }
        ]
    }
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose

```

```
Shows how to use the AWS SDK for Python (Boto3) to create a private offer with
hourly annual pricing for my AMI product
CAPI-31
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set, "create private offer with hourly annual pricing for my AMI
product"
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 제품의 시간당 요금으로 비공개 제안 생성

다음 코드 예제에서는 AMI 제품에 대한 시간 단위 요금이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for AmiProduct using AWS Marketplace API Reference Code",
        "Description": "Test private offer with hourly pricing for AmiProduct using AWS Marketplace API Reference Code"
      }
    },
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
```

```

        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PositiveTargeting": {
            "BuyerAccounts": [
                "111111111111",
                "222222222222"
            ]
        }
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2025-01-01"
    }
},
{
    "ChangeType": "UpdatePricingTerms",

```

```

    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Usage",
      "Terms": [
        {
          "Type": "UsageBasedPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "RateCard": [
                {
                  "DimensionKey": "t2.micro",
                  "Price": "0.15"
                }
              ]
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateValidityTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "ValidityTerm",
          "AgreementDuration": "P30D"
        }
      ]
    }
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    }
  }
}

```

```

    },
    "DetailsDocument": {}
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for AmiProduct using AWS Marketplace  
API Reference Code",
        "Description": "Test private offer with hourly pricing for  
AmiProduct using AWS Marketplace API Reference Code"
      }
    }
  ]
}

```



```

    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111",
          "222222222222"
        ]
      }
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "StandardEula",
              "Version": "2022-07-14"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {

```

```

        "AvailabilityEndDate": "2025-01-01"
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Usage",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.micro",
                                    "Price": "0.15"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateValidityTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "ValidityTerm",
                    "AgreementDuration": "P30D"
                }
            ]
        }
    },

```

```

        {
            "ChangeType": "ReleaseOffer",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "$CreateOfferChange.Entity.Identifier"
            },
            "DetailsDocument": {}
        }
    ]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer with
hourly pricing for my AMI product
CAPI-32
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set, "create private offer with hourly pricing for my AMI product"
    )

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 비공개 제안 생성

다음 코드 예제에서는 SaaS 제품에 대한 구독 요금이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
      "Name": "Test private offer for SaaSProduct using AWS Marketplace
API Reference Code",
      "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111",
          "222222222222"
        ]
      }
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Usage",
      "Terms": [
        {
          "Type": "UsageBasedPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "RateCard": [
                {
                  "DimensionKey": "WorkloadSmall",
                  "Price": "0.13"
                },
                {
                  "DimensionKey": "WorkloadMedium",

```

```

        "Price": "0.22"
      }
    ]
  }
},
{
  "ChangeType": "UpdateValidityTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "ValidityTerm",
        "AgreementDuration": "P30D"
      }
    ]
  }
},
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "LegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
          }
        ]
      }
    ]
  }
}

```

```

    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
      }
    },
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {}
    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },

```

```

        "ChangeName": "CreateOfferChange",
        "DetailsDocument": {
            "ProductId": "prod-111111111111"
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test private offer for SaaSProduct using AWS Marketplace
API Reference Code",
            "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Usage",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",

```



```

        "CurrencyCode": "USD",
        "RateCards": [
            {
                "RateCard": [
                    {
                        "DimensionKey": "WorkloadSmall",
                        "Price": "0.13"
                    },
                    {
                        "DimensionKey": "WorkloadMedium",
                        "Price": "0.22"
                    }
                ]
            }
        ]
    },
    {
        "ChangeType": "UpdateValidityTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "ValidityTerm",
                    "AgreementDuration": "P30D"
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "LegalTerm",

```

```

        "Documents": [
            {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
        ]
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-12-31"
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer with
subscription pricing for my SaaS product
CAPI-33
"""

```

```
import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set, "Create private offer with subscription pricing for my SaaS
product"
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 제품에 대한 계층형 계약 요금으로 비공개 제안 생성

다음 코드 예제에서는 SaaS 제품에 대한 계층형 계약 요금이 적용되는 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for SaaSProduct using AWS Marketplace
API Reference Code",
        "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
      }
    },
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "BuyerAccounts": [
            "111111111111",
            "222222222222"
          ]
        }
      }
    }
  ]
}
```

```

    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
          {
            "Type": "ConfigurableUpfrontPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
              {
                "Selector": {
                  "Type": "Duration",
                  "Value": "P12M"
                },
                "RateCard": [
                  {
                    "DimensionKey": "BasicService",
                    "Price": "120.00"
                  },
                  {
                    "DimensionKey": "PremiumService",
                    "Price": "200.00"
                  }
                ],
                "Constraints": {
                  "MultipleDimensionSelection": "Disallowed",
                  "QuantityConfiguration": "Disallowed"
                }
              }
            ]
          }
        ]
      }
    },
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      }
    }
  ]
}

```

```

    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-12-31"
    }
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test private offer for SaaSProduct using AWS Marketplace
API Reference Code",
        "Description": "Test private offer with subscription pricing for
SaaSProduct using AWS Marketplace API Reference Code"
      }
    },
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      }
    }
  ]
}
```

```

        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        },
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",
                                "Value": "P12M"
                            },
                            "RateCard": [
                                {
                                    "DimensionKey": "BasicService",
                                    "Price": "120.00"
                                },
                                {
                                    "DimensionKey": "PremiumService",
                                    "Price": "200.00"
                                }
                            ]
                        },
                        {
                            "Constraints": {
                                "MultipleDimensionSelection": "Disallowed",
                                "QuantityConfiguration": "Disallowed"
                            }
                        }
                    ]
                }
            ]
        }
    }
}

```



```

        ]
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "LegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",
                            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                        }
                    ]
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-12-31"
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]

```

```
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a private offer with
tiered contract pricing for my SaaS product
CAPI-XX
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set,
        "Create private offer with tiered contract pricing for my SaaS product",
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 제품의 구독 요금으로 공개 무료 평가판 제안 생성

다음 코드 예제에서는 SaaS 제품에 대한 구독 요금이 적용되는 공개 무료 평가판 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test public free trial offer for SaaSProduct using AWS Marketplace API Reference Code",
        "Description": "Test public free trial offer with subscription pricing for SaaSProduct using AWS Marketplace API Reference Code"
      }
    }
  ]
}
```

```

    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Free",
      "Terms": [
        {
          "Type": "FreeTrialPricingTerm",
          "Duration": "P20D",
          "Grants": [
            {
              "DimensionKey": "WorkloadSmall"
            },
            {
              "DimensionKey": "WorkloadMedium"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "StandardEula",
              "Version": "2022-07-14"
            }
          ]
        }
      ]
    }
  ]
}

```

```

    }
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateOfferChange",
      "DetailsDocument": {
        "ProductId": "prod-111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {

```

```

        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Name": "Test public free trial offer for SaaSProduct using AWS
Marketplace API Reference Code",
        "Description": "Test public free trial offer with subscription
pricing for SaaSProduct using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Free",
        "Terms": [
            {
                "Type": "FreeTrialPricingTerm",
                "Duration": "P20D",
                "Grants": [
                    {
                        "DimensionKey": "WorkloadSmall"
                    },
                    {
                        "DimensionKey": "WorkloadMedium"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",

```

```

        "Documents": [
            {
                "Type": "StandardEula",
                "Version": "2022-07-14"
            }
        ]
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a public free trial
offer with subscription pricing for SaaS product
CAPI-13
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

```

```
def main():
    sc.usage_demo(
        change_set,
        "Create public free trial offer with subscription pricing for SaaS
product",
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 요금이 포함된 기존 계약에서 대체 비공개 제안 생성

다음 코드 예제에서는 계약 요금이 적용되는 기존 계약에서 대체 비공개 제안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType" : "CreateReplacementOffer",
```



```

        "Entity": {
            "Type": "Offer@1.0"
        },
        "ChangeName": "CreateReplacementOffer",
        "DetailsDocument": {
            "AgreementId": "agmt-11111111111111111111111111111111"
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateReplacementOffer.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test replacement offer for SaaSProduct using AWS
Marketplace API Reference Codes",
            "Description": "Test private replacement offer with contract
pricing for SaaSProduct"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateReplacementOffer.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "FixedUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "Price": "0.0",
                    "Grants": [
                        {
                            "DimensionKey": "BasicService",
                            "MaxQuantity": 2
                        }
                    ]
                }
            ]
        }
    }
],
},
},

```

```

{
  "ChangeType": "UpdateValidityTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateReplacementOffer.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "ValidityTerm",
        "AgreementEndDate": "2024-01-30"
      }
    ]
  }
},
{
  "ChangeType": "UpdatePaymentScheduleTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateReplacementOffer.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "PaymentScheduleTerm",
        "CurrencyCode": "USD",
        "Schedule": [
          {
            "ChargeDate": "2024-01-01",
            "ChargeAmount": "0"
          }
        ]
      }
    ]
  }
},
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateReplacementOffer.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [

```

```

        {
            "Type": "LegalTerm",
            "Documents": [
                {
                    "Type": "StandardEula",
                    "Version": "2022-07-14"
                }
            ]
        }
    ],
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateReplacementOffer.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-12-31"
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateReplacementOffer.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType" : "CreateReplacementOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateReplacementOffer",
      "DetailsDocument": {
        "AgreementId": "agmt-11111111111111111111111111111111"
      }
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateReplacementOffer.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "Test replacement offer for SaaSProduct using AWS Marketplace API Reference Codes",
        "Description": "Test private replacement offer with contract pricing for SaaSProduct"
      }
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateReplacementOffer.Entity.Identifier"
      },
    },
  ]
}
```

```

        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "FixedUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "Price": "0.0",
                    "Grants": [
                        {
                            "DimensionKey": "BasicService",
                            "MaxQuantity": 2
                        }
                    ]
                }
            ]
        },
        {
            "ChangeType": "UpdateValidityTerms",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "$CreateReplacementOffer.Entity.Identifier"
            },
            "DetailsDocument": {
                "Terms": [
                    {
                        "Type": "ValidityTerm",
                        "AgreementEndDate": "2024-01-30"
                    }
                ]
            }
        },
        {
            "ChangeType": "UpdatePaymentScheduleTerms",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "$CreateReplacementOffer.Entity.Identifier"
            },
            "DetailsDocument": {
                "Terms": [
                    {
                        "Type": "PaymentScheduleTerm",
                        "CurrencyCode": "USD",
                        "Schedule": [

```

```

        {
            "ChargeDate": "2024-01-01",
            "ChargeAmount": "0"
        }
    ]
}
],
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateReplacementOffer.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateReplacementOffer.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateReplacementOffer.Entity.Identifier"
    }
}

```

```

        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a replacement private
offer
from an existing agreement with contract pricing
CAPI-95
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create replacement private offer with contract pricing..",
    )

    return response

```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 공개 제안 설명

다음 코드 예제에서는 공개 제안을 설명하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;

public class DescribeEntity {

    /*
```



```
    * Describe my AMI or SaaS or Container product and check if it contains all the
    information I need to know about the product
    */
    public static void main(String[] args) {

        String offerId = args.length > 0 ? args[0] : OFFER_ID;

        DescribeEntityResponse describeEntityResponse =
            getDescribeEntityResponse(offerId);

        ReferenceCodesUtils.formatOutput(describeEntityResponse);
    }

    public static DescribeEntityResponse getDescribeEntityResponse(String offerId) {
        MarketplaceCatalogClient marketplaceCatalogClient =
            MarketplaceCatalogClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        DescribeEntityRequest describeEntityRequest =
            DescribeEntityRequest.builder()
                .catalog(AWS_MP_CATALOG)
                .entityId(offerId)
                .build();

        DescribeEntityResponse describeEntityResponse =
            marketplaceCatalogClient.describeEntity(describeEntityRequest);
        return describeEntityResponse;
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeEntity](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) search for offer information in
the AWS Marketplace Catalog
CAPI-29
"""

import json
import logging

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

OFFER_ID = "offer-11111111111111"

def pretty_print(response):
    json_object = json.dumps(response, indent=4)
    print(json_object)

def get_offer_information(mp_client, entity_id):
    """
    Returns information about a given offer
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of offer information
    """
```

```

    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace",
            EntityId=entity_id,
        )

        return response

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Offer with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for an offer in the AWS Marketplace Catalog.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-catalog")

    pretty_print(get_offer_information(mp_client, OFFER_ID))

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeEntity](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 비공개 제안 만료

다음 코드 예제에서는 구매자가 더 이상 제안을 볼 수 없도록 비공개 제안의 만료 날짜를 과거 날짜로 설정하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2023-01-01"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "Test Private Offer"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create "draft" Private Offer
for any AMI or SAAS product type that can be reviewed internally
before publishing to buyers
CAPI-30
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Private offer for AMI product")
```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 모든 비공개 제안 나열

다음 코드 예제에서는 모든 비공개 제안을 나열하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import java.util.ArrayList;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;
import software.amazon.awssdk.services.marketplacecatalog.model.EntitySummary;
```

```

import
    software.amazon.awssdk.services.marketplacecatalog.model.EntityTypeFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesResponse;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferAvailabilityEndDateFilter;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferAvailabilityEndDateFilterD
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferBuyerAccountsFilter;
import software.amazon.awssdk.services.marketplacecatalog.model.OfferFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferReleaseDateFilter;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferReleaseDateFilterDateRange
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferTargetingFilter;

public class ListAllPrivateOffers {

    /*
     * List all my private offers and sort or filter them by Offer Publish Date,
     Offer Expiry Date and Buyer IDs
     *
     * OfferTargetingFilter = BuyerAccounts (private offer);
     * OfferBuyerAccountsFilter: Buyer IDs filter
     * OfferAvailabilityEndDateFilter : Offer Expiry Date filter
     * OfferReleaseDateFilter : Offer Publish Date filter
     */

    private static MarketplaceCatalogClient marketplaceCatalogClient =
        MarketplaceCatalogClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    public static void main(String[] args) {

        String offerReleaseDateAfterValue = "2023-01-01T23:59:59Z";
        String offerAvailableEndDateAfterValue = "2040-12-24T23:59:59Z";
    }
}

```

```

    List<EntitySummary> entitySummaryList =
getEntitySummaryList(offerReleaseDateAfterValue,
offerAvailableEndDateAfterValue);

// for each offer id, output the offer detail using DescribeEntity API

for (EntitySummary entitySummary : entitySummaryList) {
    DescribeEntityRequest describeEntityRequest =
        DescribeEntityRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityId(entitySummary.entityId())
            .build();
    DescribeEntityResponse describeEntityResponse =
marketplaceCatalogClient.describeEntity(describeEntityRequest);
    ReferenceCodesUtils.formatOutput(describeEntityResponse);
}
}

public static List<EntitySummary> getEntitySummaryList (String
offerReleaseDateAfterValue, String offerAvailableEndDateAfterValue) {

    EntityTypeFilters entityTypeFilters =
        EntityTypeFilters.builder()
            .offerFilters(OfferFilters.builder()
                .targeting(OfferTargetingFilter.builder()
                    .valueListWithStrings(OFFER_TARGETING_BUYERACCOUNTS)
                    .build())
                .buyerAccounts(OfferBuyerAccountsFilter.builder()
                    .wildCardValue(BUYER_ACCOUNT_ID)
                    .build())
                .availabilityEndDate(OfferAvailabilityEndDateFilter.builder()
                    .dateRange(OfferAvailabilityEndDateFilterDateRange.builder()
                        .afterValue(offerAvailableEndDateAfterValue).build())
                    .build())
                .releaseDate(OfferReleaseDateFilter.builder()
                    .dateRange(OfferReleaseDateFilterDateRange.builder()
                        .afterValue(offerReleaseDateAfterValue)
                        .build())
                    .build())
                .build()
            ).build();

    ListEntitiesRequest listEntitiesRequest =

```



```
ListEntitiesRequest.builder()
    .catalog(AWS_MP_CATALOG)
    .entityType(ENTITY_TYPE_OFFER).maxResults(10)
    .entityTypeFilters(entityTypeFilters)
    .nextToken(null)
    .build();

ListEntitiesResponse listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);
List<EntitySummary> entitySummaryList = new ArrayList<EntitySummary>();

entitySummaryList.addAll(listEntitiesResponse.entitySummaryList());

while ( listEntitiesResponse.nextToken() != null &&
listEntitiesResponse.nextToken().length() > 0) {
    listEntitiesRequest =
        ListEntitiesRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityType(ENTITY_TYPE_OFFER)
            .maxResults(10)
            .entityTypeFilters(entityTypeFilters)
            .nextToken(listEntitiesResponse.nextToken())
            .build();
    listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);
    entitySummaryList.addAll(listEntitiesResponse.entitySummaryList());
}

return entitySummaryList;
}
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) for listing offers in the AWS
Marketplace Catalog
CAPI-40
"""
import json
import logging

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# Constants
MAX_RESULTS = 10
CATALOG = "AWSMarketplace"
ENTITY_TYPE = "Offer"

def pretty_print(response):
    json_object = json.dumps(response, indent=4)
    print(json_object)

def list_private_offers(mp_client, return_all_private_offers):
    """
    This method retrieves list of all Private Offers for this account.
    """
    entity_summary_list = []
```

```

filter_list_param = {
    'OfferFilters': {
        'Targeting': {
            'ValueList': ["BuyerAccounts"]
        }
    }
}

try:
    response = mp_client.list_entities(
        Catalog=CATALOG,
        EntityType=ENTITY_TYPE,
        EntityTypeFilters=filter_list_param,
        MaxResults=MAX_RESULTS
    )
except ClientError as e:
    logger.error("Could not complete list_entities request: %s", e)
    raise

entity_summary_list.extend(response["EntitySummaryList"])
logger.info("Number of results in first iteration: %d " %
len(entity_summary_list))

# Get subsequent pages of results if previous response contained a NextToken
while "NextToken" in response and return_all_private_offers:
    try:
        logger.info("Getting Next Token results: %s " %
response["NextToken"])
        response = mp_client.list_entities(
            Catalog=CATALOG,
            EntityType=ENTITY_TYPE,
            EntityTypeFilters=filter_list_param,
            MaxResults=MAX_RESULTS,
            NextToken=response["NextToken"]
        )
    except ClientError as e:
        logger.error("Could not complete list_entities request: %s", e)
        raise

    entity_summary_list.extend(response["EntitySummaryList"])
    logger.info(
        "Number of results in the current iteration: %d "
        % len(response["EntitySummaryList"])
    )

```

```
return entity_summary_list

def get_offer_details(mp_client, offer):
    """
    Describe the details of the Offer.
    """
    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace", EntityId=offer["EntityId"]
        )

        return response
    except ClientError:
        logger.exception("Error: Couldn't get details of the Offer.")
        raise

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Demo - List Private offers.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-catalog")

    # Get list of all Offers.
    private_offers = list_private_offers(mp_client, False)
    count = len(private_offers)

    logger.info("Number of Offers: %d " % count)
    offer_counter = 0
    # Display details of each Offer.
    for offer in private_offers:
        print("-" * 88)
        offer_counter += 1
        print("Displaying Offer details for Offer# %d" % offer_counter)
        entity = get_offer_details(mp_client, offer)
        pretty_print(entity)

    print("-" * 88)
```

```
if __name__ == "__main__":
    usage_demo()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 특정 제품 ID에 대해 릴리스된 공개 및 비공개 제안 나열

다음 코드 예제에서는 특정 제품 ID에 대해 릴리스된 공개 및 비공개 제안을 나열하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import java.util.ArrayList;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import software.amazon.awssdk.services.marketplacecatalog.model.EntitySummary;
import
    software.amazon.awssdk.services.marketplacecatalog.model.EntityTypeFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesRequest;
```

```

import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesResponse;
import software.amazon.awssdk.services.marketplacecatalog.model.OfferFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferProductIdFilter;
import software.amazon.awssdk.services.marketplacecatalog.model.OfferStateFilter;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferTargetingFilter;

public class ListProductPublicOrPrivateReleasedOffers {

    /*
     * List released Public/Private offers for a specific product id.
     * Example below is to list released public offers.
     * To change to released private offers, change OFFER_TARGETING_NONE (None) to
     * OFFER_TARGETING_BUYERACCOUNTS(BuyerAccounts)
     */
    public static void main(String[] args) {

        List<EntitySummary> entitySummaryList = getEntitySummaryList();
        ReferenceCodesUtils.formatOutput(entitySummaryList);
    }

    public static List<EntitySummary> getEntitySummaryList() {
        MarketplaceCatalogClient marketplaceCatalogClient =
            MarketplaceCatalogClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        // define list entities filters

        EntityTypeFilters entityTypeFilters =
            EntityTypeFilters.builder()
                .offerFilters(OfferFilters.builder()
                    .targeting(OfferTargetingFilter.builder()
                        .valueListWithStrings(OFFER_TARGETING_NONE)
                        .build())
                    .state(OfferStateFilter.builder()
                        .valueListWithStrings(OFFER_STATE_RELEASED)
                        .build())
                    .productId(OfferProductIdFilter.builder()
                        .valueList(PRODUCT_ID)
                        .build())
                .build()
    }

```

```

        .build())
        .build();

ListEntitiesRequest listEntitiesRequest =
    ListEntitiesRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .entityType(ENTITY_TYPE_OFFER)
        .maxResults(10)
        .entityTypeFilters(entityTypeFilters)
        .nextToken(null)
        .build();

ListEntitiesResponse listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);

// save all entitySummary of the results into entitySummaryList

List<EntitySummary> entitySummaryList = new ArrayList<EntitySummary>();

entitySummaryList.addAll(listEntitiesResponse.entitySummaryList());

while ( listEntitiesResponse.nextToken() != null &&
listEntitiesResponse.nextToken().length() > 0) {
    listEntitiesRequest =
        ListEntitiesRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityType(ENTITY_TYPE_OFFER)
            .maxResults(10)
            .entityTypeFilters(entityTypeFilters)
            .nextToken(listEntitiesResponse.nextToken())
            .build();
    listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);
    entitySummaryList.addAll(listEntitiesResponse.entitySummaryList());
}
return entitySummaryList;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#) [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 Pay-As-You-Go 요금 계약을 적용하도록 제안 업데이트

다음 코드 예제에서는 종량제 요금으로 계약을 적용하도록 제안을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
          {
            "Type": "UsageBasedPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
              {
                "RateCard": [
                  {
                    "DimensionKey": "WorkloadSmall",
                    "Price": "0.15"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```



```

        {
            "DimensionKey": "WorkloadMedium",
            "Price": "0.25"
        }
    ]
}
],
{
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
        {
            "Selector": {
                "Type": "Duration",
                "Value": "P12M"
            },
            "RateCard": [
                {
                    "DimensionKey": "BasicService",
                    "Price": "150"
                },
                {
                    "DimensionKey": "PremiumService",
                    "Price": "300"
                }
            ],
            "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
            }
        }
    ]
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
          {
            "Type": "UsageBasedPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
              {
                "RateCard": [
                  {
                    "DimensionKey": "WorkloadSmall",
                    "Price": "0.15"
                  },
                  {
                    "DimensionKey": "WorkloadMedium",
                    "Price": "0.25"
                  }
                ]
              }
            ]
          }
        ]
      },
      {
        "Type": "ConfigurableUpfrontPricingTerm",
```

```

        "CurrencyCode": "USD",
        "RateCards": [
            {
                "Selector": {
                    "Type": "Duration",
                    "Value": "P12M"
                },
                "RateCard": [
                    {
                        "DimensionKey": "BasicService",
                        "Price": "150"
                    },
                    {
                        "DimensionKey": "PremiumService",
                        "Price": "300"
                    }
                ],
                "Constraints": {
                    "MultipleDimensionSelection": "Allowed",
                    "QuantityConfiguration": "Allowed"
                }
            }
        ]
    }
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update an offer to apply
contract with PAYG pricing
CAPI-21
"""

import os

```

```
import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Update offer to apply contract with PAYG pricing")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 시간당 연간 요금을 적용하도록 제안 업데이트

다음 코드 예제에서는 시간 단위 연간 요금을 적용하도록 제안을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
    "Catalog": "AWSMarketplace",
```

```

"ChangeSet": [
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "offer-111111111111"
    },
    "DetailsDocument": {
      "PricingModel": "Usage",
      "Terms": [
        {
          "Type": "UsageBasedPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "RateCard": [
                {
                  "DimensionKey": "m5.large",
                  "Price": "0.13"
                }
              ]
            }
          ]
        },
        {
          "Type": "ConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P365D"
              },
              "RateCard": [
                {
                  "DimensionKey": "m5.large",
                  "Price": "20.03"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  }
]

```

```

    }
  }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "UsageBasedPricingTerm",
            "CurrencyCode": "USD",
            "RateCards": [
              {
                "RateCard": [
                  {
                    "DimensionKey": "m5.large",
                    "Price": "0.13"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}

```

```

        ]
    }
}
],
{
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
        {
            "Selector": {
                "Type": "Duration",
                "Value": "P365D"
            },
            "RateCard": [
                {
                    "DimensionKey": "m5.large",
                    "Price": "20.03"
                }
            ],
            "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
            }
        }
    ]
}
]
}
]
}
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update an offer to apply
hourly annual pricing
CAPI-20
"""

```

```
import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Update offer with hourly annual pricing")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 특정 지리적 리전에 대상 지정을 적용하도록 제안 업데이트

다음 코드 예제에서는 특정 지리적 리전을 대상으로 적용하도록 제안을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "ES",
            "FR",
            "AU"
          ]
        }
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
```

```

    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "ES",
            "FR",
            "AU"
          ]
        }
      }
    }
  ]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update an offer to apply
targeting to certain geographic regions.
CAPI-19
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

```

```
def main():
    sc.usage_demo(change_set, "Update offer targeting")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 공개 제안의 이름 및 설명 업데이트

다음 코드 예제에서는 공개 제안의 이름과 설명을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "Terms": [
```

```

        {
            "Type": "LegalTerm",
            "Documents": [
                {
                    "Type": "CustomEula",
                    "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                }
            ]
        }
    ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "UpdateInformation",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "offer-111111111111"
            },
            "DetailsDocument": {
                "Name": "New offer name",
                "Description": "New offer description"
            }
        }
    ]
}

```

```
]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update name and description of
my offer
CAPI-18
"""

import os

import utils.start_changeset as sc # type: ignore
import utils.stringify_details as sd # type: ignore

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Update name and description of my offer")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 제안의 EULA 업데이트

다음 코드 예제에서는 제안의 EULA를 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "Name": "New offer name",
        "Description": "New offer description"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "CustomEula",
                "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update EULA of my offer
CAPI-18
"""

import os

import utils.start_changeset as sc # type: ignore
import utils.stringify_details as sd # type: ignore
```

```

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Update EULA of my offer")

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 비공개 제안의 만료 날짜를 미래 날짜로 업데이트

다음 코드 예제에서는 구매자에게 제안을 평가하고 수락할 시간을 더 많이 제공할 수 있도록 비공개 제안의 만료 날짜를 미래 날짜로 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {

```



```

        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "offer-111111111111"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2026-01-01"
        }
    }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "UpdateAvailability",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "offer-111111111111"
            },
            "DetailsDocument": {
                "AvailabilityEndDate": "2026-01-01"
            }
        }
    ]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to set expiry date of a private
offer to a date in the future so that my buyers get more time to evaluate and
accept the offer.
CAPI-37
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Update offer expiration date")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 제품에 대한 공개 무료 평가판 제안의 무료 평가판 기간 업데이트

다음 코드 예제에서는 SaaS 제품에 대한 공개 무료 평가판 제안의 무료 평가판 기간을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-1111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "FreeTrialPricingTerm",
            "Duration": "P21D",
            "Grants": [
              {
                "DimensionKey": "WorkloadSmall"
              },
              {
                "DimensionKey": "WorkloadMedium"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-1111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "FreeTrialPricingTerm",
            "Duration": "P21D",
            "Grants": [
              {
                "DimensionKey": "WorkloadSmall"
              },
              {
                "DimensionKey": "WorkloadMedium"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to change free trial duration for
my SaaS product by modifying my free trial public offer
CAPI-14
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Change free trial duration for SaaS product")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 제안의 환불 정책 업데이트

다음 코드 예제에서는 제안의 환불 정책을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "SupportTerm",
            "RefundPolicy": "Updated refund policy description"
          }
        ]
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "SupportTerm",
            "RefundPolicy": "Updated refund policy description"
          }
        ]
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update refund policy of my
offer
CAPI-18
"""
```

```
import os

import utils.start_changeset as sc # type: ignore
import utils.stringify_details as sd # type: ignore

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Update refund policy of my public offer")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API용 제품

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 AMI, SaaS 또는 컨테이너 제품 설명](#)
- [AWSSDK를 사용하여 모든 AMI, SaaS 또는 컨테이너 제품 및 관련 공개 제안 나열](#)

AWSSDK를 사용하여 AMI, SaaS 또는 컨테이너 제품 설명

다음 코드 예제에서는 AMI, SaaS 또는 컨테이너 제품을 설명하고 제품에 대해 알고 싶은 모든 정보가 포함되어 있는지 확인하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;

public class DescribeEntity {

    /*
     * Describe my AMI or SaaS or Container product and check if it contains all the
     information I need to know about the product
     */
    public static void main(String[] args) {

        String offerId = args.length > 0 ? args[0] : OFFER_ID;

        DescribeEntityResponse describeEntityResponse =
            getDescribeEntityResponse(offerId);

        ReferenceCodesUtils.formatOutput(describeEntityResponse);
    }

    public static DescribeEntityResponse getDescribeEntityResponse(String offerId) {
        MarketplaceCatalogClient marketplaceCatalogClient =
```

```

MarketplaceCatalogClient.builder()
    .httpClient(ApacheHttpClient.builder().build())
    .credentialsProvider(ProfileCredentialsProvider.create())
    .build();

DescribeEntityRequest describeEntityRequest =
    DescribeEntityRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .entityId(offerId)
        .build();

DescribeEntityResponse describeEntityResponse =
    marketplaceCatalogClient.describeEntity(describeEntityRequest);
return describeEntityResponse;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeEntity](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) search for product information in
the AWS Marketplace Catalog
CAPI-28
"""

import json
import logging

import boto3
from botocore.exceptions import ClientError

```

```
logger = logging.getLogger(__name__)

PRODUCT_ID = "prod-111111111111"

def pretty_print(response):
    json_object = json.dumps(response, indent=4)
    print(json_object)

def get_product_information(mp_client, entity_id):
    """
    Returns information about a given product
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of product information
    """

    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace",
            EntityId=entity_id,
        )

        return response

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Product with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for a product in the AWS Marketplace Catalog.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-catalog")

    pretty_print(get_product_information(mp_client, PRODUCT_ID))
```

```
if __name__ == "__main__":
    usage_demo()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeEntity](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 모든 AMI, SaaS 또는 컨테이너 제품 및 관련 공개 제안 나열

다음 코드 예제에서는 모든 AMI, SaaS 또는 컨테이너 제품 및 관련 공개 제안을 나열하는 방법을 보여 줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import software.amazon.awssdk.services.marketplacecatalog.model.EntitySummary;
```

```

import
    software.amazon.awssdk.services.marketplacecatalog.model.EntityTypeFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.ListEntitiesResponse;
import software.amazon.awssdk.services.marketplacecatalog.model.OfferFilters;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferProductIdFilter;
import software.amazon.awssdk.services.marketplacecatalog.model.OfferStateFilter;
import
    software.amazon.awssdk.services.marketplacecatalog.model.OfferTargetingFilter;

public class ListEntities {

    /*
     * List all my AMI or SaaS or Container products and associated public offers
     */
    public static void main(String[] args) {

        Map<String, List<EntitySummary>> allProductsWithOffers =
            getAllProductsWithOffers();

        ReferenceCodesUtils.formatOutput(allProductsWithOffers);
    }

    public static Map<String, List<EntitySummary>> getAllProductsWithOffers() {
        MarketplaceCatalogClient marketplaceCatalogClient =
            MarketplaceCatalogClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        Map<String, List<EntitySummary>> allProductsWithOffers = new HashMap<String,
            List<EntitySummary>> ();

        // get all product entities
        List<EntitySummary> productEntityList = new ArrayList<EntitySummary>();

        ListEntitiesRequest listEntitiesRequest =
            ListEntitiesRequest.builder()
                .catalog(AWS_MP_CATALOG)
                .entityType(PRODUCT_TYPE_AMI)
                .maxResults(10)
    
```

```
.nextToken(null)
.build();

ListEntitiesResponse listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);

productEntityList.addAll(listEntitiesResponse.entitySummaryList());

while (listEntitiesResponse.nextToken() != null) {
    listEntitiesRequest =
        ListEntitiesRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityType(PRODUCT_TYPE_AMI)
            .maxResults(10)
            .nextToken(listEntitiesResponse.nextToken())
            .build();
    listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);
    productEntityList.addAll(listEntitiesResponse.entitySummaryList());
}

// loop through each product entity and get the public released offers
associated using product id filter

for ( EntitySummary productEntitySummary : productEntityList) {
    EntityTypeFilters entityTypeFilters =
        EntityTypeFilters.builder()
            .offerFilters(OfferFilters.builder()
                .targeting(OfferTargetingFilter.builder()
                    .valueListWithStrings(OFFER_TARGETING_NONE)
                    .build())
                .state(OfferStateFilter.builder()
                    .valueListWithStrings(OFFER_STATE_RELEASED)
                    .build())
                .productId(OfferProductIdFilter.builder()
                    .valueList(productEntitySummary.entityId())
                    .build())
                .build()
            ).build();

    listEntitiesRequest =
        ListEntitiesRequest.builder()
```

```

        .catalog(AWS_MP_CATALOG)
        .entityType(ENTITY_TYPE_OFFER)
        .maxResults(10)
        .entityTypeFilters(entityTypeFilters)
        .nextToken(null)
        .build();

    listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);

    // save all entitySummary of the results into entitySummaryList

    List<EntitySummary> offerEntitySummaryList = new ArrayList<EntitySummary>();

    offerEntitySummaryList.addAll(listEntitiesResponse.entitySummaryList());

    while ( listEntitiesResponse.nextToken() != null &&
listEntitiesResponse.nextToken().length() > 0) {
        listEntitiesRequest =
            ListEntitiesRequest.builder()
                .catalog(AWS_MP_CATALOG)
                .entityType(ENTITY_TYPE_OFFER)
                .maxResults(10)
                .entityTypeFilters(entityTypeFilters)
                .nextToken(listEntitiesResponse.nextToken())
                .build();
        listEntitiesResponse =
marketplaceCatalogClient.listEntities(listEntitiesRequest);
        offerEntitySummaryList.addAll(listEntitiesResponse.entitySummaryList());
    }

    // save final results into map; key = product id; value = offer entity summary
list

    allProductsWithOffers.put(productEntitySummary.entityId(),
offerEntitySummaryList);
}
return allProductsWithOffers;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 다음 주제를 참조하세요.

- [DescribeEntity](#)
- [ListEntities](#)

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to display information about AMI
products and their associated offers in the AWS Marketplace Catalog
CAPI-27
"""

import json
import logging

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

MAX_PAGE_RESULTS = 10

try:
    mp_client = boto3.client("marketplace-catalog")
except ClientError as e:
    logger.error("Could not create boto3 client.")
    raise

def pretty_print(response):
    json_object = json.dumps(response, indent=4)
    print(json_object)
```



```
def describe_entity(entity_id):
    """
    Returns entity details
    Args: entity_id str: The entity ID of the product or offer
    Returns: dict: The entity details
    """
    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace", EntityId=entity_id
        )
    except ClientError as e:
        logger.error("Could not complete describe_entity request.")
        raise

    # De-stringify the details
    response["Details"] = json.loads(response["Details"])

    return response


def get_entities(entity_type, visibility=None):
    """
    Returns list of entities for provided entity_type
    Args: entity_type str: Type of entity list to return, in our case AmiProduct
    or Offer
    Returns: list: Abbreviated list of entity information
    """
    EntitySummaryList = []

    # Get the first page of results
    try:
        response = mp_client.list_entities(
            Catalog="AWSMarketplace",
            EntityType=entity_type,
            MaxResults=MAX_PAGE_RESULTS,
        )
    except ClientError as e:
        logger.error("Could not complete list_entities request.")
        raise

    EntitySummaryList.extend(response["EntitySummaryList"])

    # Get subsequent pages of results if previous response contained a NextToken
```

```

while "NextToken" in response:
    try:
        response = mp_client.list_entities(
            Catalog="AWSMarketplace",
            EntityType=entity_type,
            MaxResults=MAX_PAGE_RESULTS,
            NextToken=response["NextToken"],
        )
    except ClientError as e:
        logger.error("Could not complete list_entities request.")
        raise

    EntitySummaryList.extend(response["EntitySummaryList"])

    # if visibility is provided, filter the list to only include entities with
    # that visibility
    if visibility is not None:
        EntitySummaryList = [
            entity for entity in EntitySummaryList if entity["Visibility"] ==
visibility
        ]

    return EntitySummaryList

def get_enhanced_product_list(entity_type):
    """
    Returns an enhanced list of products with product details and offer details
    Args: entity_type str: Type of entity list to return, in our case AmiProduct
    Returns: list: Enhanced list of dictionary objects containing product and
    offer details
    """

    product_list = get_entities(entity_type)

    # Loop through product list and append product details to each product
    for product in product_list:
        # appends product details to product dictionary
        product["ProductDetails"] = describe_entity(product["EntityId"])
["Details"]
        # creating an empty list for offer details
        product["OfferDetailsList"] = []

    return product_list

```

```

def attach_offer_details(product_list):
    """
    Loops through offer information and appends offer details to product list
    Args: product_list list: List of product dictionaries
    Returns: list: Enhanced list of dictionary objects containing product and
    offer details
    """
    offer_list = get_entities("Offer", "Public")

    # Loop through offer list and append offer details to each product
    for offer in offer_list:
        offer["OfferDetails"] = describe_entity(offer["EntityId"])["Details"]

        # Extracts product-id from offer
        product_id = offer["OfferDetails"]["ProductId"]

        # Determines if product-id referenced in offer matches product-id in
        product list
        product_dict = next(
            filter(lambda product: product["EntityId"] == product_id,
                product_list),
            None,
        )

        # If product-id matches, appends offer details to product dictionary
        if product_dict is not None:
            # logger.info(f"Offer product Id {offer['OfferDetails']['ProductId']}
            found in product dictionary. Updating product dictionary with offer details")
            product_dict["OfferDetailsList"].append(offer["OfferDetails"])

        else:
            # logger.info("Offer product Id {offer['OfferDetails']['ProductId']}
            not found. Skipping offer details update")
            pass

    return product_list

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)

```

```

print("Retrieving products and public offer information....")
print("-" * 88)

# Builds a list of products and their details
product_list = get_enhanced_product_list("AmiProduct")

# Queries offer information and attaches it to the product list
product_offer_list = attach_offer_details(product_list)

pretty_print(product_offer_list)
return product_offer_list

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWSSDK for Python (Boto3) API 참조의 다음 주제를 참조하세요.
 - [DescribeEntity](#)
 - [ListEntities](#)

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API에 대한 재판매 권한 부여

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 모든 제품 유형에 대한 초안 재판매 권한 부여 생성](#)
- [AWSSDK를 사용하여 재판매 권한 부여 설명](#)
- [AWSSDK를 사용하여 비공개 제안으로 일회성 재판매 권한 부여 게시](#)
- [AWSSDK를 사용하여 만료 날짜로 다중 사용 재판매 권한 부여 게시](#)
- [AWSSDK를 사용하여 구매자에게 보낼 만료 날짜 및 EULA와 함께 다중 사용 재판매 권한 부여 게시](#)
- [만료 날짜가 포함된 다중 사용 재판매 권한 부여 게시 및 AWSSDK를 사용하여 리셀러 계약 설명서 추가](#)

- [만료와 함께 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.](#)
- [AWSSDK를 사용하여 만료 날짜 없이 다중 사용 재판매 권한 부여 게시](#)
- [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 사용자 지정 EULA 추가](#)
- [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 리셀러 계약 문서 추가](#)
- [만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.](#)
- [AWSSDK를 사용하여 일회성 재판매 권한 부여 게시 및 유연한 결제 일정 추가](#)
- [AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 EULA 추가](#)
- [SDK를 사용하여 일회성 재판매 권한 부여를 게시하고 재판매에 대한 특정 구매자 계정을 추가합니다.AWS](#)
- [AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 리셀러 계약 문서 추가](#)
- [에 대한 일회성 재판매 권한 부여 게시 및 AWSSDK를 사용한 갱신 여부 추가](#)
- [AWSSDK를 사용하여 재판매 권한 부여 제한](#)
- [AWSSDK를 사용하여 일회성 또는 다목적 재판매 권한 부여의 이름 및 설명 업데이트](#)

AWSSDK를 사용하여 모든 제품 유형에 대한 초안 재판매 권한 부여 생성

다음 코드 예제에서는 채널 파트너에게 게시하기 전에 내부적으로 검토할 수 있도록 모든 제품 유형에 대한 재판매 권한 부여 초안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
```

```

        "Type": "ResaleAuthorization@1.0"
    },
    "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    }
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a one-time resale authorization on my SaaS/AMI/Container product so my CP
can use that to create Channel Partner Private Offer (CPP0)
CAPI-41
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "draft resale auth")

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 재판매 권한 부여 설명

다음 코드 예제에서는 재판매 권한 부여를 설명하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.catalogapi;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;

public class DescribeEntity {

    /*
     * Describe my AMI or SaaS or Container product and check if it contains all the
     information I need to know about the product
     */
    public static void main(String[] args) {

        String offerId = args.length > 0 ? args[0] : OFFER_ID;
```



```

    DescribeEntityResponse describeEntityResponse =
        getDescribeEntityResponse(offerId);

    ReferenceCodesUtils.formatOutput(describeEntityResponse);
}

public static DescribeEntityResponse getDescribeEntityResponse(String offerId) {
    MarketplaceCatalogClient marketplaceCatalogClient =
        MarketplaceCatalogClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    DescribeEntityRequest describeEntityRequest =
        DescribeEntityRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityId(offerId)
            .build();

    DescribeEntityResponse describeEntityResponse =
        marketplaceCatalogClient.describeEntity(describeEntityRequest);
    return describeEntityResponse;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeEntity](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""

```

Purpose

Shows how to use the AWS SDK for Python (Boto3) search for product information in the AWS Marketplace Catalog

"""

```
import json
import logging
```

```
import boto3
from botocore.exceptions import ClientError
```

```
logger = logging.getLogger(__name__)
```

```
resaleAuthorizationId = "resaleauthz-111111111111"
```

```
def pretty_print(response):
    json_object = json.dumps(response, indent=4)
    print(json_object)
```

```
def get_product_information(mp_client, entity_id):
    """
    Returns information about a given product
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of product information
    """
```

```
    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace",
            EntityId=entity_id,
        )

        return response

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Product with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)
```

```
def usage_demo():
```

```

logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

print("-" * 88)
print("Looking for a product in the AWS Marketplace Catalog.")
print("-" * 88)

mp_client = boto3.client("marketplace-catalog")

pretty_print(get_product_information(mp_client, resaleAuthorizationId))

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeEntity](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 비공개 제안으로 일회성 재판매 권한 부여 게시

다음 코드 예제에서는 채널 파트너가 권한을 사용하여 채널 파트너 비공개 제안(CPPO)을 생성할 수 있도록 비공개 제안과 함께 일회성 재판매 권한 부여를 게시하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [

```

```

{
  "ChangeType": "CreateResaleAuthorization",
  "ChangeName": "ResaleAuthorization",
  "Entity": {
    "Type": "ResaleAuthorization@1.0"
  },
  "DetailsDocument": {
    "ProductId": "prod-111111111111",
    "Name": "TestResaleAuthorization",
    "Description": "Worldwide ResaleAuthorization for Test Product",
    "ResellerAccountId": "111111111111"
  }
},
{
  "ChangeType": "ReleaseResaleAuthorization",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {}
},
{
  "ChangeType": "UpdatePricingTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "PricingModel": "Contract",
    "Terms": [
      {
        "Type": "ResaleConfigurableUpfrontPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
          {
            "Selector": {
              "Type": "Duration",
              "Value": "P12M"
            },
            "RateCard": [
              {
                "DimensionKey": "t2.small",
                "Price": "150"
              }
            ]
          }
        ]
      }
    ]
  }
}

```

```

        ],
        "Constraints": {
            "MultipleDimensionSelection": "Allowed",
            "QuantityConfiguration": "Allowed"
        }
    }
}
],
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "BuyerLegalTerm",
                "Documents": [
                    {
                        "Type": "CustomEula",
                        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "OffersMaxQuantity": 1
    }
}
]

```

```
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
```

```

        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ResaleConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.small",
                                "Price": "150"
                            }
                        ],
                        "Constraints": {
                            "MultipleDimensionSelection": "Allowed",
                            "QuantityConfiguration": "Allowed"
                        }
                    }
                ]
            }
        ]
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "BuyerLegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",

```

```

        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
    }
}
    }
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "OffersMaxQuantity": 1
    }
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a one-time resale authorization on my SaaS/AMI/Container product so my CP
can use that to create Channel Partner Private Offer (CPP0)
CAPI-42
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)

```



```

        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set, "onetime resale auth with private offer"
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 만료 날짜로 다중 사용 재판매 권한 부여 게시

다음 코드 예제에서는 채널 파트너가 권한을 사용하여 CPPO를 생성할 수 있도록 시간 단위 연간 요금 이 적용되는 AMI 제품에 대한 만료 날짜가 있는 다중 사용 재판매 권한 부여를 게시하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
```

```

"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType": "CreateResaleAuthorization",
    "ChangeName": "ResaleAuthorization",
    "Entity": {
      "Type": "ResaleAuthorization@1.0"
    },
    "DetailsDocument": {
      "ProductId": "prod-111111111111",
      "Name": "TestResaleAuthorization",
      "Description": "Worldwide ResaleAuthorization for Test Product",
      "ResellerAccountId": "111111111111"
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",

```

```

        "Terms": [
            {
                "Type": "ResaleConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.small",
                                "Price": "150"
                            }
                        ],
                        "Constraints": {
                            "MultipleDimensionSelection": "Allowed",
                            "QuantityConfiguration": "Allowed"
                        }
                    }
                ]
            }
        ],
        {
            "ChangeType": "UpdateAvailability",
            "Entity": {
                "Type": "ResaleAuthorization@1.0",
                "Identifier": "$ResaleAuthorization.Entity.Identifier"
            },
            "DetailsDocument": {
                "AvailabilityEndDate": "2023-05-31"
            }
        },
        {
            "ChangeType": "ReleaseResaleAuthorization",
            "Entity": {
                "Type": "ResaleAuthorization@1.0",
                "Identifier": "$ResaleAuthorization.Entity.Identifier"
            },
            "DetailsDocument": {}
        }
    ]
}

```

```
]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "BuyerLegalTerm",
            "Documents": [
```

```

        {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
        }
    ]
}
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ResaleConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.small",
                                "Price": "150"
                            }
                        ],
                        "Constraints": {
                            "MultipleDimensionSelection": "Allowed",
                            "QuantityConfiguration": "Allowed"
                        }
                    }
                ]
            }
        ]
    }
},

```

```

    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-05-31"
        }
    },
    {
        "ChangeType": "ReleaseResaleAuthorization",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a multi-use resale authorization with expiry date on my SaaS/AMI product
so my CP can use that to create Channel Partner Private Offer (CPP0)
CAPI-48
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

```

```
def main():
    sc.usage_demo(change_set, "multi-use resale auth with expiry date")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 구매자에게 보낼 만료 날짜 및 EULA와 함께 다중 사용 재판매 권한 부여 게시

다음 코드 예제에서는 모든 제품 유형에 대한 만료 날짜가 포함된 다중 사용 재판매 권한 부여를 게시하고 구매자에게 보낼 사용자 지정 EULA를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
```

```

        "Type": "ResaleAuthorization@1.0"
    },
    "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    }
},
{
    "ChangeType": "ReleaseResaleAuthorization",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {}
},
{
    "ChangeType": "UpdateAvailability",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-05-31"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ResaleConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        }
                    }
                ]
            }
        ]
    }
}

```



```

    },
    "RateCard": [
      {
        "DimensionKey": "t2.small",
        "Price": "150"
      }
    ],
    "Constraints": {
      "MultipleDimensionSelection": "Allowed",
      "QuantityConfiguration": "Allowed"
    }
  }
]
},
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "BuyerLegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
          }
        ]
      }
    ]
  }
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2023-05-31"
      }
    }
  ]
}
```

```

    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [

```

```

        {
            "Type": "BuyerLegalTerm",
            "Documents": [
                {
                    "Type": "CustomEula",
                    "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                }
            ]
        }
    ]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a multi-use resale authorization with expiry date on my SaaS/AMI/
Container product and add custom EULA to be sent to the buyer
CAPI-56
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "multiuse resale auth with expiry date and custom
EULA")

```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

만료 날짜가 포함된 다중 사용 재판매 권한 부여 게시 및 AWSSDK를 사용하여 리셀러 계약 설명서 추가

다음 코드 예제에서는 모든 제품 유형의 만료 날짜가 포함된 다중 사용 재판매 권한 부여를 게시하고 ISV와 채널 파트너 간에 리셀러 계약 설명서를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
```

```

        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    },
    {
        "ChangeType": "ReleaseResaleAuthorization",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-05-31"
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "BuyerLegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",
                            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                        }
                    ]
                }
            ],
            "Type": "ResaleLegalTerm",
            "Documents": [
                {

```

```

        "Type": "CustomResellerContract",
        "Url": "https://s3.amazonaws.com/aws-mp-standard-
contracts/Standard-Contact-for-AWS-Marketplace-2022-07-14.pdf"}
    ]
  }
}
},
{
  "ChangeType": "UpdatePricingTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "PricingModel": "Contract",
    "Terms": [
      {
        "Type": "ResaleConfigurableUpfrontPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
          {
            "Selector": {
              "Type": "Duration",
              "Value": "P12M"
            },
            "RateCard": [
              {
                "DimensionKey": "t2.small",
                "Price": "150"
              }
            ],
            "Constraints": {
              "MultipleDimensionSelection": "Allowed",
              "QuantityConfiguration": "Allowed"
            }
          }
        ]
      }
    ]
  }
}
]

```

```
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",

```



```

        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2023-12-31"
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "BuyerLegalTerm",
                "Documents": [
                    {
                        "Type": "CustomEula",
                        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                    }
                ]
            },
            {
                "Type": "ResaleLegalTerm",
                "Documents": [
                    {
                        "Type": "CustomResellerContract",
                        "Url": "https://s3.amazonaws.com/aws-mp-standard-
contracts/Standard-Contact-for-AWS-Marketplace-2022-07-14.pdf"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",

```

```

        "Terms": [
            {
                "Type": "ResaleUsageBasedPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "RateCard": [
                            {
                                "DimensionKey": "t2.micro",
                                "Price": "150"
                            }
                        ]
                    }
                ]
            }
        ]
    }
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a multi-use resale authorization with expiry date on my SaaS/AMI/
Container product
and add reseller contract documentation between the ISV and channel partner
CAPI-57
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

```

```
def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "multi use resale auth with contract doc",
    )

    return response

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

만료와 함께 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.

다음 코드 예제에서는 모든 제품 유형의 만료 날짜가 포함된 다중 사용 재판매 권한 부여를 게시하고 재판매에 대한 특정 구매자 계정을 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-05-31"
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateBuyerTargetingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    }
  }
}

```

```

    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerTargetingTerm",
          "PositiveTargeting": {
            "BuyerAccounts": [
              "111111111111"
            ]
          }
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
          ]
        }
      ]
    }
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2023-05-31"
      }
    }
  ]
}
```

```

    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateBuyerTargetingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [

```



```

        {
            "Type": "BuyerTargetingTerm",
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111"
                ]
            }
        }
    ],
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "BuyerLegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",
                            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                        }
                    ]
                }
            ]
        }
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose

```

```

Publish multi-use resale authorization with expiry date for any product type
(AMI/SaaS/Container) and add specific buyer account for the resale
CAPI-82
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set,
        "Publish multi-use resale authorization with expiry date and add specific
        buyer account",
    )

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 만료 날짜 없이 다중 사용 재판매 권한 부여 게시

다음 코드 예제에서는 CP가 권한을 사용하여 CPPO를 생성할 수 있도록 AMI 제품에 대해 시간 단위 연간 요금이 적용되고 만료 날짜가 없는 다중 사용 재판매 권한 부여를 게시하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    },
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/custom-eula.pdf"
            }
          ]
        }
      ]
    }
  }
]

```

```

    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}

```

```

    },
    "DetailsDocument": {}
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {

```

```

        "Terms": [
            {
                "Type": "BuyerLegalTerm",
                "Documents": [
                    {
                        "Type": "CustomEula",
                        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                    }
                ]
            }
        ]
    }
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a multi-use resale authorization with no expiry date on my SaaS/AMI
product so my CP can use that to create Channel Partner Private Offer (CPP0)
CAPI-52
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "multi use resale auth with no expiry date")

```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 사용자 지정 EULA 추가

다음 코드 예제에서는 모든 제품 유형에 대한 만료 날짜 없이 다중 사용 재판매 권한부여를 게시하고 구매자에게 보낼 사용자 지정 EULA를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
```



```

        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    },
    {
        "ChangeType": "ReleaseResaleAuthorization",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ResaleConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",
                                "Value": "P12M"
                            },
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.small",
                                    "Price": "150"
                                }
                            ],
                            "Constraints": {
                                "MultipleDimensionSelection": "Allowed",
                                "QuantityConfiguration": "Allowed"
                            }
                        }
                    ]
                }
            ]
        }
    }
]

```

```

    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
          ]
        }
      ]
    }
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {

```

```

    "ChangeType": "CreateResaleAuthorization",
    "ChangeName": "ResaleAuthorization",
    "Entity": {
        "Type": "ResaleAuthorization@1.0"
    },
    "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    }
},
{
    "ChangeType": "ReleaseResaleAuthorization",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {}
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ResaleConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.small",
                                "Price": "150"
                            }
                        ]
                    }
                ]
            }
        ]
    }
},

```

```

        "Constraints": {
            "MultipleDimensionSelection": "Allowed",
            "QuantityConfiguration": "Allowed"
        }
    }
}
],
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "BuyerLegalTerm",
                "Documents": [
                    {
                        "Type": "CustomEula",
                        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                    }
                ]
            }
        ]
    }
}
]
}
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a multi-use resale authorization with no expiry date on my SaaS/AMI/
Container product and add custom EULA to be sent to the buyer

```

```

CAPI-58
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set, "multi use resale auth with no expiry date and custom EULA"
    )

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 리셀러 계약 문서 추가

다음 코드 예제에서는 모든 제품 유형에 대한 만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 ISV와 채널 파트너 간에 리셀러 계약 문서를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    },
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/custom-eula.pdf"
            }
          ]
        }
      ]
    }
  }
]

```

```

        }
      ],
    },
    {
      "Type": "ResaleLegalTerm",
      "Documents": [
        {
          "Type": "CustomResellerContract",
          "Url": "https://s3.amazonaws.com/aws-mp-standard-
contracts/Standard-Contract-for-AWS-Marketplace-2022-07-14.pdf"
        }
      ]
    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",

```



```

        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    },
    {
        "ChangeType": "ReleaseResaleAuthorization",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ResaleConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",
                                "Value": "P12M"
                            },
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.small",
                                    "Price": "150"
                                }
                            ]
                        }
                    ],
                    "Constraints": {
                        "MultipleDimensionSelection": "Allowed",
                        "QuantityConfiguration": "Allowed"
                    }
                }
            ]
        }
    }
}

```

```

        ]
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "BuyerLegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",
                            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                        }
                    ]
                }
            ],
            "Type": "ResaleLegalTerm",
            "Documents": [
                {
                    "Type": "CustomResellerContract",
                    "Url": "https://s3.amazonaws.com/aws-mp-standard-
contracts/Standard-Contact-for-AWS-Marketplace-2022-07-14.pdf"
                }
            ]
        }
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""

```

```

Purpose
Publish a multi-use resale authorization with no expiry date on my SaaS/AMI/
Container product and add reseller contract documentation between the ISV and
channel partner
CAPI-59
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set, "multi use resale auth with no expiry date and contract doc"
    )

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 AWSSDK를 사용하여 재판매에 대한 특정 구매자 계정을 추가합니다.

다음 코드 예제에서는 모든 제품 유형의 만료 날짜 없이 다중 사용 재판매 권한 부여를 게시하고 재판매에 대한 특정 구매자 계정을 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateBuyerTargetingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerTargetingTerm",
          "PositiveTargeting": {
            "BuyerAccounts": [
              "111111111111"
            ]
          }
        }
      ]
    }
  }
]

```

```

        }
    ]
}
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "BuyerLegalTerm",
                "Documents": [
                    {
                        "Type": "CustomEula",
                        "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                    }
                ]
            }
        ]
    }
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
    "Catalog": "AWSMarketplace",

```

```

"ChangeSet": [
  {
    "ChangeType": "CreateResaleAuthorization",
    "ChangeName": "ResaleAuthorization",
    "Entity": {
      "Type": "ResaleAuthorization@1.0"
    },
    "DetailsDocument": {
      "ProductId": "prod-111111111111",
      "Name": "TestResaleAuthorization",
      "Description": "Worldwide ResaleAuthorization for Test Product",
      "ResellerAccountId": "111111111111"
    }
  },
  {
    "ChangeType": "ReleaseResaleAuthorization",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {}
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ]
            }
          ]
        }
      ]
    }
  }
]

```

```

        }
    ],
    "Constraints": {
        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
    }
}
]
}
]
}
},
{
    "ChangeType": "UpdateBuyerTargetingTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "BuyerTargetingTerm",
                "PositiveTargeting": {
                    "BuyerAccounts": [
                        "111111111111"
                    ]
                }
            }
        ]
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "BuyerLegalTerm",
                "Documents": [
                    {
                        "Type": "CustomEula",

```



```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 일회성 재판매 권한 부여 게시 및 유연한 결제 일정 추가

다음 코드 예제에서는 모든 제품 유형에 대한 일회성 재판매 권한 부여를 게시하고 유연한 결제 일정을 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    }
  ]
}
```

```

    }
  },
  {
    "ChangeType": "ReleaseResaleAuthorization",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {}
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleFixedUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "Price": "0.00",
          "Duration": "P12M",
          "Grants": [
            {
              "DimensionKey": "Users",
              "MaxQuantity": 10
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdatePaymentScheduleTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "ResalePaymentScheduleTerm",

```

```

        "CurrencyCode": "USD",
        "Schedule": [
            {
                "ChargeDate": "2023-09-01",
                "ChargeAmount": "200.00"
            },
            {
                "ChargeDate": "2023-12-01",
                "ChargeAmount": "250.00"
            }
        ]
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "AvailabilityEndDate": "2023-06-30",
            "OffersMaxQuantity": 1
        }
    },
    {
        "ChangeType": "UpdateLegalTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "BuyerLegalTerm",
                    "Documents": [
                        {
                            "Type": "CustomEula",
                            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                        }
                    ]
                }
            ]
        }
    }
}

```

```

    ]
  }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    }
  ],
}

```

```

{
  "ChangeType": "UpdatePricingTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "PricingModel": "Contract",
    "Terms": [
      {
        "Type": "ResaleFixedUpfrontPricingTerm",
        "CurrencyCode": "USD",
        "Price": "0.00",
        "Duration": "P12M",
        "Grants": [
          {
            "DimensionKey": "Users",
            "MaxQuantity": 10
          }
        ]
      }
    ]
  }
},
{
  "ChangeType": "UpdatePaymentScheduleTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "ResalePaymentScheduleTerm",
        "CurrencyCode": "USD",
        "Schedule": [
          {
            "ChargeDate": "2023-09-01",
            "ChargeAmount": "200.00"
          },
          {
            "ChargeDate": "2023-12-01",
            "ChargeAmount": "250.00"
          }
        ]
      }
    ]
  }
}

```

```

        ]
      }
    ]
  },
  {
    "ChangeType": "UpdateAvailability",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "AvailabilityEndDate": "2023-06-30",
      "OffersMaxQuantity": 1
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
          ]
        }
      ]
    }
  }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish one-time resale authorization for any product type (AMI/SaaS/Container)
and add Flexible payment schedule
CAPI-78
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "onetime resale auth with flexible payment
    schedule")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 EULA 추가

다음 코드 예제에서는 모든 제품 유형에 대한 일회성 재판매 권한 부여를 게시하고 구매자에게 보낼 사용자 지정 EULA를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
      "OffersMaxQuantity": 1
    }
  },
  {
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ]
            },
            {
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    }
  }
}

```

```

    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
          ]
        }
      ]
    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",

```

```

        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    },
    {
        "ChangeType": "ReleaseResaleAuthorization",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "OffersMaxQuantity": 1
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ResaleConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",
                                "Value": "P12M"
                            },
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.small",

```

```

        "Price": "150"
      }
    ],
    "Constraints": {
      "MultipleDimensionSelection": "Allowed",
      "QuantityConfiguration": "Allowed"
    }
  }
]
},
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "BuyerLegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
          }
        ]
      }
    ]
  }
}
]
}
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""

```

```

Purpose
Publish a one-time resale authorization on my SaaS/AMI/Container product and add
custom EULA to be sent to the buyer
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "onetime resale auth with custom EULA")

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

SDK를 사용하여 일회성 재판매 권한 부여를 게시하고 재판매에 대한 특정 구매자 계정을 추가합니다.AWS

다음 코드 예제에서는 모든 제품 유형에 대한 일회성 재판매 권한 부여를 게시하고 재판매에 대한 특정 구매자 계정을 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
      "PricingModel": "Contract",
      "Terms": [
        {
          "Type": "ResaleConfigurableUpfrontPricingTerm",
          "CurrencyCode": "USD",
          "RateCards": [
            {
              "Selector": {
                "Type": "Duration",
                "Value": "P12M"
              },
              "RateCard": [
                {
                  "DimensionKey": "t2.small",
                  "Price": "150"
                }
              ],
              "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
              }
            }
          ]
        }
      ]
    },
  },
  {
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
      "Type": "ResaleAuthorization@1.0",
      "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/custom-eula.pdf"
            }
          ]
        }
      ]
    }
  }
]

```



```

    }
  ]
}
},
{
  "ChangeType": "UpdateAvailability",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "OffersMaxQuantity": "1"
  }
},
{
  "ChangeType": "UpdateBuyerTargetingTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "BuyerTargetingTerm",
        "PositiveTargeting": {
          "BuyerAccounts": [
            "111111111111"
          ]
        }
      }
    ]
  }
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {
        "PricingModel": "Contract",

```

```

        "Terms": [
            {
                "Type": "ResaleConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.small",
                                "Price": "150"
                            }
                        ],
                        "Constraints": {
                            "MultipleDimensionSelection": "Allowed",
                            "QuantityConfiguration": "Allowed"
                        }
                    }
                ]
            }
        ],
        {
            "ChangeType": "UpdateLegalTerms",
            "Entity": {
                "Type": "ResaleAuthorization@1.0",
                "Identifier": "$ResaleAuthorization.Entity.Identifier"
            },
            "DetailsDocument": {
                "Terms": [
                    {
                        "Type": "BuyerLegalTerm",
                        "Documents": [
                            {
                                "Type": "CustomEula",
                                "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
                            }
                        ]
                    }
                ]
            }
        }
    ]
}

```

```

        ]
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "OffersMaxQuantity": 1
        }
    },
    {
        "ChangeType": "UpdateBuyerTargetingTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "BuyerTargetingTerm",
                    "PositiveTargeting": {
                        "BuyerAccounts": [
                            "111111111111"
                        ]
                    }
                }
            ]
        }
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose

```

```

Publish one-time resale authorization for any product type (AMI/SaaS/Container)
and add specific buyer account for the resale
CAPI-81
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "onetime resale authorization for specific buyer
    account")

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 모든 제품 유형에 대한 일회성 재판매 권한 부여 게시 및 리셀러 계약 문서 추가

다음 코드 예제에서는 모든 제품 유형에 대한 일회성 재판매 권한 부여를 게시하고 ISV와 채널 파트너 간의 리셀러 계약 설명서를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      }
    }
  ]
}
```

```

    },
    "DetailsDocument": {
        "OffersMaxQuantity": 1
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ResaleConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.small",
                                "Price": "150"
                            }
                        ]
                    },
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P12M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "t2.xlarge",
                                "Price": "150"
                            }
                        ]
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
    }
}

```

```

    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "BuyerLegalTerm",
          "Documents": [
            {
              "Type": "CustomEula",
              "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
            }
          ]
        }
      ]
    }
  ]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",

```



```

        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
    },
    {
        "ChangeType": "ReleaseResaleAuthorization",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "OffersMaxQuantity": 1
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ResaleConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",
                                "Value": "P12M"
                            },
                            "RateCard": [
                                {
                                    "DimensionKey": "t2.small",

```

```

        "Price": "150"
      }
    ],
    "Constraints": {
      "MultipleDimensionSelection": "Allowed",
      "QuantityConfiguration": "Allowed"
    }
  }
]
}
],
{
  "ChangeType": "UpdateLegalTerms",
  "Entity": {
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "$ResaleAuthorization.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "BuyerLegalTerm",
        "Documents": [
          {
            "Type": "CustomEula",
            "Url": "https://s3.amazonaws.com/sample-bucket/
custom-eula.pdf"
          }
        ]
      },
      {
        "Type": "ResaleLegalTerm",
        "Documents": [
          {
            "Type": "CustomResellerContract",
            "Url": "https://s3.amazonaws.com/aws-mp-standard-
contracts/Standard-Contact-for-AWS-Marketplace-2022-07-14.pdf"
          }
        ]
      }
    ]
  }
}
}

```

```
]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish a one-time resale authorization on my SaaS/AMI/Container product and add
reseller contract documentation between the ISV and channel partner
CAPI-47
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "onetime resale auth with reseller contract doc")

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

에 대한 일회성 재판매 권한 부여 게시 및 AWSSDK를 사용한 갱신 여부 추가

다음 코드 예제에서는 모든 제품 유형에 대한 일회성 재판매 권한 부여를 게시하고 갱신 여부를 추가하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 `changeset`를 시작하기 위한 유틸리티에서 다음 JSON `changeset`를 `RunChangesets`에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "UpdateBuyerTargetingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
```

```

        {
            "Type": "BuyerTargetingTerm",
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "222222222222"
                ]
            }
        }
    ],
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "OffersMaxQuantity": 1
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "TestResaleAuthorization",
            "Description": "Worldwide ResaleAuthorization for Test Product",
            "PreExistingBuyerAgreement": {
                "AcquisitionChannel": "AwsMarketplace",
                "PricingModel": "Contract"
            }
        }
    }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateResaleAuthorization",
      "ChangeName": "ResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product",
        "ResellerAccountId": "111111111111"
      }
    },
    {
      "ChangeType": "UpdateBuyerTargetingTerms",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "$ResaleAuthorization.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "BuyerTargetingTerm",
            "PositiveTargeting": {
              "BuyerAccounts": [
                "222222222222"
              ]
            }
          }
        ]
      }
    }
  ]
}
```

```

        ]
    },
    {
        "ChangeType": "UpdateAvailability",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "OffersMaxQuantity": 1
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "ResaleAuthorization@1.0",
            "Identifier": "$ResaleAuthorization.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "TestResaleAuthorization",
            "Description": "Worldwide ResaleAuthorization for Test Product",
            "PreExistingBuyerAgreement": {
                "AcquisitionChannel": "AwsMarketplace",
                "PricingModel": "Contract"
            }
        }
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish one-time resale authorization for any product type (AMI/SaaS/Container)
and add whether it is renewal or not
CAPI-90
"""

```

```

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(stringified_change_set, "onetime resale auth
renewal")

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#)[AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 재판매 권한 부여 제한

다음 코드 예제에서는 재판매 권한 부여를 제한하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-111111111111"
      },
      "DetailsDocument": {}
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
```

```

    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "RestrictResaleAuthorization",
            "Entity": {
                "Type": "ResaleAuthorization@1.0",
                "Identifier": "resaleauthz-111111111111"
            },
            "DetailsDocument": {}
        }
    ]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Restrict a  authorization for any product type (AMI/SaaS/Container)
CAPI-84
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "Restrict resale authorization")

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 일회성 또는 다목적 재판매 권한 부여의 이름 및 설명 업데이트

다음 코드 예제에서는 모든 제품 유형에 대해 권한 부여를 게시하기 전에 일회성 또는 다중 사용 재판매 권한 부여의 이름과 설명을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-111111111111"
      },
      "DetailsDocument": {
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product"
      }
    }
  ]
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-111111111111"
      },
      "DetailsDocument": {
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Update name/description of one-time or multi-use resale authorization before
publishing for any product type (AMI/SaaS/Container)
CAPI-77
"""

import os

import utils.start_changeset as sc
```

```
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set,
        "update name and description of one-time or multi-use resale
        authorization before publishing",
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API용 SaaS 제품

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 공개 제안 초안을 사용하여 SaaS 제품 초안 생성](#)
- [AWSSDK를 사용하여 계약 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 Pay-As-You-Go 요금 계약으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 구독 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성](#)
- [AWSSDK를 사용하여 SaaS 제품 및 관련 공개 제안 게시](#)
- [AWSSDK를 사용하여 기존 초안에서 SaaS 제품 및 관련 공개 제안 게시](#)
- [AWSSDK를 사용하여 AMI 또는 SaaS 제품의 차원 업데이트](#)

AWSSDK를 사용하여 공개 제안 초안을 사용하여 SaaS 제품 초안 생성

다음 코드 예제에서는 공개 제안 초안을 사용하여 SaaS 제품 초안을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product"
      }
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier",
        "Name": "Test Offer"
      }
    }
  ]
}
```

```
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "CreateOffer",
      "ChangeName": "CreateOfferChange",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier",
        "Name": "Test Offer"
      }
    }
  ]
}
```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create an SaaS draft product
with a draft public offer.
CAPI-04
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

    response = sc.usage_demo(
        stringified_change_set,
        "Create a draft saas product with a draft public offer",
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성

다음 코드 예제에서는 공개 또는 제한된 SaaS 제품과 계약 요금이 적용되는 공개 제안을 생성하는 방법을 보여줍니다. 이 예제에서는 표준 또는 사용자 지정 EULA를 생성합니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "ChangeName": "CreateProductChange",
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ]
      }
    }
  ]
}
```

```

        "SearchKeywords": [
            "Sample keyword"
        ],
        "Categories": [
            "Data Catalogs"
        ],
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [
            "https://sample.amazonaws.com/awsmvp-video-1"
        ],
        "AdditionalResources": []
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "AddDeliveryOptions",
        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "DeliveryOptions": [
                {
                    "Details": {
                        "SaaSUrlDeliveryOptionDetails": {
                            "FulfillmentUrl": "https://sample.amazonaws.com/sample-saas-fulfillment-url"
                        }
                    }
                }
            ]
        }
    }
]

```

```

    ]
  },
  {
    "ChangeType": "AddDimensions",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
      {
        "Key": "BasicService",
        "Description": "Basic Service",
        "Name": "Basic Service",
        "Types": [
          "Entitled"
        ],
        "Unit": "Units"
      },
      {
        "Key": "PremiumService",
        "Description": "Premium Service",
        "Name": "Premium Service",
        "Types": [
          "Entitled"
        ],
        "Unit": "Units"
      }
    ]
  },
  {
    "ChangeType": "ReleaseProduct",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  },
  {
    "ChangeType": "CreateOffer",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "ChangeName": "CreateOfferChange",

```

```

        "DetailsDocument": {
            "ProductId": "$CreateProductChange.Entity.Identifier"
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test public offer for SaaSProduct using AWS Marketplace
API Reference Code",
            "Description": "Test public offer with contract pricing for
SaaSProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",
                                "Value": "P1M"
                            },
                            "RateCard": [
                                {
                                    "DimensionKey": "BasicService",
                                    "Price": "20"
                                },
                                {
                                    "DimensionKey": "PremiumService",
                                    "Price": "25"
                                }
                            ]
                        }
                    ]
                }
            ]
        }
    }
}

```

```

        ],
        "Constraints": {
            "MultipleDimensionSelection": "Allowed",
            "QuantityConfiguration": "Allowed"
        }
    },
    {
        "Selector": {
            "Type": "Duration",
            "Value": "P12M"
        },
        "RateCard": [
            {
                "DimensionKey": "BasicService",
                "Price": "150"
            },
            {
                "DimensionKey": "PremiumService",
                "Price": "300"
            }
        ],
        "Constraints": {
            "MultipleDimensionSelection": "Allowed",
            "QuantityConfiguration": "Allowed"
        }
    }
]
}
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",

```

```

        "Version": "2022-07-14"
    }
}
]
}
},
{
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "SupportTerm",
                "RefundPolicy": "Absolutely no refund, period."
            }
        ]
    }
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "ChangeName": "CreateProductChange",
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [
          "Data Catalogs"
        ],
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [
```

```

        "https://sample.amazonaws.com/awsmvp-video-1"
    ],
    "AdditionalResources": []
  }
},
{
  "ChangeType": "UpdateTargeting",
  "Entity": {
    "Type": "SaaSProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "PositiveTargeting": {
      "BuyerAccounts": [
        "111111111111",
        "222222222222"
      ]
    }
  }
},
{
  "ChangeType": "AddDeliveryOptions",
  "Entity": {
    "Type": "SaaSProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "DeliveryOptions": [
      {
        "Details": {
          "SaaSUrlDeliveryOptionDetails": {
            "FulfillmentUrl": "https://sample.amazonaws.com/
sample-saas-fulfillment-url"
          }
        }
      ]
    }
  }
},
{
  "ChangeType": "AddDimensions",
  "Entity": {
    "Type": "SaaSProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  }
}

```



```

    },
    "DetailsDocument": [
      {
        "Key": "BasicService",
        "Description": "Basic Service",
        "Name": "Basic Service",
        "Types": [
          "Entitled"
        ],
        "Unit": "Units"
      },
      {
        "Key": "PremiumService",
        "Description": "Premium Service",
        "Name": "Premium Service",
        "Types": [
          "Entitled"
        ],
        "Unit": "Units"
      }
    ]
  },
  {
    "ChangeType": "ReleaseProduct",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  },
  {
    "ChangeType": "CreateOffer",
    "Entity": {
      "Type": "Offer@1.0"
    },
    "ChangeName": "CreateOfferChange",
    "DetailsDocument": {
      "ProductId": "$CreateProductChange.Entity.Identifier"
    }
  },
  {
    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "Offer@1.0",

```

```

        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Name": "Test public offer for SaaSProduct using AWS Marketplace
API Reference Code",
        "Description": "Test public offer with contract pricing for
SaaSProduct using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Contract",
        "Terms": [
            {
                "Type": "ConfigurableUpfrontPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "Selector": {
                            "Type": "Duration",
                            "Value": "P1M"
                        },
                        "RateCard": [
                            {
                                "DimensionKey": "BasicService",
                                "Price": "20"
                            },
                            {
                                "DimensionKey": "PremiumService",
                                "Price": "25"
                            }
                        ]
                    },
                    {
                        "Constraints": {
                            "MultipleDimensionSelection": "Allowed",
                            "QuantityConfiguration": "Allowed"
                        }
                    }
                ],
                "Selector": {

```

```

        "Type": "Duration",
        "Value": "P12M"
    },
    "RateCard": [
        {
            "DimensionKey": "BasicService",
            "Price": "150"
        },
        {
            "DimensionKey": "PremiumService",
            "Price": "300"
        }
    ],
    "Constraints": {
        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
    }
}

]
}

],
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{

```

```

        "ChangeType": "UpdateSupportTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "SupportTerm",
                    "RefundPolicy": "Absolutely no refund, period."
                }
            ]
        }
    },
    {
        "ChangeType": "UpdateRenewalTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "RenewalTerm"
                }
            ]
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
```

```
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a
public or limited SaaS product and public offer with contract pricing and
standard EULA
CAPI-11
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set,
        "Create a limited saas product with a public offer with contract
pricing",
    )

if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 Pay-As-You-Go 요금 계약으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성

다음 코드 예제에서는 공개 또는 제한된 SaaS 제품 및 종량제 요금으로 계약이 적용되는 공개 제안을 생성하는 방법을 보여줍니다. 이 예제에서는 표준 또는 사용자 지정 EULA를 생성합니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "ChangeName": "CreateProductChange",
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [
          "Data Catalogs"
        ]
      }
    }
  ]
}
```

```

        ],
        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [
            "https://sample.amazonaws.com/awsmvp-video-1"
        ],
        "AdditionalResources": []
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "AddDeliveryOptions",
        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "DeliveryOptions": [
                {
                    "Details": {
                        "SaaSUrlDeliveryOptionDetails": {
                            "FulfillmentUrl": "https://sample.amazonaws.com/sample-saas-fulfillment-url"
                        }
                    }
                }
            ]
        }
    },
    {
        "ChangeType": "AddDimensions",

```

```

    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
      {
        "Key": "BasicService",
        "Description": "Basic Service",
        "Name": "Basic Service",
        "Types": [
          "Entitled"
        ],
        "Unit": "Units"
      },
      {
        "Key": "PremiumService",
        "Description": "Premium Service",
        "Name": "Premium Service",
        "Types": [
          "Entitled"
        ],
        "Unit": "Units"
      },
      {
        "Key": "WorkloadSmall",
        "Description": "Workload: Per medium instance",
        "Name": "Workload: Per medium instance",
        "Types": [
          "ExternallyMetered"
        ],
        "Unit": "Units"
      },
      {
        "Key": "WorkloadMedium",
        "Description": "Workload: Per large instance",
        "Name": "Workload: Per large instance",
        "Types": [
          "ExternallyMetered"
        ],
        "Unit": "Units"
      }
    ]
  },
  {

```



```

        "ChangeType": "ReleaseProduct",
        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "CreateOffer",
        "Entity": {
            "Type": "Offer@1.0"
        },
        "ChangeName": "CreateOfferChange",
        "DetailsDocument": {
            "ProductId": "$CreateProductChange.Entity.Identifier"
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test public offer for SaaSProduct using AWS Marketplace
API Reference Code",
            "Description": "Test public offer with contract pricing for
SaaSProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [

```

```

        "RateCard": [
            {
                "DimensionKey": "WorkloadSmall",
                "Price": "0.15"
            },
            {
                "DimensionKey": "WorkloadMedium",
                "Price": "0.25"
            }
        ]
    },
    ],
    },
    {
        "Type": "ConfigurableUpfrontPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
            {
                "Selector": {
                    "Type": "Duration",
                    "Value": "P12M"
                },
                "RateCard": [
                    {
                        "DimensionKey": "BasicService",
                        "Price": "150"
                    },
                    {
                        "DimensionKey": "PremiumService",
                        "Price": "300"
                    }
                ],
                "Constraints": {
                    "MultipleDimensionSelection": "Allowed",
                    "QuantityConfiguration": "Allowed"
                }
            }
        ]
    }
    ],
    },
    {
        "ChangeType": "UpdateLegalTerms",

```

```

    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "LegalTerm",
          "Documents": [
            {
              "Type": "StandardEula",
              "Version": "2022-07-14"
            }
          ]
        }
      ]
    }
  },
  {
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "Terms": [
        {
          "Type": "SupportTerm",
          "RefundPolicy": "Absolutely no refund, period."
        }
      ]
    }
  },
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
  }
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "ChangeName": "CreateProductChange",
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [
          "Data Catalogs"
        ],
      },
    },
  ],
}
```

```

        "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
        "VideoUrls": [
            "https://sample.amazonaws.com/awsmvp-video-1"
        ],
        "AdditionalResources": []
    },
    {
        "ChangeType": "UpdateTargeting",
        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PositiveTargeting": {
                "BuyerAccounts": [
                    "111111111111",
                    "222222222222"
                ]
            }
        }
    },
    {
        "ChangeType": "AddDeliveryOptions",
        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "DeliveryOptions": [
                {
                    "Details": {
                        "SaaSUrlDeliveryOptionDetails": {
                            "FulfillmentUrl": "https://sample.amazonaws.com/sample-saas-fulfillment-url"
                        }
                    }
                ]
            }
        }
    },
    {
        "ChangeType": "AddDimensions",
        "Entity": {

```

```

        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
        {
            "Key": "BasicService",
            "Description": "Basic Service",
            "Name": "Basic Service",
            "Types": [
                "Entitled"
            ],
            "Unit": "Units"
        },
        {
            "Key": "PremiumService",
            "Description": "Premium Service",
            "Name": "Premium Service",
            "Types": [
                "Entitled"
            ],
            "Unit": "Units"
        },
        {
            "Key": "WorkloadSmall",
            "Description": "Workload: Per medium instance",
            "Name": "Workload: Per medium instance",
            "Types": [
                "ExternallyMetered"
            ],
            "Unit": "Units"
        },
        {
            "Key": "WorkloadMedium",
            "Description": "Workload: Per large instance",
            "Name": "Workload: Per large instance",
            "Types": [
                "ExternallyMetered"
            ],
            "Unit": "Units"
        }
    ]
},
{
    "ChangeType": "ReleaseProduct",

```

```

        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "CreateOffer",
        "Entity": {
            "Type": "Offer@1.0"
        },
        "ChangeName": "CreateOfferChange",
        "DetailsDocument": {
            "ProductId": "$CreateProductChange.Entity.Identifier"
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test public offer for SaaSProduct using AWS Marketplace
API Reference Code",
            "Description": "Test public offer with contract pricing for
SaaSProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [

```

```

        {
            "DimensionKey": "WorkloadSmall",
            "Price": "0.15"
        },
        {
            "DimensionKey": "WorkloadMedium",
            "Price": "0.25"
        }
    ]
}
],
},
{
    "Type": "ConfigurableUpfrontPricingTerm",
    "CurrencyCode": "USD",
    "RateCards": [
        {
            "Selector": {
                "Type": "Duration",
                "Value": "P12M"
            },
            "RateCard": [
                {
                    "DimensionKey": "BasicService",
                    "Price": "150"
                },
                {
                    "DimensionKey": "PremiumService",
                    "Price": "300"
                }
            ],
            "Constraints": {
                "MultipleDimensionSelection": "Allowed",
                "QuantityConfiguration": "Allowed"
            }
        }
    ]
}
]
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {

```



```

        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "SupportTerm",
                "RefundPolicy": "Absolutely no refund, period."
            }
        ]
    }
},
{
    "ChangeType": "UpdateRenewalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "RenewalTerm"
            }
        ]
    }
}

```

```

        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a
public or limited SaaS product and public offer with contract with PAYG pricing
and standard EULA
CAPI-10
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(
        change_set,
        "Create limited SaaS product with public offer with contract with payg
pricing",
    )

```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 구독 요금으로 공개 또는 제한된 SaaS 제품 및 공개 제안 생성

다음 코드 예제에서는 공개 또는 제한된 SaaS 제품과 구독 요금이 적용되는 공개 제안을 생성하는 방법을 보여줍니다. 이 예제에서는 표준 또는 사용자 지정 EULA를 생성합니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "ChangeName": "CreateProductChange",
      "DetailsDocument": {}
    },
    {
```

```

    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "ProductTitle": "Sample product",
      "ShortDescription": "Brief description",
      "LongDescription": "Detailed description",
      "Highlights": [
        "Sample highlight"
      ],
      "SearchKeywords": [
        "Sample keyword"
      ],
      "Categories": [
        "Data Catalogs"
      ],
      "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
      "VideoUrls": [
        "https://sample.amazonaws.com/awssmp-video-1"
      ],
      "AdditionalResources": []
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111",
          "222222222222"
        ]
      }
    }
  },
  {
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
      "Type": "SaaSProduct@1.0",

```

```

        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "DeliveryOptions": [
            {
                "Details": {
                    "SaaSUrlDeliveryOptionDetails": {
                        "FulfillmentUrl": "https://sample.amazonaws.com/sample-saas-fulfillment-url"
                    }
                }
            }
        ]
    }
},
{
    "ChangeType": "AddDimensions",
    "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
        {
            "Key": "WorkloadSmall",
            "Description": "Workload: Per medium instance",
            "Name": "Workload: Per medium instance",
            "Types": [
                "ExternallyMetered"
            ],
            "Unit": "Units"
        },
        {
            "Key": "WorkloadMedium",
            "Description": "Workload: Per large instance",
            "Name": "Workload: Per large instance",
            "Types": [
                "ExternallyMetered"
            ],
            "Unit": "Units"
        }
    ]
},
{
    "ChangeType": "ReleaseProduct",

```

```

        "Entity": {
            "Type": "SaaSProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "CreateOffer",
        "Entity": {
            "Type": "Offer@1.0"
        },
        "ChangeName": "CreateOfferChange",
        "DetailsDocument": {
            "ProductId": "$CreateProductChange.Entity.Identifier"
        }
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Name": "Test public offer for SaaSProduct using AWS Marketplace
API Reference Code",
            "Description": "Test public offer with contract pricing for
SaaSProduct using AWS Marketplace API Reference Code"
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Usage",
            "Terms": [
                {
                    "Type": "UsageBasedPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "RateCard": [

```

```

        {
            "DimensionKey": "WorkloadSmall",
            "Price": "0.15"
        },
        {
            "DimensionKey": "WorkloadMedium",
            "Price": "0.25"
        }
    ]
}

]
}

]
}

],
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {

```

```

        "Type": "SupportTerm",
        "RefundPolicy": "Absolutely no refund, period."
    }
]
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "CreateProduct",
            "Entity": {
                "Type": "SaaSProduct@1.0"
            },
            "ChangeName": "CreateProductChange",
            "DetailsDocument": {}
        },
        {
            "ChangeType": "UpdateInformation",

```



```

    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "ProductTitle": "Sample product",
      "ShortDescription": "Brief description",
      "LongDescription": "Detailed description",
      "Highlights": [
        "Sample highlight"
      ],
      "SearchKeywords": [
        "Sample keyword"
      ],
      "Categories": [
        "Data Catalogs"
      ],
      "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
      "VideoUrls": [
        "https://sample.amazonaws.com/awsmp-video-1"
      ],
      "AdditionalResources": []
    }
  },
  {
    "ChangeType": "UpdateTargeting",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "PositiveTargeting": {
        "BuyerAccounts": [
          "111111111111",
          "222222222222"
        ]
      }
    }
  }
],
{
  "ChangeType": "AddDeliveryOptions",
  "Entity": {
    "Type": "SaaSProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  }
}

```

```

    },
    "DetailsDocument": {
      "DeliveryOptions": [
        {
          "Details": {
            "SaaSUrlDeliveryOptionDetails": {
              "FulfillmentUrl": "https://sample.amazonaws.com/sample-saas-fulfillment-url"
            }
          }
        }
      ]
    }
  },
  {
    "ChangeType": "AddDimensions",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
      {
        "Key": "WorkloadSmall",
        "Description": "Workload: Per medium instance",
        "Name": "Workload: Per medium instance",
        "Types": [
          "ExternallyMetered"
        ],
        "Unit": "Units"
      },
      {
        "Key": "WorkloadMedium",
        "Description": "Workload: Per large instance",
        "Name": "Workload: Per large instance",
        "Types": [
          "ExternallyMetered"
        ],
        "Unit": "Units"
      }
    ]
  },
  {
    "ChangeType": "ReleaseProduct",
    "Entity": {

```

```

        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
},
{
    "ChangeType": "CreateOffer",
    "Entity": {
        "Type": "Offer@1.0"
    },
    "ChangeName": "CreateOfferChange",
    "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier"
    }
},
{
    "ChangeType": "UpdateInformation",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Name": "Test public offer for SaaSProduct using AWS Marketplace
API Reference Code",
        "Description": "Test public offer with contract pricing for
SaaSProduct using AWS Marketplace API Reference Code"
    }
},
{
    "ChangeType": "UpdatePricingTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
            {
                "Type": "UsageBasedPricingTerm",
                "CurrencyCode": "USD",
                "RateCards": [
                    {
                        "RateCard": [

```

```

        "DimensionKey": "WorkloadSmall",
        "Price": "0.15"
    },
    {
        "DimensionKey": "WorkloadMedium",
        "Price": "0.25"
    }
]
}
]
}
],
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    }
},
{
    "ChangeType": "UpdateSupportTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Terms": [
            {
                "Type": "SupportTerm",

```

```

        "RefundPolicy": "Absolutely no refund, period."
    }
    ]
},
{
    "ChangeType": "ReleaseOffer",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {}
}
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to create a
public or limited SaaS product and public offer with subscription(usage) pricing
and standard EULA
CAPI-09
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

fname = "changeset.json"
change_set_file = os.path.join(os.path.dirname(__file__), fname)

change_set = sd.stringify_changeset(change_set_file)

def main():
    sc.usage_demo(change_set, "public saas public offer with subscription
pricing")

```

```
if __name__ == "__main__":
    main()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 제품 및 관련 공개 제안 게시

다음 코드 예제에서는 SaaS 제품 및 관련된 공개 제안을 게시하는 방법을 보여줍니다. 제품은 기본적으로 제한된 상태입니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
```

```

    "ChangeType": "UpdateInformation",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
      "ProductTitle": "Sample product",
      "ShortDescription": "Brief description",
      "LongDescription": "Detailed description",
      "Highlights": [
        "Sample highlight"
      ],
      "SearchKeywords": [
        "Sample keyword"
      ],
      "Categories": [
        "Data Catalogs"
      ],
      "LogoUrl": "https://bucketname.s3.amazonaws.com/logo.png",
      "VideoUrls": [
        "https://sample.amazonaws.com/awsmvp-video-1"
      ],
      "AdditionalResources": []
    }
  },
  {
    "ChangeType": "AddDimensions",
    "Entity": {
      "Type": "SaaSProduct@1.0",
      "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": [
      {
        "Key": "BasicService",
        "Description": "Basic Service",
        "Name": "Basic Service",
        "Types": [
          "Entitled"
        ],
        "Unit": "Units"
      },
      {
        "Key": "PremiumService",
        "Description": "Premium Service",

```

```

        "Name": "Premium Service",
        "Types": [
            "Entitled"
        ],
        "Unit": "Units"
    }
]
},
{
    "ChangeType": "AddDeliveryOptions",
    "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "DeliveryOptions": [
            {
                "Details": {
                    "SaaSUrlDeliveryOptionDetails": {
                        "FulfillmentUrl": "https://www.aws.amazon.com/
marketplace/management"
                    }
                }
            }
        ]
    }
},
{
    "ChangeType": "ReleaseProduct",
    "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
},
{
    "ChangeType": "CreateOffer",
    "ChangeName": "CreateOfferChange",
    "Entity": {
        "Type": "Offer@1.0"
    },
    "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier"
    }
}

```



```

    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Name": "New Test Offer",
        "Description": "New offer description"
      }
    },
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {
                "Type": "StandardEula",
                "Version": "2022-07-14"
              }
            ]
          }
        ]
      }
    },
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "SupportTerm",
            "RefundPolicy": "Updated refund policy description"
          }
        ]
      }
    }
  ]
}

```

```

    ]
  }
},
{
  "ChangeType": "UpdatePricingTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "PricingModel": "Contract",
    "Terms": [
      {
        "Type": "ConfigurableUpfrontPricingTerm",
        "CurrencyCode": "USD",
        "RateCards": [
          {
            "Selector": {
              "Type": "Duration",
              "Value": "P1M"
            },
            "RateCard": [
              {
                "DimensionKey": "BasicService",
                "Price": "20"
              },
              {
                "DimensionKey": "PremiumService",
                "Price": "25"
              }
            ],
            "Constraints": {
              "MultipleDimensionSelection": "Allowed",
              "QuantityConfiguration": "Allowed"
            }
          },
          {
            "Selector": {
              "Type": "Duration",
              "Value": "P12M"
            },
            "RateCard": [
              {
                "DimensionKey": "BasicService",

```

```

        "Price": "150"
      },
      {
        "DimensionKey": "PremiumService",
        "Price": "300"
      }
    ],
    "Constraints": {
      "MultipleDimensionSelection": "Allowed",
      "QuantityConfiguration": "Allowed"
    }
  }
]
}
},
{
  "ChangeType": "UpdateRenewalTerms",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "Terms": [
      {
        "Type": "RenewalTerm"
      }
    ]
  }
},
{
  "ChangeType": "ReleaseOffer",
  "Entity": {
    "Type": "Offer@1.0",
    "Identifier": "$CreateOfferChange.Entity.Identifier"
  },
  "DetailsDocument": {}
}
]
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "SaaSProduct@1.0"
      },
      "DetailsDocument": {}
    },
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
      },
      "DetailsDocument": {
        "ProductTitle": "Sample product",
        "ShortDescription": "Brief description",
        "LongDescription": "Detailed description",
        "Highlights": [
          "Sample highlight"
        ],
        "SearchKeywords": [
          "Sample keyword"
        ],
        "Categories": [
          "Data Catalogs"
        ],
        "LogoUrl": "https://bucketname.s3.amazonaws.com/logo.png",
        "VideoUrls": [
```

```

        "https://sample.amazonaws.com/awsmvp-video-1"
    ],
    "AdditionalResources": []
  }
},
{
  "ChangeType": "AddDimensions",
  "Entity": {
    "Type": "SaaSProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": [
    {
      "Key": "BasicService",
      "Description": "Basic Service",
      "Name": "Basic Service",
      "Types": [
        "Entitled"
      ],
      "Unit": "Units"
    },
    {
      "Key": "PremiumService",
      "Description": "Premium Service",
      "Name": "Premium Service",
      "Types": [
        "Entitled"
      ],
      "Unit": "Units"
    }
  ]
},
{
  "ChangeType": "AddDeliveryOptions",
  "Entity": {
    "Type": "SaaSProduct@1.0",
    "Identifier": "$CreateProductChange.Entity.Identifier"
  },
  "DetailsDocument": {
    "DeliveryOptions": [
      {
        "Details": {
          "SaaSUrlDeliveryOptionDetails": {

```

```

        "FulfillmentUrl": "https://www.aws.amazon.com/
marketplace/management"
    }
}
]
}
},
{
    "ChangeType": "ReleaseProduct",
    "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "$CreateProductChange.Entity.Identifier"
    },
    "DetailsDocument": {}
},
{
    "ChangeType": "CreateOffer",
    "ChangeName": "CreateOfferChange",
    "Entity": {
        "Type": "Offer@1.0"
    },
    "DetailsDocument": {
        "ProductId": "$CreateProductChange.Entity.Identifier"
    }
},
{
    "ChangeType": "UpdateInformation",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {
        "Name": "New Test Offer",
        "Description": "New offer description"
    }
},
{
    "ChangeType": "UpdateLegalTerms",
    "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "$CreateOfferChange.Entity.Identifier"
    },
    "DetailsDocument": {

```

```

        "Terms": [
            {
                "Type": "LegalTerm",
                "Documents": [
                    {
                        "Type": "StandardEula",
                        "Version": "2022-07-14"
                    }
                ]
            }
        ]
    },
    {
        "ChangeType": "UpdateSupportTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "SupportTerm",
                    "RefundPolicy": "Updated refund policy description"
                }
            ]
        }
    },
    {
        "ChangeType": "UpdatePricingTerms",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "PricingModel": "Contract",
            "Terms": [
                {
                    "Type": "ConfigurableUpfrontPricingTerm",
                    "CurrencyCode": "USD",
                    "RateCards": [
                        {
                            "Selector": {
                                "Type": "Duration",

```

```

        "Value": "P1M"
    },
    "RateCard": [
        {
            "DimensionKey": "BasicService",
            "Price": "20"
        },
        {
            "DimensionKey": "PremiumService",
            "Price": "25"
        }
    ],
    "Constraints": {
        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
    }
},
{
    "Selector": {
        "Type": "Duration",
        "Value": "P12M"
    },
    "RateCard": [
        {
            "DimensionKey": "BasicService",
            "Price": "150"
        },
        {
            "DimensionKey": "PremiumService",
            "Price": "300"
        }
    ],
    "Constraints": {
        "MultipleDimensionSelection": "Allowed",
        "QuantityConfiguration": "Allowed"
    }
}
]
}
},
{
    "ChangeType": "UpdateRenewalTerms",

```



```

        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "RenewalTerm"
                }
            ]
        }
    },
    {
        "ChangeType": "ReleaseOffer",
        "Entity": {
            "Type": "Offer@1.0",
            "Identifier": "$CreateOfferChange.Entity.Identifier"
        },
        "DetailsDocument": {}
    }
]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish my SaaS product and associated public offer (product will be in limited
state by default)
CAPI-05A
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):

```

```

if change_set is None:
    fname = "changeset1.json"
    change_set_file = os.path.join(os.path.dirname(__file__), fname)
    stringified_change_set = sd.stringify_changeset(change_set_file)

else:
    stringified_change_set = change_set

response = sc.usage_demo(
    stringified_change_set,
    "publish saas product and associated public offer",
)

return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 기존 초안에서 SaaS 제품 및 관련 공개 제안 게시

다음 코드 예제에서는 기존 초안을 바탕으로 SaaS 제품 및 관련된 공개 제안을 게시하는 방법을 보여줍니다. 제품은 기본적으로 제한된 상태입니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
{
```

```

    "Catalog": "AWSMarketplace",
    "ChangeSet": [
      {
        "ChangeType": "UpdateVisibility",
        "ChangeName": "CreateProductChange",
        "Entity": {
          "Type": "SaaSProduct@1.0",
          "Identifier": "prod-111111111111"
        },
        "DetailsDocument": {
          "TargetVisibility": "Public"
        }
      }
    ]
  }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 또는 SaaS 제품의 차원 업데이트

다음 코드 예제에서는 AMI 또는 SaaS 제품의 차원을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

이 예제를 실행하려면 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에서 다음 JSON changeset를 RunChangesets에 전달합니다.

```

{
  "Catalog": "AWSMarketplace",

```

```

    "ChangeSet": [
      {
        "ChangeType": "UpdateDimensions",
        "Entity": {
          "Type": "SaaSProduct@1.0",
          "Identifier": "prod-111111111111"
        },
        "DetailsDocument": [
          {
            "Key": "BasicService",
            "Types": [
              "Entitled"
            ],
            "Name": "Some new name",
            "Description": "Some new description"
          }
        ]
      }
    ]
  }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateDimensions",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-111111111111"
      }
    }
  ]
}

```

```

    },
    "DetailsDocument": [
        {
            "Key": "BasicService",
            "Types": [
                "Entitled"
            ],
            "Name": "Some new name",
            "Description": "Some new description"
        }
    ]
}

```

이 스크립트를 실행하여 changeset를 시작합니다. 헬퍼 함수는 유틸리티 섹션의 changeset를 시작하기 위한 유틸리티에 정의되어 있습니다.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to update (e.g name) dimensions
on my AMI or SaaS product
CAPI-24
"""

import os

import utils.start_changeset as sc
import utils.stringify_details as sd

def main(change_set=None):
    if change_set is None:
        fname = "changeset.json"
        change_set_file = os.path.join(os.path.dirname(__file__), fname)
        stringified_change_set = sd.stringify_changeset(change_set_file)

    else:
        stringified_change_set = change_set

```

```

    response = sc.usage_demo(
        stringified_change_set,
        "Update name dimensions on my AMI or SaaS product",
    )

    return response

if __name__ == "__main__":
    main()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace 카탈로그 API의 유틸리티

다음 코드 예제에서는 AWS MarketplaceCatalog API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 변경 세트를 시작하는 유틸리티](#)

AWSSDK를 사용하여 변경 세트를 시작하는 유틸리티

다음 코드 예제에서는 유틸리티를 정의하여 changeset를 시작하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

JSON 파일에서 changeset를 로드하고 처리를 시작하는 유틸리티입니다.

```
package com.example.awsmarketplace.catalogapi;

import java.io.ByteArrayInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;

import org.apache.commons.io.IOUtils;
import org.apache.commons.lang3.StringUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.core.document.Document;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.protocols.json.internal.unmarshall.document.DocumentUnmarshaller;
import software.amazon.awssdk.protocols.jsoncore.JsonNodeParser;
import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import software.amazon.awssdk.services.marketplacecatalog.model.Change;
import software.amazon.awssdk.services.marketplacecatalog.model.Entity;
import
    software.amazon.awssdk.services.marketplacecatalog.model.StartChangeSetRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.StartChangeSetResponse;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.ToNumberPolicy;
import com.example.awsmarketplace.catalogapi.Entity.ChangeSet;
import com.example.awsmarketplace.catalogapi.Entity.ChangeSetEntity;
import com.example.awsmarketplace.catalogapi.Entity.Root;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;
import com.example.awsmarketplace.utils.StringSerializer;

/**
 * Before running this Java V2 code example, convert all Details attribute to
 * DetailsDocument if any
 */

public class RunChangesets {

    private static final Gson GSON = new GsonBuilder()
```

```
.setObjectToNumberStrategy(ToNumberPolicy.LAZILY_PARSED_NUMBER)
.registerTypeAdapter(String.class, new StringSerializer())
.create();

public static void main(String[] args) {

    // input json can be specified here or passed from input parameter
    String inputChangeSetFile = "changeSets/offers/
CreateReplacementOfferFromAGWithContractPricingDetailDocument.json";

    if (args.length > 0)
        inputChangeSetFile = args[0];

    // parse the input changeset file to string for process
    String changeSetsInput = readChangeSetToString(inputChangeSetFile);

    // process the changeset request
    try {
        StartChangeSetResponse result = getChangeSetRequestResult(changeSetsInput);
        ReferenceCodesUtils.formatOutput(result);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static StartChangeSetResponse getChangeSetRequestResult(String
changeSetsInput) throws IOException {

    //set up AWS credentials
    MarketplaceCatalogClient marketplaceCatalogClient =
        MarketplaceCatalogClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    //changeset list to save all the changesets in the changesets file
    List<Change> changeSetLists = new ArrayList<Change>();

    // read all changesets into object
    Root root = GSON.fromJson(changeSetsInput, Root.class);

    // process each changeset and add each changeset request to changesets list
    for (ChangeSet cs : root.changeSet) {
```



```
ChangeSetEntity entity = cs.Entity;
String entityType = entity.Type;
String entityIdIdentifier = StringUtils.defaultIfBlank(entity.Identifier, null);
Document detailsDocument = getDocumentFromObject(cs.DetailsDocument);

Entity awsEntity =
    Entity.builder()
        .type(entityType)
        .identifier(entityIdentifier)
        .build();

Change inputChangeRequest =
    Change.builder()
        .changeType(cs.ChangeType)
        .changeName(cs.ChangeName)
        .entity(awsEntity)
        .detailsDocument(detailsDocument)
        .build();

changeSetLists.add(inputChangeRequest);
}

// process all changeset requests
StartChangeSetRequest startChangeSetRequest =
    StartChangeSetRequest.builder()
        .catalog(root.catalog)
        .changeSet(changeSetLists)
        .build();

StartChangeSetResponse result =
marketplaceCatalogClient.startChangeSet(startChangeSetRequest);

return result;
}

public static Document getDocumentFromObject(Object detailsObject) {

    String detailsString = "{}";
    try {
        detailsString = IOUtils.toString(new
        ByteArrayInputStream(GSON.toJson(detailsObject).getBytes()), "UTF-8");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

```

    JsonNodeParser jsonNodeParser = JsonNodeParser.create();
    Document doc = jsonNodeParser.parse(detailsString).visit(new
    DocumentUnmarshaller());
    return doc;
}

public static String readChangeSetToString (String inputChangeSetFile) {

    InputStream changesetInputStream =
    RunChangesets.class.getClassLoader().getResourceAsStream(inputChangeSetFile);

    String changeSetsInput = null;

    try {
        changeSetsInput = IOUtils.toString(changesetInputStream, "UTF-8");
    } catch (IOException e) {
        e.printStackTrace();
    }

    return changeSetsInput;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [StartChangeSet](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

changeset를 시작하는 유틸리티입니다.

```
"""
```

Purpose:

Generic function to start a changeset
"""

import logging

import boto3

from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

def generate_changeset(mp_client, change_set, change_set_name):
 """

Start changeset
 """

try:

response = mp_client.start_change_set(
 Catalog="AWSMarketplace",
 ChangeSet=change_set,
 ChangeSetName=change_set_name,
)
 logger.info("Changeset created!")
 logger.info("ChangeSet ID: %s", response["ChangeSetId"])
 logger.info("ChangeSet ARN: %s", response["ChangeSetArn"])

return response

except ClientError as e:

logger.exception("Unexpected error: %s", e)
 raise

def usage_demo(change_set, change_set_name):

logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

print("-" * 88)

print("Executing changeset: " + change_set_name)

print("-" * 88)

mp_client = boto3.client("marketplace-catalog")

response = generate_changeset(mp_client, change_set, change_set_name)

```

return response

print("-" * 88)

```

JSON 파일에서 changeset를 로드하는 유틸리티입니다.

```

"""
Purpose:

This module will stringify the details sections of a changeset file.
"""

import json

def pretty_print(response):
    json_object = json.dumps(response, indent=4)
    print(json_object)

# open json file from path
def open_json_file(filename):
    with open(filename, "r") as f:
        return json.load(f)

def stringify_details_sections(json_object):
    """
    Loops through every change type in the changeset to look for non-empty
    details section and stringifies them
    """
    for change_type in json_object["ChangeSet"]:
        # Only stringify details section if it is not empty
        if "Details" in change_type and change_type["Details"] != "{}":
            string_details = json.dumps(change_type["Details"])
            change_type["Details"] = string_details
        else:
            pass

    return json_object["ChangeSet"]

```

```
def stringify_changeset(file_path):
    changeset_file = open_json_file(file_path)
    changeset_stringified = stringify_details_sections(changeset_file)

    return changeset_stringified
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [StartChangeSet](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDKs를 사용한 AWS Marketplace계약 API의 코드 예제

다음 코드 예제에서는 AWS 소프트웨어 개발 키트(SDK)와 함께 AWS Marketplace계약 API를 사용하는 방법을 보여줍니다.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWS Marketplace계약 API

- [계약 API에 대한 AWS Marketplace계약](#)
 - [AWSSDK를 사용하여 계약 취소 요청 수락](#)
 - [AWSSDK를 사용하여 계약 결제 요청 수락](#)
 - [AWSSDK를 사용하여 SaaS 계약 갱신 기간 수정](#)
 - [AWSSDK를 사용하여 사용량 요금 모델에 대한 AMI ConfigurableUpfrontPricingTerm 수정](#)
 - [AWSSDK를 사용하여 새 AMI 무료 평가판 계약 생성](#)
 - [AWSSDK를 사용하여 선결제로 새 SaaS 계약 생성](#)
 - [AWSSDK를 사용하여 모든 계약 IDs 가져오기](#)
 - [AWSSDK를 사용하여 모든 계약 가져오기](#)
 - [AWSSDK를 사용하여 계약에서 고객 ID 가져오기](#)
 - [AWSSDK를 사용하여 계약 취소 요청 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약 결제 요청 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약에서 재무 세부 정보 가져오기](#)
 - [AWSSDK를 사용하여 계약에서 무료 평가판 세부 정보 가져오기](#)

- [AWSSDK를 사용하여 계약에 대한 정보 가져오기](#)
- [AWSSDK를 사용하여 계약에서 제품 및 제안 세부 정보 가져오기](#)
- [AWSSDK를 사용하여 계약의 EULA 가져오기](#)
- [AWSSDK를 사용하여 계약의 자동 갱신 조건 가져오기](#)
- [AWSSDK를 사용하여 계약에서 구매한 차원 가져오기](#)
- [AWSSDK를 사용하여 계약에서 구매한 각 차원의 인스턴스 가져오기](#)
- [AWSSDK를 사용하여 계약의 결제 일정 가져오기](#)
- [AWSSDK를 사용하여 계약에서 차원당 요금 가져오기](#)
- [AWSSDK를 사용하여 계약의 요금 유형 가져오기](#)
- [AWSSDK를 사용하여 계약의 제품 유형 가져오기](#)
- [AWSSDK를 사용하여 계약 상태 가져오기](#)
- [AWSSDK를 사용하여 계약의 지원 조건 가져오기](#)
- [AWSSDK를 사용하여 계약 조건 가져오기](#)
- [AWSSDK를 사용하여 계약 취소 요청 나열](#)
- [AWSSDK를 사용하여 계약 결제 요청 나열](#)
- [AWSSDK를 사용하여 계약 취소 요청 거부](#)
- [AWSSDK를 사용하여 계약 결제 요청 거부](#)
- [AWSSDK를 사용하여 SaaS 계약 계약을 계약 기반 제안으로 대체](#)
- [AWSSDK를 사용하여 SaaS 무료 평가판을 소비 요금으로 계약으로 교체](#)
- [AWSSDK를 사용하여 SaaS 사용량 기반 요금 기간 대체 및 이전 계약 취소](#)
- [AWSSDK를 사용하여 AMI 사용량 기반 요금 기간을 바꾸고 기존 ConfigurableUpfrontPricingTerm 유지](#)
- [AWSSDK를 사용하여 계정 ID별로 계약 검색](#)
- [AWSSDK를 사용하여 계약 ID로 계약 검색](#)
- [AWSSDK를 사용하여 종료 날짜별로 계약 검색](#)
- [AWSSDK를 사용하여 제안 ID로 계약 검색](#)
- [AWSSDK를 사용하여 제품 ID별로 계약 검색](#)
- [AWSSDK를 사용하여 상태별로 계약 검색](#)
- [AWSSDK를 사용하여 하나의 사용자 지정 필터가 있는 계약 검색](#)
- [AWSSDK를 사용하여 두 개의 사용자 지정 필터가 있는 계약 검색](#)

- [AWSSDK를 사용하여 계약 수락 후 구매 주문 업데이트](#)

계약 API에 대한 AWS Marketplace 계약

다음 코드 예제에서는 AWS Marketplace 계약 API를 AWSSDKs와 함께 사용하는 방법을 보여줍니다.

예제

- [AWSSDK를 사용하여 계약 취소 요청 수락](#)
- [AWSSDK를 사용하여 계약 결제 요청 수락](#)
- [AWSSDK를 사용하여 SaaS 계약 갱신 기간 수정](#)
- [AWSSDK를 사용하여 사용량 요금 모델에 대한 AMI ConfigurableUpfrontPricingTerm 수정](#)
- [AWSSDK를 사용하여 새 AMI 무료 평가판 계약 생성](#)
- [AWSSDK를 사용하여 선결제로 새 SaaS 계약 생성](#)
- [AWSSDK를 사용하여 모든 계약 IDs 가져오기](#)
- [AWSSDK를 사용하여 모든 계약 가져오기](#)
- [AWSSDK를 사용하여 계약에서 고객 ID 가져오기](#)
- [AWSSDK를 사용하여 계약 취소 요청 세부 정보 가져오기](#)
- [AWSSDK를 사용하여 계약 결제 요청 세부 정보 가져오기](#)
- [AWSSDK를 사용하여 계약에서 재무 세부 정보 가져오기](#)
- [AWSSDK를 사용하여 계약에서 무료 평가판 세부 정보 가져오기](#)
- [AWSSDK를 사용하여 계약에 대한 정보 가져오기](#)
- [AWSSDK를 사용하여 계약에서 제품 및 제안 세부 정보 가져오기](#)
- [AWSSDK를 사용하여 계약의 EULA 가져오기](#)
- [AWSSDK를 사용하여 계약의 자동 갱신 조건 가져오기](#)
- [AWSSDK를 사용하여 계약에서 구매한 차원 가져오기](#)
- [AWSSDK를 사용하여 계약에서 구매한 각 차원의 인스턴스 가져오기](#)
- [AWSSDK를 사용하여 계약의 결제 일정 가져오기](#)
- [AWSSDK를 사용하여 계약에서 차원당 요금 가져오기](#)
- [AWSSDK를 사용하여 계약의 요금 유형 가져오기](#)
- [AWSSDK를 사용하여 계약의 제품 유형 가져오기](#)
- [AWSSDK를 사용하여 계약 상태 가져오기](#)

- [AWSSDK를 사용하여 계약의 지원 조건 가져오기](#)
- [AWSSDK를 사용하여 계약 조건 가져오기](#)
- [AWSSDK를 사용하여 계약 취소 요청 나열](#)
- [AWSSDK를 사용하여 계약 결제 요청 나열](#)
- [AWSSDK를 사용하여 계약 취소 요청 거부](#)
- [AWSSDK를 사용하여 계약 결제 요청 거부](#)
- [AWSSDK를 사용하여 SaaS 계약 계약을 계약 기반 제안으로 대체](#)
- [AWSSDK를 사용하여 SaaS 무료 평가판을 소비 요금으로 계약으로 교체](#)
- [AWSSDK를 사용하여 SaaS 사용량 기반 요금 기간 대체 및 이전 계약 취소](#)
- [AWSSDK를 사용하여 AMI 사용량 기반 요금 기간을 바꾸고 기존 ConfigurableUpfrontPricingTerm 유지](#)
- [AWSSDK를 사용하여 계정 ID별로 계약 검색](#)
- [AWSSDK를 사용하여 계약 ID로 계약 검색](#)
- [AWSSDK를 사용하여 종료 날짜별로 계약 검색](#)
- [AWSSDK를 사용하여 제안 ID로 계약 검색](#)
- [AWSSDK를 사용하여 제품 ID별로 계약 검색](#)
- [AWSSDK를 사용하여 상태별로 계약 검색](#)
- [AWSSDK를 사용하여 하나의 사용자 지정 필터가 있는 계약 검색](#)
- [AWSSDK를 사용하여 두 개의 사용자 지정 필터가 있는 계약 검색](#)
- [AWSSDK를 사용하여 계약 수락 후 구매 주문 업데이트](#)

AWSSDK를 사용하여 계약 취소 요청 수락

다음 코드 예제에서는 판매자가 시작한 계약 취소 요청을 수락하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.


```

package com.example.awsmarketplace.agreementapi.buyer.agreementCancellation;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementCancellationRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementCancellationResponse;

public class AcceptAgreementCancellationRequest {

    private static final String AGREEMENT_ID = "<AGREEMENT ID HERE>";
    private static final String AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT
    CANCELLATION REQUEST ID HERE>";

    public static void main(String[] args) {
        acceptAgreementCancellationRequest();
    }

    private static void acceptAgreementCancellationRequest() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        AcceptAgreementCancellationRequest request =
            AcceptAgreementCancellationRequest.builder()
                .agreementId(AGREEMENT_ID)

                .agreementCancellationRequestId(AGREEMENT_CANCELLATION_REQUEST_ID)
                .build();

        AcceptAgreementCancellationResponse response =
            marketplaceAgreementClient.acceptAgreementCancellationRequest(request);

        System.out.println("Agreement ID: " + response.agreementId());
        System.out.println("Cancellation Request ID: " +
            response.agreementCancellationRequestId());
        System.out.println("Status: " + response.statusAsString());
    }
}

```

```

        System.out.println("Description: " + response.description());
        System.out.println("Reason Code: " + response.reasonCodeAsString());
        System.out.println("Created At: " + response.createdAt());
        System.out.println("Updated At: " + response.updatedAt());
    }
}

```

- API 세부 정보는 API 참조의 [AcceptAgreementCancellationRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

const {
    MarketplaceAgreementClient,
    AcceptAgreementCancellationRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const AGREEMENT_ID = "<AGREEMENT ID HERE>";
const AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT CANCELLATION REQUEST ID HERE>";

async function acceptAgreementCancellationRequest() {
    const client = new MarketplaceAgreementClient();

    const response = await client.send(
        new AcceptAgreementCancellationRequestCommand({
            agreementId: AGREEMENT_ID,
            agreementCancellationRequestId: AGREEMENT_CANCELLATION_REQUEST_ID,
        })
    );

    console.log("Agreement ID: " + response.agreementId);
}

```

```

        console.log("Cancellation Request ID: " +
response.agreementCancellationRequestId);
        console.log("Status: " + response.status);
        console.log("Description: " + response.description);
        console.log("Reason Code: " + response.reasonCode);
        console.log("Created At: " + response.createdAt);
        console.log("Updated At: " + response.updatedAt);
    }

    acceptAgreementCancellationRequest();

```

- API 세부 정보는 API 참조의 [AcceptAgreementCancellationRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class AcceptAgreementCancellationRequest:

    AGREEMENT_ID = "<AGREEMENT ID HERE>"
    AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT CANCELLATION REQUEST ID HERE>"

    @staticmethod
    def accept_agreement_cancellation_request():

```

```

client = boto3.client("marketplace-agreement")

response = client.accept_agreement_cancellation_request(
    agreementId=AcceptAgreementCancellationRequest.AGREEMENT_ID,
    agreementCancellationRequestId=AcceptAgreementCancellationRequest.AGREEMENT_CANCELLATION_ID
)

print("Agreement ID: " + response["agreementId"])
print("Cancellation Request ID: " +
response["agreementCancellationRequestId"])
print("Status: " + str(response.get("status", "")))
print("Description: " + str(response.get("description", "")))
print("Reason Code: " + str(response.get("reasonCode", "")))
print("Created At: " + str(response.get("createdAt", "")))
print("Updated At: " + str(response.get("updatedAt", "")))

if __name__ == "__main__":
    AcceptAgreementCancellationRequest.accept_agreement_cancellation_request()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [AcceptAgreementCancellationRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 결제 요청 수락

다음 코드 예제에서는 판매자가 시작한 계약 결제 요청을 수락하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer.paymentRequest;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementPaymentRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementPaymentRequestRequest;

public class AcceptAgreementPaymentRequest {

    private static final String AGREEMENT_ID = "<AGREEMENT ID HERE>";
    private static final String PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>";
    private static final String PURCHASE_ORDER_REFERENCE = "<PURCHASE ORDER
REFERENCE HERE>";

    public static void main(String[] args) {
        acceptAgreementPaymentRequest();
    }

    private static void acceptAgreementPaymentRequest() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        AcceptAgreementPaymentRequestRequest request =
            AcceptAgreementPaymentRequestRequest.builder()
                .agreementId(AGREEMENT_ID)
                .paymentRequestId(PAYMENT_REQUEST_ID)
                .purchaseOrderReference(PURCHASE_ORDER_REFERENCE)
                .build();

        AcceptAgreementPaymentRequestResponse response =
            marketplaceAgreementClient.acceptAgreementPaymentRequest(request);

        System.out.println("Payment Request ID: " + response.paymentRequestId());
        System.out.println("Agreement ID: " + response.agreementId());
        System.out.println("Status: " + response.statusAsString());
    }
}
```

```

        System.out.println("Name: " + response.name());
        System.out.println("Charge Amount: " + response.chargeAmount());
        System.out.println("Currency Code: " + response.currencyCode());
        System.out.println("Created At: " + response.createdAt());
        System.out.println("Updated At: " + response.updatedAt());
    }
}

```

- API 세부 정보는 API 참조의 [AcceptAgreementPaymentRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

const {
    MarketplaceAgreementClient,
    AcceptAgreementPaymentRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const AGREEMENT_ID = "<AGREEMENT ID HERE>";
const PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>";
const PURCHASE_ORDER_REFERENCE = "<PURCHASE ORDER REFERENCE HERE>";

async function acceptAgreementPaymentRequest() {
    const client = new MarketplaceAgreementClient();

    const response = await client.send(
        new AcceptAgreementPaymentRequestCommand({
            agreementId: AGREEMENT_ID,
            paymentRequestId: PAYMENT_REQUEST_ID,
            purchaseOrderReference: PURCHASE_ORDER_REFERENCE,
        })
    );
}

```

```

        console.log("Payment Request ID: " + response.paymentRequestId);
        console.log("Agreement ID: " + response.agreementId);
        console.log("Status: " + response.status);
        console.log("Name: " + response.name);
        console.log("Charge Amount: " + response.chargeAmount);
        console.log("Currency Code: " + response.currencyCode);
        console.log("Created At: " + response.createdAt);
        console.log("Updated At: " + response.updatedAt);
    }

    acceptAgreementPaymentRequest();

```

- API 세부 정보는 API 참조의 [AcceptAgreementPaymentRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class AcceptAgreementPaymentRequest:

    AGREEMENT_ID = "<AGREEMENT ID HERE>"
    PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>"
    PURCHASE_ORDER_REFERENCE = "<PURCHASE ORDER REFERENCE HERE>"

```

```

@staticmethod
def accept_agreement_payment_request():
    client = boto3.client("marketplace-agreement")

    response = client.accept_agreement_payment_request(
        agreementId=AcceptAgreementPaymentRequest.AGREEMENT_ID,
        paymentRequestId=AcceptAgreementPaymentRequest.PAYMENT_REQUEST_ID,
        purchaseOrderReference=AcceptAgreementPaymentRequest.PURCHASE_ORDER_REFERENCE,
    )

    print("Payment Request ID: " + response["paymentRequestId"])
    print("Agreement ID: " + response["agreementId"])
    print("Status: " + str(response.get("status", "")))
    print("Name: " + str(response.get("name", "")))
    print("Charge Amount: " + str(response.get("chargeAmount", "")))
    print("Currency Code: " + str(response.get("currencyCode", "")))
    print("Created At: " + str(response.get("createdAt", "")))
    print("Updated At: " + str(response.get("updatedAt", "")))

if __name__ == "__main__":
    AcceptAgreementPaymentRequest.accept_agreement_payment_request()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [AcceptAgreementPaymentRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 계약 갱신 기간 수정

다음 코드 예제에서는 SaaS 계약 계약을 수정하여 갱신 기간을 추가하거나 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse
import
    software.amazon.awssdk.services.marketplaceagreement.model.ConfigurableUpfrontPricingTerm
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse
import software.amazon.awssdk.services.marketplaceagreement.model.Dimension;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementEntitlementsResponse
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RenewalTermConfiguration;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RequestedTermConfiguration;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to create a SaaS agreement with CONTRACT pricing model and
 * then turn on
 * the auto-renewal setting using the AWS Marketplace Agreement Service APIs.
 */
```

```

* <p>Scenario: A buyer subscribes to a SaaS product using a public offer that
supports
* auto-renewal. After acceptance, the buyer decides to amend the agreement to
enable
* auto-renewal via the RenewalTerm configuration.
*
* <p>Before running this sample, replace the placeholder constants below with
values from
* your AWS Marketplace offer:
* <ul>
*   <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
offer.</li>
*   <li>Term IDs (starting with {@code term-}) – found in the offer's term
list.</li>
*   <li>{@code SELECTOR_VALUE} – duration for the agreement (e.g., {@code P1M}
for 1 month).</li>
*   <li>{@code DIMENSION_1_KEY} – the dimension key defined in the offer.</li>
* </ul>
*/
public class AmendSaaSContractRenewalTerm {

    // The agreementProposalId from the offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-
proposal-identifier>";

    // Term ID for the ConfigurableUpfrontPricingTerm in your offer.
    private static final String CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-
configurable-upfront-pricing-term-id>";

    // Duration for the agreement (e.g., "P1M" for 1 month, "P12M" for 1 year).
    private static final String SELECTOR_VALUE = "<your-selector-value>";

    // The dimension key defined in your offer.
    private static final String DIMENSION_1_KEY = "<your-dimension-key>";

    // Quantity for the dimension.
    private static final int DIMENSION_1_VALUE = 1;

    // Term ID for the RenewalTerm in your offer.
    private static final String RENEWAL_TERM_ID = "<your-renewal-term-id>";

    // Term ID for the LegalTerm in your offer.
    private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

```

```

// Term ID for the SupportTerm in your offer.
private static final String SUPPORT_TERM_ID = "<your-support-term-id>";

public static void main(String[] args) {
    amendSaaSContractAgreementRenewalTerm();
}

/**
 * Full end-to-end flow:
 * 1. Create a SaaS agreement with CONTRACT pricing model with auto-renewal
disabled.
 * 2. Wait for entitlements to become active.
 * 3. Amend the agreement to enable auto-renewal.
 */
private static void amendSaaSContractAgreementRenewalTerm() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    RequestedTerm configurableUpfrontPricingTerm = RequestedTerm.builder()
        .id(CONFIGURABLE_UPFRONT_PRICING_TERM_ID)

        .configuration(RequestedTermConfiguration.fromConfigurableUpfrontPricingTermConfiguration(
            ConfigurableUpfrontPricingTermConfiguration.builder()
                .selectorValue(SELECTOR_VALUE)
                .dimensions(Dimension.builder()

                    .dimensionKey(DIMENSION_1_KEY)

                    .dimensionValue(DIMENSION_1_VALUE)

                    .build())

                .build()))

        .build();

    // Initial agreement: auto-renewal disabled.
    RequestedTerm renewalTerm = RequestedTerm.builder()
        .id(RENEWAL_TERM_ID)

        .configuration(RequestedTermConfiguration.fromRenewalTermConfiguration(
            RenewalTermConfiguration.builder()
                .enableAutoRenew(false)

                .build()))

```

```

        .build();

        RequestedTerm legalTerm =
        RequestedTerm.builder().id(LEGAL_TERM_ID).build();
        RequestedTerm supportTerm =
        RequestedTerm.builder().id(SUPPORT_TERM_ID).build();

        // --- Create and accept the initial SaaS agreement request with CONTRACT
        pricing model ---
        CreateAgreementRequestRequest createAgreementRequestRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.NEW)
                .requestedTerms(configurableUpfrontPricingTerm,
renewalTerm, legalTerm, supportTerm)

        .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
            .build();
        CreateAgreementRequestResponse createAgreementRequestResponse =

marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
        System.out.println("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId());

        AcceptAgreementRequestRequest acceptAgreementRequestRequest =
            AcceptAgreementRequestRequest.builder()

        .agreementRequestId(createAgreementRequestResponse.agreementRequestId())
            .build();
        AcceptAgreementRequestResponse acceptAgreementRequestResponse =

marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
        System.out.println("Agreement request accepted. AgreementId: " +
acceptAgreementRequestResponse.agreementId());

        // Wait for entitlements to become active before amending.
        System.out.println("Waiting for entitlements to become active...");
        GetAgreementEntitlementsResponse entitlementsResponse =
        AgreementApiUtils.pollUntilEntitlementsAvailable(
            marketplaceAgreementClient,
acceptAgreementRequestResponse.agreementId());
        System.out.println("Entitlements are now active.");
        AgreementApiUtils.formatOutput(entitlementsResponse);

```

```

// --- Amend: enable auto-renewal ---
RequestedTerm renewalTermAmended = RequestedTerm.builder()
    .id(RENEWAL_TERM_ID)

    .configuration(RequestedTermConfiguration.fromRenewalTermConfiguration(
        RenewalTermConfiguration.builder()
            .enableAutoRenew(true)
            .build()))
    .build();

// Use Intent.AMEND and sourceAgreementIdentifier to target the existing
// agreement.
CreateAgreementRequestRequest carRequest =
    CreateAgreementRequestRequest.builder()
        .clientToken(AgreementApiUtils.generateClientToken())
        .intent(Intent.AMEND)
        .requestedTerms(configurableUpfrontPricingTerm,
            renewalTermAmended, legalTerm, supportTerm)

        .sourceAgreementIdentifier(acceptAgreementRequestResponse.agreementId())
        .build();
CreateAgreementRequestResponse carResponse =
    marketplaceAgreementClient.createAgreementRequest(carRequest);
System.out.println("Amend agreement request created. AgreementRequestId:
" + carResponse.agreementRequestId());

AcceptAgreementRequestRequest aarRequest =
    AcceptAgreementRequestRequest.builder()
        .agreementRequestId(carResponse.agreementRequestId())
        .build();
AcceptAgreementRequestResponse aarResponse =
    marketplaceAgreementClient.acceptAgreementRequest(aarRequest);
System.out.println("Amendment accepted. Auto-renewal enabled. New
AgreementId: " + aarResponse.agreementId());
    }
}

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
    MarketplaceAgreementClient,
    CreateAgreementRequestCommand,
    AcceptAgreementRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken, formatOutput, pollUntilEntitlementsAvailable } =
    require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to create a SaaS agreement with CONTRACT pricing model and
 * then turn on
 * the auto-renewal setting using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: A buyer subscribes to a SaaS product using a public offer that
 * supports
 * auto-renewal. After acceptance, the buyer decides to amend the agreement to
 * enable
 * auto-renewal via the RenewalTerm configuration.
 *
 * Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
 *   - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
 *   - Term IDs (starting with "term-") – found in the offer's term list.
 *   - SELECTOR_VALUE – duration for the agreement (e.g., "P1M" for 1 month).
 *   - DIMENSION_1_KEY – the dimension key defined in the offer.
 */

// The agreementProposalId from the offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>";

// Term ID for the ConfigurableUpfrontPricingTerm in your offer.
```

```

const CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-term-id>";

// Duration for the agreement (e.g., "P1M" for 1 month, "P12M" for 1 year).
const SELECTOR_VALUE = "<your-selector-value>";

// The dimension key defined in your offer.
const DIMENSION_1_KEY = "<your-dimension-key>";

// Quantity for the dimension.
const DIMENSION_1_VALUE = 1;

// Term ID for the RenewalTerm in your offer.
const RENEWAL_TERM_ID = "<your-renewal-term-id>";

// Term ID for the LegalTerm in your offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// Term ID for the SupportTerm in your offer.
const SUPPORT_TERM_ID = "<your-support-term-id>";

/**
 * Full end-to-end flow:
 * 1. Create a SaaS agreement with CONTRACT pricing model with auto-renewal disabled.
 * 2. Wait for entitlements to become active.
 * 3. Amend the agreement to enable auto-renewal.
 */
async function amendSaaSContractAgreementRenewalTerm() {
    const client = new MarketplaceAgreementClient();

    const configurableUpfrontPricingTerm = {
        id: CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        configuration: {
            configurableUpfrontPricingTermConfiguration: {
                selectorValue: SELECTOR_VALUE,
                dimensions: [
                    { dimensionKey: DIMENSION_1_KEY, dimensionValue:
DIMENSION_1_VALUE },
                ],
            },
        },
    };
};

```

```
// Initial agreement: auto-renewal disabled.
const renewalTerm = {
  id: RENEWAL_TERM_ID,
  configuration: {
    renewalTermConfiguration: {
      enableAutoRenew: false,
    },
  },
};

const legalTerm = { id: LEGAL_TERM_ID };
const supportTerm = { id: SUPPORT_TERM_ID };

// --- Create and accept the initial SaaS agreement request with CONTRACT
pricing model ---
const createAgreementRequestResponse = await client.send(
  new CreateAgreementRequestCommand({
    clientToken: generateClientToken(),
    intent: "NEW",
    requestedTerms: [configurableUpfrontPricingTerm, renewalTerm,
legalTerm, supportTerm],
    agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
  })
);
console.log("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId);

const acceptAgreementRequestResponse = await client.send(
  new AcceptAgreementRequestCommand({
    agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
  })
);
console.log("Agreement request accepted. AgreementId: " +
acceptAgreementRequestResponse.agreementId);

// Wait for entitlements to become active before amending.
console.log("Waiting for entitlements to become active...");
const entitlementsResponse = await pollUntilEntitlementsAvailable(client,
acceptAgreementRequestResponse.agreementId);
console.log("Entitlements are now active.");
formatOutput(entitlementsResponse);

// --- Amend: enable auto-renewal ---
```



```

const renewalTermAmended = {
  id: RENEWAL_TERM_ID,
  configuration: {
    renewalTermConfiguration: {
      enableAutoRenew: true,
    },
  },
};

// Use Intent.AMEND and sourceAgreementIdentifier to target the existing
// agreement.
const carResponse = await client.send(
  new CreateAgreementRequestCommand({
    clientToken: generateClientToken(),
    intent: "AMEND",
    requestedTerms: [configurableUpfrontPricingTerm, renewalTermAmended,
legalTerm, supportTerm],
    sourceAgreementIdentifier:
acceptAgreementRequestResponse.agreementId,
  })
);
console.log("Amend agreement request created. AgreementRequestId: " +
carResponse.agreementRequestId);

const aarResponse = await client.send(
  new AcceptAgreementRequestCommand({
    agreementRequestId: carResponse.agreementRequestId,
  })
);
console.log("Amendment accepted. Auto-renewal enabled. New AgreementId: " +
aarResponse.agreementId);
}

amendSaaSContractAgreementRenewalTerm();

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Demonstrates how to create a SaaS agreement with CONTRACT pricing model and then
turn on
the auto-renewal setting using the AWS Marketplace Agreement Service APIs.

Scenario: A buyer subscribes to a SaaS product using a public offer that supports
auto-renewal. After acceptance, the buyer decides to amend the agreement to
enable
auto-renewal via the RenewalTerm configuration.

Before running this sample, replace the placeholder constants below with values
from
your AWS Marketplace offer:
    - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
    - Term IDs (starting with term-) – found in the offer's term list.
    - SELECTOR_VALUE – duration for the agreement (e.g., P1M for 1 month).
    - DIMENSION_1_KEY – the dimension key defined in the offer.
"""

import boto3

from utils.agreement_api_utils import (
    format_output,
    generate_client_token,
    poll_until_entitlements_available,
)

class AmendSaaSContractRenewalTerm:

    # The agreementProposalId from the offer.
    AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>"

```

```

# Term ID for the ConfigurableUpfrontPricingTerm in your offer.
CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-
term-id>"

# Duration for the agreement (e.g., "P1M" for 1 month, "P12M" for 1 year).
SELECTOR_VALUE = "<your-selector-value>"

# The dimension key defined in your offer.
DIMENSION_1_KEY = "<your-dimension-key>"

# Quantity for the dimension.
DIMENSION_1_VALUE = 1

# Term ID for the RenewalTerm in your offer.
RENEWAL_TERM_ID = "<your-renewal-term-id>"

# Term ID for the LegalTerm in your offer.
LEGAL_TERM_ID = "<your-legal-term-id>"

# Term ID for the SupportTerm in your offer.
SUPPORT_TERM_ID = "<your-support-term-id>"

@staticmethod
def amend_saas_contract_agreement_renewal_term():
    """
    Full end-to-end flow:
    1. Create a SaaS agreement with CONTRACT pricing model with auto-renewal
disabled.
    2. Wait for entitlements to become active.
    3. Amend the agreement to enable auto-renewal.
    """
    client = boto3.client("marketplace-agreement")
    cls = AmendSaaSContractRenewalTerm

    configurable_upfront_pricing_term = {
        "id": cls.CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        "configuration": {
            "configurableUpfrontPricingTermConfiguration": {
                "selectorValue": cls.SELECTOR_VALUE,
                "dimensions": [
                    {
                        "dimensionKey": cls.DIMENSION_1_KEY,
                        "dimensionValue": cls.DIMENSION_1_VALUE,

```

```

        },
    ],
}

# Initial agreement: auto-renewal disabled.
renewal_term = {
    "id": cls.RENEWAL_TERM_ID,
    "configuration": {
        "renewalTermConfiguration": {
            "enableAutoRenew": False,
        }
    },
}

legal_term = {"id": cls.LEGAL_TERM_ID}
support_term = {"id": cls.SUPPORT_TERM_ID}

# --- Create and accept the initial SaaS agreement request with CONTRACT
pricing model ---
create_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="NEW",
    requestedTerms=[configurable_upfront_pricing_term, renewal_term,
legal_term, support_term],
    agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
)
agreement_request_id = create_response["agreementRequestId"]
print("Agreement request created. AgreementRequestId: " +
agreement_request_id)

accept_response = client.accept_agreement_request(
    agreementRequestId=agreement_request_id
)
agreement_id = accept_response["agreementId"]
print("Agreement request accepted. AgreementId: " + agreement_id)

# Wait for entitlements to become active before amending.
print("Waiting for entitlements to become active...")
entitlements_response = poll_until_entitlements_available(client,
agreement_id)
print("Entitlements are now active.")
format_output(entitlements_response)

```

```

# --- Amend: enable auto-renewal ---
renewal_term_amended = {
    "id": cls.RENEWAL_TERM_ID,
    "configuration": {
        "renewalTermConfiguration": {
            "enableAutoRenew": True,
        }
    },
}

# Use Intent.AMEND and sourceAgreementIdentifier to target the existing
agreement.
car_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="AMEND",
    requestedTerms=[configurable_upfront_pricing_term,
renewal_term_amended, legal_term, support_term],
    sourceAgreementIdentifier=agreement_id,
)
print("Amend agreement request created. AgreementRequestId: " +
car_response["agreementRequestId"])

aar_response = client.accept_agreement_request(
    agreementRequestId=car_response["agreementRequestId"]
)
print("Amendment accepted. Auto-renewal enabled. New AgreementId: " +
aar_response["agreementId"])

if __name__ == "__main__":
    AmendSaaSContractRenewalTerm.amend_saas_contract_agreement_renewal_term()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 사용량 요금 모델에 대한 AMI ConfigurableUpfrontPricingTerm 수정

다음 코드 예제에서는 사용 요금 모델의 차원 수량을 업데이트하여 ConfigurableUpfrontPricingTerm과 AMI 계약을 수정하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse;
import
    software.amazon.awssdk.services.marketplaceagreement.model.ConfigurableUpfrontPricingTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Dimension;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementEntitlementsResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RequestedTermConfiguration;
import utils.AgreementApiUtils;

/**
```

```

* Demonstrates how to create an AMI agreement with
ConfigurableUpfrontPricingTerm and then amend the dimension quantity
* using the AWS Marketplace Agreement Service APIs.
*
* <p>Scenario: An AMI product with USAGE pricing requires two agreements:
* <ol>
*   <li>An agreement with <b>usageBasedPricingTerm (UBPT)</b> – accepted first
to establish the base agreement.</li>
*   <li>An agreement with <b>configurableUpfrontPricingTerm (CUPT)</b> –
accepted after the UBPT agreement entitlements are active.</li>
* </ol>
* Once both agreement entitlements are available, this sample shows how to
<b>amend</b> the agreement
* with configurableUpfrontPricingTerm (CUPT) to increase the dimension quantity.
*
* <p>Before running this sample, replace the placeholder constants below with
values from
* your AWS Marketplace offer:
* <ul>
*   <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
offer.</li>
*   <li>Term IDs (starting with {@code term-}) – found in the offer's term
list.</li>
*   <li>{@code SELECTOR_VALUE} – duration for the agreement
(e.g., {@code P365D} for one year).</li>
*   <li>{@code DIMENSION_1_KEY} – the dimension key defined in the offer (e.g.,
instance type).</li>
*   <li>{@code DIMENSION_1_VALUE} – initial quantity; {@code
NEW_DIMENSION_1_VALUE} – amended quantity.</li>
* </ul>
*/
public class AmendAmiConfigurableUpfrontPricingTermForUsagePricingModel {

    // The agreementProposalId from the offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-
proposal-identifier>";

    // Term ID for the ConfigurableUpfrontPricingTerm in your offer.
    private static final String CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-
configurable-upfront-pricing-term-id>";

    // Duration for the agreement (e.g., "P365D" for 365 days).
    private static final String SELECTOR_VALUE = "<your-selector-value>";

```

```

// The dimension key defined in your offer (e.g., an EC2 instance type like
"c6gn.medium").
private static final String DIMENSION_1_KEY = "<your-dimension-key>";

// Initial quantity for the dimension.
private static final int DIMENSION_1_VALUE = 1;

// Term ID for the UsageBasedPricingTerm in your offer.
private static final String USAGE_TERM_ID = "<your-usage-term-id>";

// Term ID for the LegalTerm in your offer.
private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

// Term ID for the ValidityTerm in your offer.
private static final String VALIDITY_TERM_ID = "<your-validity-term-id>";

// New quantity to use when amending the dimension of CUPT.
private static final int NEW_DIMENSION_1_VALUE = 5;

public static void main(String[] args) {
    amendAmiCUPTAgreement();
}

/**
 * Full end-to-end flow:
 * 1. Create and accept an agreement request with usageBasedPricingTerm.
 * 2. Wait for entitlements to become active, then create and accept an
agreement request with configurableUpfrontPricingTerm (CUPT).
 * 3. Wait for CUPT entitlements to become active, then amend the dimension
quantity.
 */
private static void amendAmiCUPTAgreement() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    RequestedTerm usageTerm = RequestedTerm.builder()
        .id(USAGE_TERM_ID)
        .build();
    RequestedTerm legalTerm = RequestedTerm.builder()
        .id(LEGAL_TERM_ID)
        .build();

```



```

        RequestedTerm validityTerm = RequestedTerm.builder()
            .id(VALIDITY_TERM_ID)
            .build();

// --- Agreement with UBPT ---
CreateAgreementRequestRequest createAgreementRequestRequest =
    CreateAgreementRequestRequest.builder()
        .clientToken(AgreementApiUtils.generateClientToken())
        .intent(Intent.NEW)
        .requestedTerms(usageTerm, legalTerm, validityTerm)

.agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
    .build();
CreateAgreementRequestResponse createAgreementRequestResponse =
marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
    System.out.println("Agreement request with UBPT created.
AgreementRequestId: " + createAgreementRequestResponse.agreementRequestId());

    AcceptAgreementRequestRequest acceptAgreementRequestRequest =
        AcceptAgreementRequestRequest.builder()

.agreementRequestId(createAgreementRequestResponse.agreementRequestId())
    .build();
    AcceptAgreementRequestResponse acceptAgreementRequestResponse =
marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
    System.out.println("Agreement request with UBPT accepted. AgreementId: "
+ acceptAgreementRequestResponse.agreementId());

    // Wait for entitlements to become active before creating the agreement
    with CUPT.
    System.out.println("Waiting for UBPT agreement entitlements to become
    active...");
    GetAgreementEntitlementsResponse entitlementsResponse =
    AgreementApiUtils.pollUntilEntitlementsAvailable(
        marketplaceAgreementClient,
        acceptAgreementRequestResponse.agreementId());
    System.out.println("UBPT agreement entitlements are now active.");
    AgreementApiUtils.formatOutput(entitlementsResponse);

// --- Agreement with configurableUpfrontPricingTerm (CUPT) ---
    RequestedTerm configurableUpfrontPricingTerm = RequestedTerm.builder()
        .id(CONFIGURABLE_UPFRONT_PRICING_TERM_ID)

```

```

        .configuration(RequestedTermConfiguration.fromConfigurableUpfrontPricingTermConfiguration(
            ConfigurableUpfrontPricingTermConfiguration.builder()
                .selectorValue(SELECTOR_VALUE)
                .dimensions(Dimension.builder()

                    .dimensionKey(DIMENSION_1_KEY)

                    .dimensionValue(DIMENSION_1_VALUE)

                                .build())
                            .build()))
                .build();

        CreateAgreementRequestRequest carRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.NEW)
                .requestedTerms(configurableUpfrontPricingTerm,
                    legalTerm, validityTerm)

                    .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
                        .build();

        CreateAgreementRequestResponse carResponse =
            marketplaceAgreementClient.createAgreementRequest(carRequest);
        System.out.println("Agreement request with CUPT created.
AgreementRequestId: " + carResponse.agreementRequestId());

        AcceptAgreementRequestRequest aarRequest =
            AcceptAgreementRequestRequest.builder()
                .agreementRequestId(carResponse.agreementRequestId())
                .build();

        AcceptAgreementRequestResponse aarResponse =
            marketplaceAgreementClient.acceptAgreementRequest(aarRequest);
        final String cuptAgreementId = aarResponse.agreementId();
        System.out.println("Agreement request with CUPT accepted. AgreementId: "
+ cuptAgreementId);

        // Wait for entitlements to become active before amending.
        System.out.println("Waiting for CUPT agreement entitlements to become
active...");
        GetAgreementEntitlementsResponse cuptEntitlementsResponse =
            AgreementApiUtils.pollUntilEntitlementsAvailable(
                marketplaceAgreementClient, cuptAgreementId);
        System.out.println("CUPT agreement entitlements are now active.");

```

```

        AgreementApiUtils.formatOutput(cuptEntitlementsResponse);

        // --- Amend Agreement with CUPT ---
        // Increase the dimension quantity using Intent.AMEND and
        sourceAgreementIdentifier.
            RequestedTerm newConfig = RequestedTerm.builder()
                .id(CONFIGURABLE_UPFRONT_PRICING_TERM_ID)

            .configuration(RequestedTermConfiguration.fromConfigurableUpfrontPricingTermConfiguration(
                ConfigurableUpfrontPricingTermConfiguration.builder()
                    .selectorValue(SELECTOR_VALUE)
                    .dimensions(Dimension.builder()

            .dimensionKey(DIMENSION_1_KEY)

            .dimensionValue(NEW_DIMENSION_1_VALUE) // Increase quantity for this dimension
            key

                                                    .build()))
                .build()))
            .build();

        CreateAgreementRequestRequest carAmendRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.AMEND)
                .requestedTerms(newConfig, legalTerm, validityTerm)
                .sourceAgreementIdentifier(cuptAgreementId)
                .build();
        CreateAgreementRequestResponse carAmendResponse =

        marketplaceAgreementClient.createAgreementRequest(carAmendRequest);
        System.out.println("Amendment of CUPT agreement request created.
        AgreementRequestId: " + carAmendResponse.agreementRequestId());

        AcceptAgreementRequestRequest aarAmendRequest =
            AcceptAgreementRequestRequest.builder()

        .agreementRequestId(carAmendResponse.agreementRequestId())
            .build();
        AcceptAgreementRequestResponse aarAmendResponse =

        marketplaceAgreementClient.acceptAgreementRequest(aarAmendRequest);
        System.out.println("Amendment accepted. New AgreementId: " +
        aarAmendResponse.agreementId());

```

```
}
}
```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
  MarketplaceAgreementClient,
  CreateAgreementRequestCommand,
  AcceptAgreementRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken, formatOutput, pollUntilEntitlementsAvailable } =
  require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to create an AMI agreement with
 * ConfigurableUpfrontPricingTerm and then amend the dimension quantity
 * using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: An AMI product with USAGE pricing requires two agreements:
 * 1. An agreement with usageBasedPricingTerm (UBPT) – accepted first to
 * establish the base agreement.
 * 2. An agreement with configurableUpfrontPricingTerm (CUPT) – accepted after
 * the UBPT agreement entitlements are active.
 *
 * Once both agreement entitlements are available, this sample shows how to amend
 * the agreement
 * with configurableUpfrontPricingTerm (CUPT) to increase the dimension quantity.
 *
 * Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
```

```

*   - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
*   - Term IDs (starting with "term-") – found in the offer's term list.
*   - SELECTOR_VALUE – duration for the agreement (e.g., "P365D" for one year).
*   - DIMENSION_1_KEY – the dimension key defined in the offer (e.g., instance
type).
*   - DIMENSION_1_VALUE – initial quantity; NEW_DIMENSION_1_VALUE – amended
quantity.
*/

// The agreementProposalId from the offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>";

// Term ID for the ConfigurableUpfrontPricingTerm in your offer.
const CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-
term-id>";

// Duration for the agreement (e.g., "P365D" for 365 days).
const SELECTOR_VALUE = "<your-selector-value>";

// The dimension key defined in your offer (e.g., an EC2 instance type like
"c6gn.medium").
const DIMENSION_1_KEY = "<your-dimension-key>";

// Initial quantity for the dimension.
const DIMENSION_1_VALUE = 1;

// Term ID for the UsageBasedPricingTerm in your offer.
const USAGE_TERM_ID = "<your-usage-term-id>";

// Term ID for the LegalTerm in your offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// Term ID for the ValidityTerm in your offer.
const VALIDITY_TERM_ID = "<your-validity-term-id>";

// New quantity to use when amending the dimension of CUPT.
const NEW_DIMENSION_1_VALUE = 5;

/**
* Full end-to-end flow:
* 1. Create and accept an agreement request with usageBasedPricingTerm.
* 2. Wait for entitlements to become active, then create and accept an agreement
request with configurableUpfrontPricingTerm (CUPT).

```

```

* 3. Wait for CUPT entitlements to become active, then amend the dimension
quantity.
*/
async function amendAmiCUPTAgreement() {
    const client = new MarketplaceAgreementClient();

    const usageTerm = { id: USAGE_TERM_ID };
    const legalTerm = { id: LEGAL_TERM_ID };
    const validityTerm = { id: VALIDITY_TERM_ID };

    // --- Agreement with UBPT ---
    const createAgreementRequestResponse = await client.send(
        new CreateAgreementRequestCommand({
            clientToken: generateClientToken(),
            intent: "NEW",
            requestedTerms: [usageTerm, legalTerm, validityTerm],
            agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
        })
    );
    console.log("Agreement request with UBPT created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId);

    const acceptAgreementRequestResponse = await client.send(
        new AcceptAgreementRequestCommand({
            agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
        })
    );
    console.log("Agreement request with UBPT accepted. AgreementId: " +
acceptAgreementRequestResponse.agreementId);

    // Wait for entitlements to become active before creating the agreement with
CUPT.
    console.log("Waiting for UBPT agreement entitlements to become active...");
    const entitlementsResponse = await pollUntilEntitlementsAvailable(client,
acceptAgreementRequestResponse.agreementId);
    console.log("UBPT agreement entitlements are now active.");
    formatOutput(entitlementsResponse);

    // --- Agreement with configurableUpfrontPricingTerm (CUPT) ---
    const configurableUpfrontPricingTerm = {
        id: CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        configuration: {
            configurableUpfrontPricingTermConfiguration: {

```

```

        selectorValue: SELECTOR_VALUE,
        dimensions: [
            { dimensionKey: DIMENSION_1_KEY, dimensionValue:
DIMENSION_1_VALUE },
        ],
    },
};

const carResponse = await client.send(
    new CreateAgreementRequestCommand({
        clientToken: generateClientToken(),
        intent: "NEW",
        requestedTerms: [configurableUpfrontPricingTerm, legalTerm,
validityTerm],
        agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
    })
);
console.log("Agreement request with CUPT created. AgreementRequestId: " +
carResponse.agreementRequestId);

const aarResponse = await client.send(
    new AcceptAgreementRequestCommand({
        agreementRequestId: carResponse.agreementRequestId,
    })
);
const cuptAgreementId = aarResponse.agreementId;
console.log("Agreement request with CUPT accepted. AgreementId: " +
cuptAgreementId);

// Wait for entitlements to become active before amending.
console.log("Waiting for CUPT agreement entitlements to become active...");
const cuptEntitlementsResponse = await pollUntilEntitlementsAvailable(client,
cuptAgreementId);
console.log("CUPT agreement entitlements are now active.");
formatOutput(cuptEntitlementsResponse);

// --- Amend Agreement with CUPT ---
// Increase the dimension quantity using Intent.AMEND and
sourceAgreementIdentifier.
const newConfig = {
    id: CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
    configuration: {
        configurableUpfrontPricingTermConfiguration: {

```

```

        selectorValue: SELECTOR_VALUE,
        dimensions: [
            { dimensionKey: DIMENSION_1_KEY, dimensionValue:
NEW_DIMENSION_1_VALUE }, // Increase quantity for this dimension key
        ],
    },
};

const carAmendResponse = await client.send(
    new CreateAgreementRequestCommand({
        clientToken: generateClientToken(),
        intent: "AMEND",
        requestedTerms: [newConfig, legalTerm, validityTerm],
        sourceAgreementIdentifier: cuptAgreementId,
    })
);
console.log("Amendment of CUPT agreement request created. AgreementRequestId:
" + carAmendResponse.agreementRequestId);

const aarAmendResponse = await client.send(
    new AcceptAgreementRequestCommand({
        agreementRequestId: carAmendResponse.agreementRequestId,
    })
);
console.log("Amendment accepted. New AgreementId: " +
aarAmendResponse.agreementId);
}

amendAmiCUPTAgreement();

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Demonstrates how to create an AMI agreement with ConfigurableUpfrontPricingTerm
and then amend
the dimension quantity using the AWS Marketplace Agreement Service APIs.

Scenario: An AMI product with USAGE pricing requires two agreements:
    1. An agreement with usageBasedPricingTerm (UBPT) – accepted first to establish
    the base agreement.
    2. An agreement with configurableUpfrontPricingTerm (CUPT) – accepted after the
    UBPT agreement
    entitlements are active.
Once both agreement entitlements are available, this sample shows how to amend
the agreement
with configurableUpfrontPricingTerm (CUPT) to increase the dimension quantity.

Before running this sample, replace the placeholder constants below with values
from
your AWS Marketplace offer:
    - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
    - Term IDs (starting with term-) – found in the offer's term list.
    - SELECTOR_VALUE – duration for the agreement (e.g., P365D for one year).
    - DIMENSION_1_KEY – the dimension key defined in the offer (e.g., instance
    type).
    - DIMENSION_1_VALUE – initial quantity; NEW_DIMENSION_1_VALUE – amended
    quantity.
"""

import boto3

from utils.agreement_api_utils import (
    format_output,
    generate_client_token,

```

```

        poll_until_entitlements_available,
    )

class AmendAmiConfigurableUpfrontPricingTermForUsagePricingModel:

    # The agreementProposalId from the offer.
    AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>"

    # Term ID for the ConfigurableUpfrontPricingTerm in your offer.
    CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-term-id>"

    # Duration for the agreement (e.g., "P365D" for 365 days).
    SELECTOR_VALUE = "<your-selector-value>"

    # The dimension key defined in your offer (e.g., an EC2 instance type like
    "c6gn.medium").
    DIMENSION_1_KEY = "<your-dimension-key>"

    # Initial quantity for the dimension.
    DIMENSION_1_VALUE = 1

    # Term ID for the UsageBasedPricingTerm in your offer.
    USAGE_TERM_ID = "<your-usage-term-id>"

    # Term ID for the LegalTerm in your offer.
    LEGAL_TERM_ID = "<your-legal-term-id>"

    # Term ID for the ValidityTerm in your offer.
    VALIDITY_TERM_ID = "<your-validity-term-id>"

    # New quantity to use when amending the dimension of CUPT.
    NEW_DIMENSION_1_VALUE = 5

    @staticmethod
    def amend_ami_cupt_agreement():
        """
        Full end-to-end flow:
        1. Create and accept an agreement request with usageBasedPricingTerm.
        2. Wait for entitlements to become active, then create and accept an
        agreement request
           with configurableUpfrontPricingTerm (CUPT).

```

3. Wait for CUPT entitlements to become active, then amend the dimension quantity.

```

"""
client = boto3.client("marketplace-agreement")
cls = AmendAmiConfigurableUpfrontPricingTermForUsagePricingModel

usage_term = {"id": cls.USAGE_TERM_ID}
legal_term = {"id": cls.LEGAL_TERM_ID}
validity_term = {"id": cls.VALIDITY_TERM_ID}

# --- Agreement with UBPT ---
create_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="NEW",
    requestedTerms=[usage_term, legal_term, validity_term],
    agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
)
agreement_request_id = create_response["agreementRequestId"]
print("Agreement request with UBPT created. AgreementRequestId: " +
agreement_request_id)

accept_response = client.accept_agreement_request(
    agreementRequestId=agreement_request_id
)
agreement_id = accept_response["agreementId"]
print("Agreement request with UBPT accepted. AgreementId: " +
agreement_id)

# Wait for entitlements to become active before creating the agreement
with CUPT.
print("Waiting for UBPT agreement entitlements to become active...")
entitlements_response = poll_until_entitlements_available(client,
agreement_id)
print("UBPT agreement entitlements are now active.")
format_output(entitlements_response)

# --- Agreement with configurableUpfrontPricingTerm (CUPT) ---
configurable_upfront_pricing_term = {
    "id": cls.CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
    "configuration": {
        "configurableUpfrontPricingTermConfiguration": {
            "selectorValue": cls.SELECTOR_VALUE,
            "dimensions": [
                {

```

```

        "dimensionKey": cls.DIMENSION_1_KEY,
        "dimensionValue": cls.DIMENSION_1_VALUE,
    },
],
}
},
}

car_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="NEW",
    requestedTerms=[configurable_upfront_pricing_term, legal_term,
validity_term],
    agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
)
print("Agreement request with CUPT created. AgreementRequestId: " +
car_response["agreementRequestId"])

aar_response = client.accept_agreement_request(
    agreementRequestId=car_response["agreementRequestId"]
)
cupt_agreement_id = aar_response["agreementId"]
print("Agreement request with CUPT accepted. AgreementId: " +
cupt_agreement_id)

# Wait for entitlements to become active before amending.
print("Waiting for CUPT agreement entitlements to become active...")
cupt_entitlements_response = poll_until_entitlements_available(client,
cupt_agreement_id)
print("CUPT agreement entitlements are now active.")
format_output(cupt_entitlements_response)

# --- Amend Agreement with CUPT ---
# Increase the dimension quantity using Intent.AMEND and
sourceAgreementIdentifier.
new_config = {
    "id": cls.CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
    "configuration": {
        "configurableUpfrontPricingTermConfiguration": {
            "selectorValue": cls.SELECTOR_VALUE,
            "dimensions": [
                {
                    "dimensionKey": cls.DIMENSION_1_KEY,

```

```

        "dimensionValue": cls.NEW_DIMENSION_1_VALUE, #
        "Increase quantity for this dimension key":
            {
                "dimensionValue": cls.NEW_DIMENSION_1_VALUE,
            },
        ],
    },
}

car_amend_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="AMEND",
    requestedTerms=[new_config, legal_term, validity_term],
    sourceAgreementIdentifier=cupt_agreement_id,
)
print(
    "Amendment of CUPT agreement request created. AgreementRequestId: "
    + car_amend_response["agreementRequestId"]
)

aar_amend_response = client.accept_agreement_request(
    agreementRequestId=car_amend_response["agreementRequestId"]
)
print("Amendment accepted. New AgreementId: " +
    aar_amend_response["agreementId"])

if __name__ == "__main__":
    AmendAmiConfigurableUpfrontPricingTermForUsagePricingModel.amend_ami_cupt_agreement()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 새 AMI 무료 평가판 계약 생성

다음 코드 예제에서는 새 AMI 무료 평가판 계약을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to create an AMI Free Trial agreement
 * using the AWS Marketplace Agreement Service APIs.
 *
 * <p>Scenario: A buyer subscribes to an AMI product that offers a free trial
 * period.
 * The free trial includes a FreeTrialPricingTerm alongside a
 * UsageBasedPricingTerm.
 *
 * <p>Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
 * <ul>
```

```

*   <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
offer.</li>
*   <li>Term IDs (starting with {@code term-}) – found in the offer's term
list.</li>
* </ul>
*/
public class NewAmiFreeTrial {

    // The agreementProposalId from the offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-
proposal-identifier>";

    // Term ID for the FreeTrialPricingTerm in your offer.
    private static final String FREE_TRIAL_PRICING_TERM_ID = "<your-free-trial-
pricing-term-id>";

    // Term ID for the UsageBasedPricingTerm in your offer (applies after the
trial ends).
    private static final String USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-
pricing-term-id>";

    // Term ID for the SupportTerm in your offer.
    private static final String SUPPORT_TERM_ID = "<your-support-term-id>";

    // Term ID for the LegalTerm in your offer.
    private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

    public static void main(String[] args) {
        createAndAcceptAmiFreeTrialAgreementRequest();
    }

    /**
     * Creates an AMI Free Trial agreement.
     * The FreeTrialPricingTerm grants access at no cost for the trial period.
     * The UsageBasedPricingTerm defines the charges that apply once the trial
ends.
     */
    private static void createAndAcceptAmiFreeTrialAgreementRequest() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
    }

```

```

        RequestedTerm freeTrialPricingTerm = RequestedTerm.builder()
            .id(FREE_TRIAL_PRICING_TERM_ID)
            .build();
        RequestedTerm usageBasedPricingTerm = RequestedTerm.builder()
            .id(USAGE_BASED_PRICING_TERM_ID)
            .build();
        RequestedTerm supportTerm = RequestedTerm.builder()
            .id(SUPPORT_TERM_ID)
            .build();
        RequestedTerm legalTerm = RequestedTerm.builder()
            .id(LEGAL_TERM_ID)
            .build();

        CreateAgreementRequestRequest createAgreementRequestRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.NEW)
                .requestedTerms(freeTrialPricingTerm,
usageBasedPricingTerm, supportTerm, legalTerm)

                .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
                .build();
        CreateAgreementRequestResponse createAgreementRequestResponse =

marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
        System.out.println("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId());

        AcceptAgreementRequestRequest acceptAgreementRequestRequest =
            AcceptAgreementRequestRequest.builder()

                .agreementRequestId(createAgreementRequestResponse.agreementRequestId())
                .build();
        AcceptAgreementRequestResponse acceptAgreementRequestResponse =

marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
        System.out.println("Agreement request with freeTrialPricingTerm accepted.
AgreementId: " + acceptAgreementRequestResponse.agreementId());
    }
}

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
    MarketplaceAgreementClient,
    CreateAgreementRequestCommand,
    AcceptAgreementRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken } = require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to create an AMI Free Trial agreement
 * using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: A buyer subscribes to an AMI product that offers a free trial
 * period.
 * The free trial includes a FreeTrialPricingTerm alongside a
 * UsageBasedPricingTerm.
 *
 * Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
 * - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
 * - Term IDs (starting with "term-") – found in the offer's term list.
 */

// The agreementProposalId from the offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>";

// Term ID for the FreeTrialPricingTerm in your offer.
const FREE_TRIAL_PRICING_TERM_ID = "<your-free-trial-pricing-term-id>";

// Term ID for the UsageBasedPricingTerm in your offer (applies after the trial
// ends).
const USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-pricing-term-id>";
```

```

// Term ID for the SupportTerm in your offer.
const SUPPORT_TERM_ID = "<your-support-term-id>";

// Term ID for the LegalTerm in your offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

/**
 * Creates an AMI Free Trial agreement.
 * The FreeTrialPricingTerm grants access at no cost for the trial period.
 * The UsageBasedPricingTerm defines the charges that apply once the trial ends.
 */
async function createAndAcceptAmiFreeTrialAgreementRequest() {
    const client = new MarketplaceAgreementClient();

    const freeTrialPricingTerm = { id: FREE_TRIAL_PRICING_TERM_ID };
    const usageBasedPricingTerm = { id: USAGE_BASED_PRICING_TERM_ID };
    const supportTerm = { id: SUPPORT_TERM_ID };
    const legalTerm = { id: LEGAL_TERM_ID };

    const createAgreementRequestResponse = await client.send(
        new CreateAgreementRequestCommand({
            clientToken: generateClientToken(),
            intent: "NEW",
            requestedTerms: [freeTrialPricingTerm, usageBasedPricingTerm,
supportTerm, legalTerm],
            agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
        })
    );
    console.log("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId);

    const acceptAgreementRequestResponse = await client.send(
        new AcceptAgreementRequestCommand({
            agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
        })
    );
    console.log("Agreement request with freeTrialPricingTerm accepted.
AgreementId: " + acceptAgreementRequestResponse.agreementId);
}

createAndAcceptAmiFreeTrialAgreementRequest();

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Demonstrates how to create an AMI Free Trial agreement
using the AWS Marketplace Agreement Service APIs.

Scenario: A buyer subscribes to an AMI product that offers a free trial period.
The free trial includes a FreeTrialPricingTerm alongside a UsageBasedPricingTerm.

Before running this sample, replace the placeholder constants below with values
from
your AWS Marketplace offer:
    - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
    - Term IDs (starting with term-) – found in the offer's term list.
"""

import boto3

from utils.agreement_api_utils import generate_client_token

class NewAmiFreeTrial:

    # The agreementProposalId from the offer.
    AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>"

    # Term ID for the FreeTrialPricingTerm in your offer.
    FREE_TRIAL_PRICING_TERM_ID = "<your-free-trial-pricing-term-id>"

```

```

    # Term ID for the UsageBasedPricingTerm in your offer (applies after the
    trial ends).
    USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-pricing-term-id>"

    # Term ID for the SupportTerm in your offer.
    SUPPORT_TERM_ID = "<your-support-term-id>"

    # Term ID for the LegalTerm in your offer.
    LEGAL_TERM_ID = "<your-legal-term-id>"

    @staticmethod
    def create_and_accept_ami_free_trial_agreement_request():
        """
        Create an AMI Free Trial agreement.

        The FreeTrialPricingTerm grants access at no cost for the trial period.
        The UsageBasedPricingTerm defines the charges that apply once the trial
ends.
        """
        client = boto3.client("marketplace-agreement")

        create_response = client.create_agreement_request(
            clientToken=generate_client_token(),
            intent="NEW",
            requestedTerms=[
                {"id": NewAmiFreeTrial.FREE_TRIAL_PRICING_TERM_ID},
                {"id": NewAmiFreeTrial.USAGE_BASED_PRICING_TERM_ID},
                {"id": NewAmiFreeTrial.SUPPORT_TERM_ID},
                {"id": NewAmiFreeTrial.LEGAL_TERM_ID},
            ],
            agreementProposalIdentifier=NewAmiFreeTrial.AGREEMENT_PROPOSAL_IDENTIFIER,
        )
        agreement_request_id = create_response["agreementRequestId"]
        print("Agreement request created. AgreementRequestId: " +
              agreement_request_id)

        accept_response = client.accept_agreement_request(
            agreementRequestId=agreement_request_id
        )
        print(
            "Agreement request with freeTrialPricingTerm accepted. AgreementId: "
            + accept_response["agreementId"]

```

```
)

if __name__ == "__main__":
    NewAmiFreeTrial.create_and_accept_ami_free_trial_agreement_request()
```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 선결제로 새 SaaS 계약 생성

다음 코드 예제에서는 선결제와 함께 새 SaaS 계약 계약을 생성하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse
import
    software.amazon.awssdk.services.marketplaceagreement.model.ConfigurableUpfrontPricingTer
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest
```

```

import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Dimension;
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RequestedTermConfiguration;
import
    software.amazon.awssdk.services.marketplaceagreement.model.TaxConfiguration;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to create a SaaS agreement with CONTRACT pricing model with
 * upfront payment
 * using the AWS Marketplace Agreement Service APIs.
 *
 * <p>Scenario: A buyer subscribes to a SaaS product using a
 * ConfigurableUpfrontPricingTerm,
 * selecting an agreement duration and specifying quantities for multiple
 * dimensions.
 * Tax estimation is enabled at the time of agreement creation.
 *
 * <p>Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
 * <ul>
 * <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
 * offer.</li>
 * <li>Term IDs (starting with {@code term-}) – found in the offer's term
 * list.</li>
 * <li>{@code SELECTOR_VALUE} – duration for the agreement (e.g., {@code P12M}
 * for 12 months).</li>
 * <li>{@code DIMENSION_1_KEY}, {@code DIMENSION_2_KEY} – dimension keys
 * defined in the offer.</li>
 * <li>{@code DIMENSION_1_VALUE}, {@code DIMENSION_2_VALUE} – quantities for
 * each dimension.</li>
 * </ul>
 */
public class NewSaaSContractWithUpfrontPayment {

    // The agreementProposalId from the offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-
    proposal-identifier>";

```

```

// Term ID for the ConfigurableUpfrontPricingTerm in your offer.
private static final String CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-
configurable-upfront-pricing-term-id>";

// Duration for the agreement (e.g., "P12M" for 12 months).
private static final String SELECTOR_VALUE = "<your-selector-value>";

// First dimension key and quantity defined in your offer.
private static final String DIMENSION_1_KEY = "<your-dimension-1-key>";
private static final int DIMENSION_1_VALUE = 10;

// Second dimension key and quantity defined in your offer.
private static final String DIMENSION_2_KEY = "<your-dimension-2-key>";
private static final int DIMENSION_2_VALUE = 20;

// Term ID for the LegalTerm in your offer.
private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

// Tax estimation setting: "ENABLED" to include estimated taxes in the
agreement.
private static final String TAX_ESTIMATION = "ENABLED";

public static void main(String[] args) {
    createSaaSContractAgreement();
}

/**
 * Creates a SaaS agreement with CONTRACT pricing model with configurable
upfront pricing
 * for multiple dimensions and tax estimation.
 */
private static void createSaaSContractAgreement() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    RequestedTerm configurableUpfrontPricingTerm = RequestedTerm.builder()
        .id(CONFIGURABLE_UPFRONT_PRICING_TERM_ID)

        .configuration(RequestedTermConfiguration.fromConfigurableUpfrontPricingTermConfiguration(
            ConfigurableUpfrontPricingTermConfiguration.builder()
                .selectorValue(SELECTOR_VALUE)

```

```

        .dimensions(Dimension.builder()

        .dimensionKey(DIMENSION_1_KEY)

        .dimensionValue(DIMENSION_1_VALUE)

        .build(),
        Dimension.builder()

        .dimensionKey(DIMENSION_2_KEY)

        .dimensionValue(DIMENSION_2_VALUE)

        .build())

        .build()))

        .build();

    RequestedTerm legalTerm = RequestedTerm.builder()
        .id(LEGAL_TERM_ID)
        .build();

    CreateAgreementRequestRequest createAgreementRequestRequest =
        CreateAgreementRequestRequest.builder()
            .clientToken(AgreementApiUtils.generateClientToken())
            .intent(Intent.NEW)
            .requestedTerms(configurableUpfrontPricingTerm,
legalTerm)

        .taxConfiguration(TaxConfiguration.builder().taxEstimation(TAX_ESTIMATION).build())

        .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
            .build();

    CreateAgreementRequestResponse createAgreementRequestResponse =

marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
    System.out.println("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId());

    AcceptAgreementRequestRequest acceptAgreementRequestRequest =
        AcceptAgreementRequestRequest.builder()

        .agreementRequestId(createAgreementRequestResponse.agreementRequestId())
            .build();

    AcceptAgreementRequestResponse acceptAgreementRequestResponse =

marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);

```



```

        System.out.println("SaaS agreement request with CONTRACT pricing model
        accepted. AgreementId: " + acceptAgreementRequestResponse.agreementId());
    }
}

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

const {
    MarketplaceAgreementClient,
    CreateAgreementRequestCommand,
    AcceptAgreementRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken } = require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to create a SaaS agreement with CONTRACT pricing model with
 * upfront payment
 * using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: A buyer subscribes to a SaaS product using a
 * ConfigurableUpfrontPricingTerm,
 * selecting an agreement duration and specifying quantities for multiple
 * dimensions.
 * Tax estimation is enabled at the time of agreement creation.
 *
 * Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
 * - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
 * - Term IDs (starting with "term-") – found in the offer's term list.
 * - SELECTOR_VALUE – duration for the agreement (e.g., "P12M" for 12 months).

```

```

*   - DIMENSION_1_KEY, DIMENSION_2_KEY – dimension keys defined in the offer.
*   - DIMENSION_1_VALUE, DIMENSION_2_VALUE – quantities for each dimension.
*/

// The agreementProposalId from the offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>";

// Term ID for the ConfigurableUpfrontPricingTerm in your offer.
const CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-term-id>";

// Duration for the agreement (e.g., "P12M" for 12 months).
const SELECTOR_VALUE = "<your-selector-value>";

// First dimension key and quantity defined in your offer.
const DIMENSION_1_KEY = "<your-dimension-1-key>";
const DIMENSION_1_VALUE = 10;

// Second dimension key and quantity defined in your offer.
const DIMENSION_2_KEY = "<your-dimension-2-key>";
const DIMENSION_2_VALUE = 20;

// Term ID for the LegalTerm in your offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// Tax estimation setting: "ENABLED" to include estimated taxes in the agreement.
const TAX_ESTIMATION = "ENABLED";

/**
 * Creates a SaaS agreement with CONTRACT pricing model with configurable upfront
 * pricing
 * for multiple dimensions and tax estimation.
 */
async function createSaaSContractAgreement() {
    const client = new MarketplaceAgreementClient();

    const configurableUpfrontPricingTerm = {
        id: CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        configuration: {
            configurableUpfrontPricingTermConfiguration: {
                selectorValue: SELECTOR_VALUE,
                dimensions: [
                    { dimensionKey: DIMENSION_1_KEY, dimensionValue:
DIMENSION_1_VALUE },

```

```

        { dimensionKey: DIMENSION_2_KEY, dimensionValue:
    DIMENSION_2_VALUE },
        ],
    },
    },
};

const legalTerm = { id: LEGAL_TERM_ID };

const createAgreementRequestResponse = await client.send(
    new CreateAgreementRequestCommand({
        clientToken: generateClientToken(),
        intent: "NEW",
        requestedTerms: [configurableUpfrontPricingTerm, legalTerm],
        taxConfiguration: { taxEstimation: TAX_ESTIMATION },
        agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
    })
);
console.log("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId);

const acceptAgreementRequestResponse = await client.send(
    new AcceptAgreementRequestCommand({
        agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
    })
);
console.log("SaaS agreement request with CONTRACT pricing model accepted.
AgreementId: " + acceptAgreementRequestResponse.agreementId);
}

createSaaSContractAgreement();

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Demonstrates how to create a SaaS agreement with CONTRACT pricing model with
upfront payment
using the AWS Marketplace Agreement Service APIs.

Scenario: A buyer subscribes to a SaaS product using a
ConfigurableUpfrontPricingTerm,
selecting an agreement duration and specifying quantities for multiple
dimensions.
Tax estimation is enabled at the time of agreement creation.

Before running this sample, replace the placeholder constants below with values
from
your AWS Marketplace offer:
- AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
- Term IDs (starting with term-) – found in the offer's term list.
- SELECTOR_VALUE – duration for the agreement (e.g., P12M for 12 months).
- DIMENSION_1_KEY, DIMENSION_2_KEY – dimension keys defined in the offer.
- DIMENSION_1_VALUE, DIMENSION_2_VALUE – quantities for each dimension.
"""

import boto3

from utils.agreement_api_utils import generate_client_token

class NewSaaSContractWithUpfrontPayment:

    # The agreementProposalId from the offer.
    AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>"

    # Term ID for the ConfigurableUpfrontPricingTerm in your offer.

```

```

CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-
term-id>"

# Duration for the agreement (e.g., "P12M" for 12 months).
SELECTOR_VALUE = "<your-selector-value>"

# First dimension key and quantity defined in your offer.
DIMENSION_1_KEY = "<your-dimension-1-key>"
DIMENSION_1_VALUE = 10

# Second dimension key and quantity defined in your offer.
DIMENSION_2_KEY = "<your-dimension-2-key>"
DIMENSION_2_VALUE = 20

# Term ID for the LegalTerm in your offer.
LEGAL_TERM_ID = "<your-legal-term-id>"

# Tax estimation setting: "ENABLED" to include estimated taxes in the
agreement.
TAX_ESTIMATION = "ENABLED"

@staticmethod
def create_saas_contract_agreement():
    """
    Create a SaaS agreement with CONTRACT pricing model with configurable
    upfront pricing
    for multiple dimensions and tax estimation.
    """
    client = boto3.client("marketplace-agreement")

    configurable_upfront_pricing_term = {
        "id":
NewSaaSContractWithUpfrontPayment.CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        "configuration": {
            "configurableUpfrontPricingTermConfiguration": {
                "selectorValue":
NewSaaSContractWithUpfrontPayment.SELECTOR_VALUE,
                "dimensions": [
                    {
                        "dimensionKey":
NewSaaSContractWithUpfrontPayment.DIMENSION_1_KEY,
                        "dimensionValue":
NewSaaSContractWithUpfrontPayment.DIMENSION_1_VALUE,
                    },

```

```

        {
            "dimensionKey":
NewSaaSContractWithUpfrontPayment.DIMENSION_2_KEY,
            "dimensionValue":
NewSaaSContractWithUpfrontPayment.DIMENSION_2_VALUE,
        },
    ],
}
},
}

legal_term = {"id": NewSaaSContractWithUpfrontPayment.LEGAL_TERM_ID}

create_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="NEW",
    requestedTerms=[configurable_upfront_pricing_term, legal_term],
    taxConfiguration={"taxEstimation":
NewSaaSContractWithUpfrontPayment.TAX_ESTIMATION},

agreementProposalIdentifier=NewSaaSContractWithUpfrontPayment.AGREEMENT_PROPOSAL_IDENTIFIER
)
agreement_request_id = create_response["agreementRequestId"]
print("Agreement request created. AgreementRequestId: " +
agreement_request_id)

accept_response = client.accept_agreement_request(
    agreementRequestId=agreement_request_id
)
print(
    "SaaS agreement request with CONTRACT pricing model accepted.
AgreementId: "
    + accept_response["agreementId"]
)

if __name__ == "__main__":
    NewSaaSContractWithUpfrontPayment.create_saas_contract_agreement()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 모든 계약 IDs 가져오기

다음 코드 예제에서는 모든 계약 ID를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AgreementViewSummary;
import software.amazon.awssdk.services.marketplaceagreement.model.Filter;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsResponse;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAllAgreementsIds {

    /*
     * Get all purchase agreements ids for your PartyType (Proposer, or Acceptor)
```

```
* Depend on the number of agreements in your account, this code may take some
time to finish.
*/
public static void main(String[] args) {

    List<String> agreementIds = getAllAgreementIds();

    ReferenceCodesUtils.formatOutput(agreementIds);

}

public static List<String> getAllAgreementIds() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    // Set PartyType filter to PARTY_TYPE_FILTER_VALUE_PROPOSER to return
    agreements where you are the proposer.
    // Change to PARTY_TYPE_FILTER_VALUE_ACCEPTOR to return agreements where you
    are the acceptor.
    Filter partyType = Filter.builder().name(PARTY_TYPE_FILTER_NAME)
        .values(PARTY_TYPE_FILTER_VALUE_PROPOSER).build();

    Filter agreementType = Filter.builder().name(AGREEMENT_TYPE_FILTER_NAME)
        .values(AGREEMENT_TYPE_FILTER_VALUE_PURCHASEAGREEMENT).build();

    List<Filter> searchFilters = new ArrayList<Filter>();

    searchFilters.addAll(Arrays.asList(partyType, agreementType));

    // Save all results in a list array
    List<AgreementViewSummary> agreementSummaryList = new
    ArrayList<AgreementViewSummary>();

    SearchAgreementsRequest searchAgreementsRequest =
        SearchAgreementsRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .filters(searchFilters)
            .build();

    SearchAgreementsResponse searchAgreementsResponse =
    marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);
```



```

    agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());

    while (searchAgreementsResponse.nextToken() != null &&
searchAgreementsResponse.nextToken().length() > 0) {
        searchAgreementsRequest =
            SearchAgreementsRequest.builder()
                .catalog(AWS_MP_CATALOG)
                .nextToken(searchAgreementsResponse.nextToken())
                .filters(searchFilters)
                .build();
        searchAgreementsResponse =
marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

        agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());
    }

    List<String> agreementIds = new ArrayList<String>();
    for (AgreementViewSummary summary : agreementSummaryList) {
        agreementIds.add(summary.agreementId());
    }
    return agreementIds;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [SearchAgreements](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""

```

Purpose

Shows how to use the AWS SDK for Python (Boto3) to get all agreement ids

AG-09

"""

```
import logging
```

```
import boto3
```

```
from botocore.exceptions import ClientError
```

```
mp_client = boto3.client("marketplace-agreement")
```

```
logger = logging.getLogger(__name__)
```

```
MAX_PAGE_RESULTS = 10
```

```
def get_agreements():
```

```
    AgreementSummaryList = []
```

```
    agreement_id_list = []
```

```
    try:
```

```
        agreements = mp_client.search_agreements(
```

```
            catalog="AWSMarketplace",
```

```
            maxResults=MAX_PAGE_RESULTS,
```

```
            # Set PartyType filter to "Proposer" to return agreements where you  
            are the proposer.
```

```
            # Change to "Acceptor" to return agreements where you are the  
            acceptor.
```

```
            filters=[
```

```
                {"name": "PartyType", "values": ["Proposer"]},
```

```
                {"name": "AgreementType", "values": ["PurchaseAgreement"]},
```

```
            ],
```

```
        )
```

```
    except ClientError as e:
```

```
        logger.error("Could not complete search_agreements request.")
```

```
        raise
```

```
    AgreementSummaryList.extend(agreements["agreementViewSummaries"])
```

```
    while "nextToken" in agreements and agreements["nextToken"] is not None:
```

```
        try:
```

```
            agreements = mp_client.search_agreements(
```

```
                catalog="AWSMarketplace",
```

```

        maxResults=MAX_PAGE_RESULTS,
        nextToken=agreements["nextToken"],
        filters=[
            {"name": "PartyType", "values": ["Proposer"]},
            {"name": "AgreementType", "values": ["PurchaseAgreement"]},
        ],
    )
except ClientError as e:
    logger.error("Could not complete search_agreements request.")
    raise

AgreementSummaryList.extend(agreements["agreementViewSummaries"])

for agreement in AgreementSummaryList:
    agreement_id_list.append(agreement["agreementId"])

return agreement_id_list

if __name__ == "__main__":
    agreement_id_list = get_agreements()

    print(agreement_id_list)

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 모든 계약 가져오기

다음 코드 예제에서는 모든 계약을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AgreementViewSummary;
import software.amazon.awssdk.services.marketplaceagreement.model.Filter;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsResponse;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;

import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAllAgreements {

    /*
     * Get all purchase agreements for your PartyType (Proposer, or Acceptor)
     * Depend on the number of agreements in your account, this code may take some
     * time to finish.
     */
    public static void main(String[] args) {

        List<AgreementViewSummary> agreementSummaryList = getAllAgreements();
    }
}
```

```
ReferenceCodesUtils.formatOutput(agreementSummaryList);
}

public static List<AgreementViewSummary> getAllAgreements() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    // Set PartyType filter to PARTY_TYPE_FILTER_VALUE_ACCEPTOR to return
    agreements where you are the acceptor.
    // Change to PARTY_TYPE_FILTER_VALUE_PROPOSER to return agreements where
    you are the proposer.
    Filter partyType = Filter.builder().name(PARTY_TYPE_FILTER_NAME)
        .values(PARTY_TYPE_FILTER_VALUE_ACCEPTOR).build();

    Filter agreementType = Filter.builder().name(AGREEMENT_TYPE_FILTER_NAME)
        .values(AGREEMENT_TYPE_FILTER_VALUE_PURCHASEAGREEMENT).build();

    List<Filter> searchFilters = new ArrayList<Filter>();

    searchFilters.addAll(Arrays.asList(partyType, agreementType));

    // Save all results in a list array

    List<AgreementViewSummary> agreementSummaryList = new
    ArrayList<AgreementViewSummary>();

    SearchAgreementsRequest searchAgreementsRequest =
        SearchAgreementsRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .filters(searchFilters)
            .build();

    SearchAgreementsResponse searchAgreementsResponse =
    marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

    agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());

    while (searchAgreementsResponse.nextToken() != null &&
    searchAgreementsResponse.nextToken().length() > 0) {
        searchAgreementsRequest =
```

```

        SearchAgreementsRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .nextToken(searchAgreementsResponse.nextToken())
            .filters(searchFilters).build();
        searchAgreementsResponse =
            marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

        agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());
    }
    return agreementSummaryList;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [SearchAgreements](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to get all agreements
AG-01
"""

import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

```

```
import utils.helpers as helper
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-agreement")

logger = logging.getLogger(__name__)

MAX_PAGE_RESULTS = 10

party_type_list = ["Acceptor"]
agreement_type_list = ["PurchaseAgreement"]

filter_list = [
    {"name": "PartyType", "values": party_type_list},
    {"name": "AgreementType", "values": agreement_type_list},
]

agreement_results_list = []

def get_agreements(filter_list=filter_list):
    try:
        agreements = mp_client.search_agreements(
            catalog="AWSMarketplace",
            maxResults=MAX_PAGE_RESULTS,
            filters=filter_list,
        )
    except ClientError as e:
        logger.error("Could not complete search_agreements request.")
        raise e

    agreement_results_list.extend(agreements["agreementViewSummaries"])

    while "nextToken" in agreements and agreements["nextToken"] is not None:
        try:
            agreements = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                nextToken=agreements["nextToken"],
                filters=filter_list,
            )
        except ClientError as e:
            logger.error("Could not complete search_agreements request.")
```

```

        raise e

    agreement_results_list.extend(agreements["agreementViewSummaries"])

    return agreement_results_list

if __name__ == "__main__":
    agreements_list = get_agreements(filter_list)
    helper.pretty_print_datetime(agreements_list)

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에서 고객 ID 가져오기

다음 코드 예제에서는 계약에서 고객 ID를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.seller;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementRequest;

```



```
import
software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementResponse;

public class GetAgreementCustomerInfo {

    /*
     * Obtain metadata about the customer who created the agreement, such as the
     * customer's AWS Account ID
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        DescribeAgreementResponse describeAgreementResponse =
getDescribeAgreementResponse(agreementId);

        System.out.println("Customer's AWS Account ID is " +
describeAgreementResponse.acceptor().accountId());

    }

    public static DescribeAgreementResponse getDescribeAgreementResponse(String
agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        DescribeAgreementRequest describeAgreementRequest =
            DescribeAgreementRequest.builder()
                .agreementId(agreementId)
                .build();

        DescribeAgreementResponse describeAgreementResponse =
marketplaceAgreementClient.describeAgreement(describeAgreementRequest);
        return describeAgreementResponse;
    }
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to get customer AWS account id
from a given agreement
AG-08
"""

import argparse
import logging

import boto3
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-agreement")

logger = logging.getLogger(__name__)

def get_agreement_information(agreement_id):
    try:
        response = mp_client.describe_agreement(agreementId=agreement_id)
    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", agreement_id)
            raise e
        else:
            logger.error("Unexpected error: %s", e)
            raise e

    return response
```

```

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--agreement-id",
        "-aid",
        help="Provide agreement ID to describe agreement status",
        required=True,
    )
    args = parser.parse_args()

    response = get_agreement_information(agreement_id=args.agreement_id)

    print(f"Customer account: {response['acceptor']['accountId']}")

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 취소 요청 세부 정보 가져오기

다음 코드 예제에서는 특정 계약 취소 요청에 대한 세부 정보를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.buyer.agreementCancellation;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;

```

```
import
software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementCancellationRequest;
import
software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementCancellationRequestResponse;

public class GetAgreementCancellationRequest {

    private static final String AGREEMENT_ID = "<AGREEMENT ID HERE>";
    private static final String AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT
CANCELLATION REQUEST ID HERE>";

    public static void main(String[] args) {
        getAgreementCancellationRequest();
    }

    private static void getAgreementCancellationRequest() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementCancellationRequest request =
            GetAgreementCancellationRequest.builder()
                .agreementId(AGREEMENT_ID)

                .agreementCancellationRequestId(AGREEMENT_CANCELLATION_REQUEST_ID)
                .build();

        GetAgreementCancellationRequestResponse response =
            marketplaceAgreementClient.getAgreementCancellationRequest(request);

        System.out.println("Agreement ID: " + response.agreementId());
        System.out.println("Cancellation Request ID: " +
            response.agreementCancellationRequestId());
        System.out.println("Status: " + response.statusAsString());
        System.out.println("Status Message: " + response.statusMessage());
        System.out.println("Reason Code: " + response.reasonCodeAsString());
        System.out.println("Created At: " + response.createdAt());
        System.out.println("Updated At: " + response.updatedAt());
    }
}
```

```
}

```

- API 세부 정보는 API 참조의 [GetAgreementCancellationRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
  MarketplaceAgreementClient,
  GetAgreementCancellationRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const AGREEMENT_ID = "<AGREEMENT ID HERE>";
const AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT CANCELLATION REQUEST ID HERE>";

async function getAgreementCancellationRequest() {
  const client = new MarketplaceAgreementClient();

  const response = await client.send(
    new GetAgreementCancellationRequestCommand({
      agreementId: AGREEMENT_ID,
      agreementCancellationRequestId: AGREEMENT_CANCELLATION_REQUEST_ID,
    })
  );

  console.log("Agreement ID: " + response.agreementId);
  console.log("Cancellation Request ID: " +
response.agreementCancellationRequestId);
  console.log("Status: " + response.status);
  console.log("Status Message: " + response.statusMessage);
  console.log("Reason Code: " + response.reasonCode);
}
```

```

    console.log("Created At: " + response.createdAt);
    console.log("Updated At: " + response.updatedAt);
}

getAgreementCancellationRequest();

```

- API 세부 정보는 API 참조의 [GetAgreementCancellationRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class GetAgreementCancellationRequest:

    AGREEMENT_ID = "<AGREEMENT ID HERE>"
    AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT CANCELLATION REQUEST ID HERE>"

    @staticmethod
    def get_agreement_cancellation_request():
        client = boto3.client("marketplace-agreement")

        response = client.get_agreement_cancellation_request(
            agreementId=GetAgreementCancellationRequest.AGREEMENT_ID,

```

```

agreementCancellationRequestId=GetAgreementCancellationRequest.AGREEMENT_CANCELLATION_REQUEST_ID
    )

    print("Agreement ID: " + response["agreementId"])
    print("Cancellation Request ID: " +
response["agreementCancellationRequestId"])
    print("Status: " + str(response.get("status", "")))
    print("Status Message: " + str(response.get("statusMessage", "")))
    print("Reason Code: " + str(response.get("reasonCode", "")))
    print("Created At: " + str(response.get("createdAt", "")))
    print("Updated At: " + str(response.get("updatedAt", "")))

if __name__ == "__main__":
    GetAgreementCancellationRequest.get_agreement_cancellation_request()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [GetAgreementCancellationRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 결제 요청 세부 정보 가져오기

다음 코드 예제에서는 특정 계약 결제 요청에 대한 세부 정보를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.buyer.paymentRequest;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;

```

```

import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementPaymentRequestRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementPaymentRequestResponse;

public class GetAgreementPaymentRequest {

    private static final String AGREEMENT_ID = "<AGREEMENT ID HERE>";
    private static final String PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>";

    public static void main(String[] args) {
        getAgreementPaymentRequest();
    }

    private static void getAgreementPaymentRequest() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementPaymentRequestRequest request =
            GetAgreementPaymentRequestRequest.builder()
                .agreementId(AGREEMENT_ID)
                .paymentRequestId(PAYMENT_REQUEST_ID)
                .build();

        GetAgreementPaymentRequestResponse response =
            marketplaceAgreementClient.getAgreementPaymentRequest(request);

        System.out.println("Payment Request ID: " + response.paymentRequestId());
        System.out.println("Agreement ID: " + response.agreementId());
        System.out.println("Status: " + response.statusAsString());
        System.out.println("Status Message: " + response.statusMessage());
        System.out.println("Name: " + response.name());
        System.out.println("Charge ID: " + response.chargeId());
        System.out.println("Charge Amount: " + response.chargeAmount());
        System.out.println("Currency Code: " + response.currencyCode());
        System.out.println("Created At: " + response.createdAt());
        System.out.println("Updated At: " + response.updatedAt());
    }
}

```



```
}

```

- API 세부 정보는 API 참조의 [GetAgreementPaymentRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
  MarketplaceAgreementClient,
  GetAgreementPaymentRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const AGREEMENT_ID = "<AGREEMENT ID HERE>";
const PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>";

async function getAgreementPaymentRequest() {
  const client = new MarketplaceAgreementClient();

  const response = await client.send(
    new GetAgreementPaymentRequestCommand({
      agreementId: AGREEMENT_ID,
      paymentRequestId: PAYMENT_REQUEST_ID,
    })
  );

  console.log("Payment Request ID: " + response.paymentRequestId);
  console.log("Agreement ID: " + response.agreementId);
  console.log("Status: " + response.status);
  console.log("Status Message: " + response.statusMessage);
  console.log("Name: " + response.name);
  console.log("Charge ID: " + response.chargeId);
  console.log("Charge Amount: " + response.chargeAmount);
}
```

```

    console.log("Currency Code: " + response.currencyCode);
    console.log("Created At: " + response.createdAt);
    console.log("Updated At: " + response.updatedAt);
}

getAgreementPaymentRequest();

```

- API 세부 정보는 API 참조의 [GetAgreementPaymentRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class GetAgreementPaymentRequest:

    AGREEMENT_ID = "<AGREEMENT ID HERE>"
    PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>"

    @staticmethod
    def get_agreement_payment_request():
        client = boto3.client("marketplace-agreement")

        response = client.get_agreement_payment_request(
            agreementId=GetAgreementPaymentRequest.AGREEMENT_ID,
            paymentRequestId=GetAgreementPaymentRequest.PAYMENT_REQUEST_ID,

```

```

    )

    print("Payment Request ID: " + response["paymentRequestId"])
    print("Agreement ID: " + response["agreementId"])
    print("Status: " + str(response.get("status", "")))
    print("Status Message: " + str(response.get("statusMessage", "")))
    print("Name: " + str(response.get("name", "")))
    print("Charge ID: " + str(response.get("chargeId", "")))
    print("Charge Amount: " + str(response.get("chargeAmount", "")))
    print("Currency Code: " + str(response.get("currencyCode", "")))
    print("Created At: " + str(response.get("createdAt", "")))
    print("Updated At: " + str(response.get("updatedAt", "")))

if __name__ == "__main__":
    GetAgreementPaymentRequest.get_agreement_payment_request()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [GetAgreementPaymentRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에서 재무 세부 정보 가져오기

다음 코드 예제에서는 계약에서 재무 세부 정보를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.seller;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;

```

```

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementResponse;

public class GetAgreementFinancialDetails {

    /**
     * Obtain financial details, such as Total Contract Value of the agreement from
     * a given agreement
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        String totalContractValue = getTotalContractValue(agreementId);

        System.out.println("Total Contract Value is " + totalContractValue);

    }

    public static String getTotalContractValue(String agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        DescribeAgreementRequest describeAgreementRequest =
            DescribeAgreementRequest.builder()
                .agreementId(agreementId)
                .build();

        DescribeAgreementResponse describeAgreementResponse =
            marketplaceAgreementClient.describeAgreement(describeAgreementRequest);

        String totalContractValue = "N/A";

        if ( describeAgreementResponse.estimatedCharges() != null ) {

```

```

    totalContractValue =
describeAgreementResponse.estimatedCharges().agreementValue()
    + " "
    + describeAgreementResponse.estimatedCharges().currencyCode();
}
return totalContractValue;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain financial details, such as Total Contract Value of the agreement from a
given agreement
AG-14

Example Usage: python3 get_agreement_financial_details.py --agreement-id
<agreement-id>
"""

import argparse
import logging

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

```

```

mp_client = boto3.client("marketplace-agreement")

def get_agreement_information(agreement_id):
    try:
        agreement = mp_client.describe_agreement(agreementId=agreement_id)

        return agreement

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", agreement_id)
        else:
            logger.error("Unexpected error: %s", e)

    return None

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--agreement-id",
        "-aid",
        help="Provide agreement ID to describe agreement status",
        required=True,
    )
    args = parser.parse_args()

    agreement = get_agreement_information(args.agreement_id)

    if agreement is not None:
        print(f"Agreement Id: {args.agreement_id}")
        print(
            f"Agreement Value: {agreement['estimatedCharges']['currencyCode']}
            {agreement['estimatedCharges']['agreementValue']}"
        )

    else:
        print(f"Agreement with ID {args.agreement_id} is not found")

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#) [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에서 무료 평가판 세부 정보 가져오기

다음 코드 예제에서는 계약에서 무료 평가판 세부 정보를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.FreeTrialPricingTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;

import static
    com.example.awsmarketplace.utils.ReferenceCodesConstants.AGREEMENT_ID;

import java.util.ArrayList;
import java.util.List;

import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementTermsFreeTrialDetails {

    /*
```

```

    * Obtain the details from an agreement of a free trial I have provided to the
    customer
    */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        List<FreeTrialPricingTerm> freeTrialPricingTerms =
            getFreeTrialPricingTerms(agreementId);

        ReferenceCodesUtils.formatOutput(freeTrialPricingTerms);
    }

    public static List<FreeTrialPricingTerm> getFreeTrialPricingTerms(String
        agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementTermsRequest getAgreementTermsRequest =
            GetAgreementTermsRequest.builder().agreementId(agreementId)
                .build();

        GetAgreementTermsResponse getAgreementTermsResponse =
            marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);

        List<FreeTrialPricingTerm> freeTrialPricingTerms = new
            ArrayList<FreeTrialPricingTerm>();

        for (AcceptedTerm acceptedTerm : getAgreementTermsResponse.acceptedTerms()) {
            if (acceptedTerm.freeTrialPricingTerm() != null) {
                freeTrialPricingTerms.add(acceptedTerm.freeTrialPricingTerm());
            }
        }
        return freeTrialPricingTerms;
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain the details from an agreement of a free trial I have provided to the
customer
AG-20

Example Usage: python3 get_agreement_free_trial_details.py --agreement-id
<agreement-id>
"""

import argparse
import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

mp_client = boto3.client("marketplace-agreement")

def get_agreement_terms(agreement_id):
    try:
        agreement = mp_client.get_agreement_terms(agreementId=agreement_id)
```

```

        return agreement

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", agreement_id)

        else:
            logger.error("Unexpected error: %s", e)

    return None

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--agreement-id",
        "-aid",
        help="Provide agreement ID to describe agreement status",
        required=True,
    )
    args = parser.parse_args()

    agreement = get_agreement_terms(agreement_id=args.agreement_id)

    if agreement is not None:
        freetrial_found = False

        for term in agreement["acceptedTerms"]:
            if "freeTrialPricingTerm" in term.keys():
                helper.pretty_print_datetime(term)
                freetrial_found = True

        if not freetrial_found:
            print(f"No free trial term found for agreement: {args.agreement_id}")

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에 대한 정보 가져오기

다음 코드 예제에서는 계약에 대한 정보를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementResponse;

public class DescribeAgreement {

    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        DescribeAgreementResponse describeAgreementResponse = getResponse(agreementId);

        ReferenceCodesUtils.formatOutput(describeAgreementResponse);

    }

    public static DescribeAgreementResponse getResponse(String agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
```

```

MarketplaceAgreementClient.builder()
    .httpClient(ApacheHttpClient.builder().build())
    .credentialsProvider(ProfileCredentialsProvider.create())
    .build();

DescribeAgreementRequest describeAgreementRequest =
    DescribeAgreementRequest.builder()
        .agreementId(agreementId)
        .build();

DescribeAgreementResponse describeAgreementResponse =
    marketplaceAgreementClient.describeAgreement(describeAgreementRequest);
return describeAgreementResponse;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to get agreement information
AG-07
"""

import argparse
import logging

import boto3

```

```
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-agreement")

logger = logging.getLogger(__name__)

def get_agreement_information(agreement_id):
    try:
        response = mp_client.describe_agreement(agreementId=agreement_id)
    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", agreement_id)
            raise e
        else:
            logger.error("Unexpected error: %s", e)
            raise e

    return response

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--agreement-id",
        "-aid",
        help="Provide agreement ID to describe agreement status",
        required=True,
    )
    args = parser.parse_args()

    response = get_agreement_information(agreement_id=args.agreement_id)

    helper.pretty_print_datetime(response)
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에서 제품 및 제안 세부 정보 가져오기

다음 코드 예제에서는 계약에서 제품 및 제안 세부 정보를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Resource;

import java.util.ArrayList;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
```

```
import
software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;

public class GetProductAndOfferDetailFromAgreement {

    public static void main(String[] args) {

        // call Agreement API to get offer and product information for the agreement

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        List<DescribeEntityResponse> entityResponseList = getEntities(agreementId);

        for (DescribeEntityResponse response : entityResponseList) {
            ReferenceCodesUtils.formatOutput(response);
        }
    }

    public static List<DescribeEntityResponse> getEntities(String agreementId) {
        List<DescribeEntityResponse> entityResponseList = new
        ArrayList<DescribeEntityResponse> ();

        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        DescribeAgreementRequest describeAgreementRequest =
            DescribeAgreementRequest.builder()
                .agreementId(agreementId)
                .build();

        DescribeAgreementResponse describeAgreementResponse =
            marketplaceAgreementClient.describeAgreement(describeAgreementRequest);

        // get offer id for the given agreement

        String offerId = describeAgreementResponse.proposalSummary().offerId();

        // get all the product ids for this agreement
```

```
List<String> productIds = new ArrayList<String>();
for (Resource resource :
describeAgreementResponse.proposalSummary().resources()) {
    productIds.add(resource.id());
}

// call Catalog API to get the details of the offer and products

MarketplaceCatalogClient marketplaceCatalogClient =
    MarketplaceCatalogClient.builder()
        .httpClient(ApacheHttpClient.builder().build())
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();

DescribeEntityRequest describeEntityRequest =
    DescribeEntityRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .entityId(offerId).build();

DescribeEntityResponse describeEntityResponse =
marketplaceCatalogClient.describeEntity(describeEntityRequest);

entityResponseList.add(describeEntityResponse);

for (String productId : productIds) {
    describeEntityRequest =
        DescribeEntityRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .entityId(productId).build();
    describeEntityResponse =
marketplaceCatalogClient.describeEntity(describeEntityRequest);
    System.out.println("Print details for product " + productId);
    entityResponseList.add(describeEntityResponse);
}
return entityResponseList;
}
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to get product and offer details
in a given agreement
AG-10
"""

import argparse
import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

mpa_client = boto3.client("marketplace-agreement")
mpc_client = boto3.client("marketplace-catalog")

logger = logging.getLogger(__name__)

def get_agreement_information(agreement_id):
    """
    Returns information about a given agreement
    Args: agreement_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """
```

```

"""

try:
    agreement = mpa_client.describe_agreement(agreementId=agreement_id)

    return agreement

except ClientError as e:
    if e.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Agreement with ID %s not found.", agreement_id)
    else:
        logger.error("Unexpected error: %s", e)

def get_entity_information(entity_id):
    """
    Returns information about a given entity
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of entity information
    """

    try:
        response = mpc_client.describe_entity(
            Catalog="AWSMarketplace",
            EntityId=entity_id,
        )

        return response

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Entity with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def get_agreement_components(agreement_id):
    agreement_component_list = []

    agreement = get_agreement_information(agreement_id)

    if agreement is not None:
        productIds = []
        for resource in agreement["proposalSummary"]["resources"]:

```

```
        productIds.append(resource["id"])

    for product_id in productIds:
        product_document = get_entity_information(product_id)

        product_document_dict = {}
        product_document_dict["product_id"] = product_id
        product_document_dict["document"] = product_document
        agreement_component_list.append(product_document_dict)

    offerId = agreement["proposalSummary"]["offerId"]

    offer_document = get_entity_information(offerId)

    offer_document_dict = {}
    offer_document_dict["offer_id"] = offerId
    offer_document_dict["document"] = offer_document
    agreement_component_list.append(offer_document_dict)

    return agreement_component_list

else:
    print("Agreement with ID " + args.agreement_id + " is not found")

if __name__ == "__main__":
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--agreement_id",
        "-aid",
        help="Provide agreement ID to search for product and offer detail",
        required=True,
    )
    args = parser.parse_args()

    product_offer_detail =
    get_agreement_components(agreement_id=args.agreement_id)

    helper.pretty_print_datetime(product_offer_detail)
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약의 EULA 가져오기

다음 코드 예제에서는 계약의 EULA를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.DocumentItem;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;

import java.util.ArrayList;
import java.util.List;

import static
    com.example.awsmarketplace.utils.ReferenceCodesConstants.AGREEMENT_ID;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementTermsEula {
```

```

/*
 * Obtain the EULA I have entered into with my customer via the agreement
 */
public static void main(String[] args) {

    String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

    List<DocumentItem> legalEulaArray = getLegalEula(agreementId);

    ReferenceCodesUtils.formatOutput(legalEulaArray);
}

public static List<DocumentItem> getLegalEula(String agreementId) {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    GetAgreementTermsRequest getAgreementTermsRequest =
        GetAgreementTermsRequest.builder().agreementId(agreementId)
            .build();

    GetAgreementTermsResponse getAgreementTermsResponse =
        marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);

    List<DocumentItem> legalEulaArray = new ArrayList<>();

    getAgreementTermsResponse.acceptedTerms().stream()
        .filter(acceptedTerm -> acceptedTerm.legalTerm() != null &&
            acceptedTerm.legalTerm().hasDocuments())
        .flatMap(acceptedTerm -> acceptedTerm.legalTerm().documents().stream())
        .filter(docItem -> docItem.type() != null)
        .forEach(legalEulaArray::add);
    return legalEulaArray;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain the EULA I have entered into with my customer via the agreement
AG-18
"""

import json
import logging
import os

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# agreement id
AGREEMENT_ID = "agmt-11111111111111111111111111111111"

# to use sample file or not
USE_SAMPLE_FILE = False
SAMPLE_FILE_NAME = "mockup_agreement_terms.json"

# attribute name
ROOT_ELEM = "acceptedTerms"
```

```

TERM_NAME = "legalTerm"
CONFIG_ELEM = "configuration"
ATTRIBUTE_NAME = "documents"

def get_agreement_information(mp_client, entity_id):
    """
    Returns customer AWS Account id about a given agreement
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """

    try:
        if USE_SAMPLE_FILE:
            sample_file = os.path.join(os.path.dirname(__file__),
SAMPLE_FILE_NAME)
            terms = open_json_file(sample_file)
        else:
            terms = mp_client.get_agreement_terms(agreementId=entity_id)

        legalEulaArray = []
        for term in terms[ROOT_ELEM]:
            if TERM_NAME in term and ATTRIBUTE_NAME in term[TERM_NAME]:
                docs = term[TERM_NAME][ATTRIBUTE_NAME]
                for doc in docs:
                    if "type" in doc:
                        legalEulaArray.append(doc)
        return legalEulaArray

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for an agreement in the AWS Marketplace.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-agreement")

```

```

        helper.pretty_print_datetime(get_agreement_information(mp_client,
            AGREEMENT_ID))

    # open json file from path

def open_json_file(filename):
    with open(filename, "r") as f:
        return json.load(f)

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAgreementTerms](#)을 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약의 자동 갱신 조건 가져오기

다음 코드 예제에서는 계약의 자동 갱신 조건을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.seller;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;

```



```
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;

public class GetAgreementAutoRenewal {

    /**
     * Obtain the auto-renewal status of the agreement
     */

    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        String autoRenewal = getAutoRenewal(agreementId);

        System.out.println("Auto-Renewal status is " + autoRenewal);
    }

    public static String getAutoRenewal(String agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementTermsRequest getAgreementTermsRequest =
            GetAgreementTermsRequest.builder()
                .agreementId(agreementId)
                .build();

        GetAgreementTermsResponse getAgreementTermsResponse =
            marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);

        String autoRenewal = "No Auto Renewal";

        for (AcceptedTerm acceptedTerm : getAgreementTermsResponse.acceptedTerms()) {
            if (acceptedTerm.renewalTerm() != null &&
                acceptedTerm.renewalTerm().configuration() != null
```

```

        && acceptedTerm.renewalTerm().configuration().enableAutoRenew() != null) {
            autoRenewal =
String.valueOf(acceptedTerm.renewalTerm().configuration().enableAutoRenew().booleanValue
            break;
        }
    }
    return autoRenewal;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain the auto-renewal status of the agreement
AG-15
"""

import json
import logging
import os
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper

```

```
import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# agreement id
AGREEMENT_ID = "agmt-11111111111111111111"

# to use sample file or not
USE_SAMPLE_FILE = False
SAMPLE_FILE_NAME = "mockup_agreement_terms.json"

# attribute name
ROOT_ELEM = "acceptedTerms"
TERM_NAME = "renewalTerm"
CONFIG_ELEM = "configuration"
ATTRIBUTE_NAME = "enableAutoRenew"

def get_agreement_information(mp_client, entity_id):
    """
    Returns customer AWS Account id about a given agreement
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """

    try:
        if USE_SAMPLE_FILE:
            sample_file = os.path.join(os.path.dirname(__file__),
SAMPLE_FILE_NAME)
            terms = open_json_file(sample_file)
        else:
            terms = mp_client.get_agreement_terms(agreementId=entity_id)

        auto_renewal = "No Auto Renewal"
        for term in terms[ROOT_ELEM]:
            if TERM_NAME in term:
                if CONFIG_ELEM in term[TERM_NAME]:
                    auto_renewal = term[TERM_NAME][CONFIG_ELEM][ATTRIBUTE_NAME]
                    break
        return auto_renewal

    except ClientError as e:
```

```

        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for an agreement in the AWS Marketplace.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-agreement")

    agreement = get_agreement_information(mp_client, AGREEMENT_ID)

    if agreement is not None:
        print(f"Auto Renewal is {agreement}")
    else:
        print("Agreement with ID " + AGREEMENT_ID + " is not found")

# open json file from path
def open_json_file(filename):
    with open(filename, "r") as f:
        return json.load(f)

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAgreementTerms](#)을 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에서 구매한 차원 가져오기

다음 코드 예제에서는 계약에서 구매한 차원을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;
import software.amazon.awssdk.services.marketplaceagreement.model.Dimension;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;

import java.util.ArrayList;
import java.util.List;

import static
    com.example.awsmarketplace.utils.ReferenceCodesConstants.AGREEMENT_ID;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementTermsDimensionPurchased {

    /*
     * Obtain the dimensions the buyer has purchased from me via the agreement
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        List<String> dimensionKeys = getDimensionKeys(agreementId);

        ReferenceCodesUtils.formatOutput(dimensionKeys);
    }
}
```

```

}

public static List<String> getDimensionKeys(String agreementId) {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    GetAgreementTermsRequest getAgreementTermsRequest =
        GetAgreementTermsRequest.builder().agreementId(agreementId)
            .build();

    GetAgreementTermsResponse getAgreementTermsResponse =
        marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);

    List<String> dimensionKeys = new ArrayList<String>();
    for (AcceptedTerm acceptedTerm : getAgreementTermsResponse.acceptedTerms()) {
        if (acceptedTerm.configurableUpfrontPricingTerm() != null) {
            if
            (acceptedTerm.configurableUpfrontPricingTerm().configuration().selectorValue() !=
            null) {
                List<Dimension> dimensions =
                    acceptedTerm.configurableUpfrontPricingTerm().configuration().dimensions();
                for (Dimension dimension : dimensions) {
                    dimensionKeys.add(dimension.dimensionKey());
                }
            }
        }
    }
    return dimensionKeys;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain the dimensions the buyer has purchased from me via the agreement
AG-28
"""

import json
import logging
import os

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# agreement id
AGREEMENT_ID = "agmt-11111111111111111111111111111111"

# to use sample file or not
USE_SAMPLE_FILE = False
SAMPLE_FILE_NAME = "mockup_agreement_terms.json"

# attribute name
ROOT_ELEM = "acceptedTerms"
```

```

TERM_NAME = "configurableUpfrontPricingTerm"
CONFIG_ELEM = "configuration"
ATTRIBUTE_NAME = "selectorValue"

def get_agreement_information(mp_client, entity_id):
    """
    Returns customer AWS Account id about a given agreement
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """

    try:
        if USE_SAMPLE_FILE:
            sample_file = os.path.join(os.path.dirname(__file__),
SAMPLE_FILE_NAME)
            terms = open_json_file(sample_file)
        else:
            terms = mp_client.get_agreement_terms(agreementId=entity_id)

        dimensionKeys = []

        for term in terms[ROOT_ELEM]:
            if TERM_NAME in term:
                if CONFIG_ELEM in term[TERM_NAME]:
                    confParam = term[TERM_NAME][CONFIG_ELEM]
                    if ATTRIBUTE_NAME in confParam:
                        if "dimensions" in confParam:
                            for dimension in confParam["dimensions"]:
                                if "dimensionKey" in dimension:
                                    dimensionKey = dimension["dimensionKey"]
                                    print(f"Dimension Key: {dimensionKey}")
                                    dimensionKeys.append(dimensionKey)

        return dimensionKeys

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

```



```

print("-" * 88)
print("Looking for an agreement in the AWS Marketplace.")
print("-" * 88)

mp_client = boto3.client("marketplace-agreement")

helper.pretty_print_datetime(get_agreement_information(mp_client,
AGREEMENT_ID))

# open json file from path

def open_json_file(filename):
    with open(filename, "r") as f:
        return json.load(f)

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAgreementTerms](#)을 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에서 구매한 각 차원의 인스턴스 가져오기

다음 코드 예제에서는 계약에서 구매한 각 차원의 인스턴스를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;
import software.amazon.awssdk.services.marketplaceagreement.model.Dimension;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import static
    com.example.awsmarketplace.utils.ReferenceCodesConstants.AGREEMENT_ID;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementTermsDimensionInstances {

    /*
     * get instances of each dimension that buyer has purchased in the agreement
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        Map<String, List<Dimension>> dimensionMap = getDimensions(agreementId);

        ReferenceCodesUtils.formatOutput(dimensionMap);
    }

    public static Map<String, List<Dimension>> getDimensions(String agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
    }

```

```

GetAgreementTermsRequest getAgreementTermsRequest =
    GetAgreementTermsRequest.builder().agreementId(agreementId)
        .build();

GetAgreementTermsResponse getAgreementTermsResponse =
    marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);

Map<String, List<Dimension>> dimensionMap = new HashMap<String,
    List<Dimension>>();

for (AcceptedTerm acceptedTerm : getAgreementTermsResponse.acceptedTerms()) {
    List<Dimension> dimensionsList = new ArrayList<Dimension>();
    if (acceptedTerm.configurableUpfrontPricingTerm() != null) {
        String selectorValue = "";
        if (acceptedTerm.configurableUpfrontPricingTerm().configuration() != null) {
            if
                (acceptedTerm.configurableUpfrontPricingTerm().configuration().selectorValue() !=
                    null) {
                selectorValue =
                    acceptedTerm.configurableUpfrontPricingTerm().configuration().selectorValue();
            }
            if
                (acceptedTerm.configurableUpfrontPricingTerm().configuration().hasDimensions())
            {
                dimensionsList =
                    acceptedTerm.configurableUpfrontPricingTerm().configuration().dimensions();
            }
        }
        if (selectorValue.length() > 0) {
            dimensionMap.put(selectorValue, dimensionsList);
        }
    }
}
return dimensionMap;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain instances of each dimension that buyer has purchased in the agreement
AG-30
"""

import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# agreement id
AGREEMENT_ID = "agmt-11111111111111111111111111111111"

# attribute name
ROOT_ELEM = "acceptedTerms"
TERM_NAME = "configurableUpfrontPricingTerm"
CONFIG_ELEM = "configuration"
ATTRIBUTE_NAME = "selectorValue"

logger = logging.getLogger(__name__)
```

```

def get_agreement_information(mp_client, entity_id):
    """
    Returns customer AWS Account id about a given agreement
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """

    try:
        terms = mp_client.get_agreement_terms(agreementId=entity_id)
        dimensionKeyValueTypeMap = {}
        for term in terms[ROOT_ELEM]:
            if TERM_NAME in term:
                if CONFIG_ELEM in term[TERM_NAME]:
                    confParam = term[TERM_NAME][CONFIG_ELEM]
                    if ATTRIBUTE_NAME in confParam:
                        selectValue = confParam["selectorValue"]
                        dimensionKeyValueTypeMap["selectorValue"] = selectValue
                        if "dimensions" in confParam:
                            dimensionKeyValueTypeMap["dimensions"] =
confParam["dimensions"]
                            """
                            for dimension in confParam['dimensions']:
                                if 'dimensionKey' in dimension:

                                    dimensionValue = dimension['dimensionValue']
                                    dimensionKey = dimension['dimensionKey']
                                    print(f"Selector: {selectValue}, Dimension
Key: {dimensionKey}, Dimension Value: {dimensionValue}")
                                    dimensionKeyValueTypeMap[dimensionKey] =
dimensionValue
                            """
                            return dimensionKeyValueTypeMap

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

```

```

print("-" * 88)
print("Looking for an agreement in the AWS Marketplace.")
print("-" * 88)

mp_client = boto3.client("marketplace-agreement")

helper.pretty_print_datetime(get_agreement_information(mp_client,
AGREEMENT_ID))

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAgreementTerms](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약의 결제 일정 가져오기

다음 코드 예제에서는 계약의 결제 일정을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;

```

```
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;
import
    software.amazon.awssdk.services.marketplaceagreement.model.PaymentScheduleTerm;
import software.amazon.awssdk.services.marketplaceagreement.model.ScheduleItem;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementTermsPaymentSchedule {

    /**
     * Obtain the payment schedule I have agreed to with the agreement, including
     * the invoice date and invoice amount
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        List<Map<String, Object>> paymentScheduleArray =
            getPaymentSchedules(agreementId);

        ReferenceCodesUtils.formatOutput(paymentScheduleArray);
    }

    public static List<Map<String, Object>> getPaymentSchedules(String agreementId)
    {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementTermsRequest getAgreementTermsRequest =
            GetAgreementTermsRequest.builder().agreementId(agreementId)
                .build();
```

```

GetAgreementTermsResponse getAgreementTermsResponse =
marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);
List<Map<String, Object>> paymentScheduleArray = new ArrayList<>();

String currencyCode = "";

for (AcceptedTerm acceptedTerm : getAgreementTermsResponse.acceptedTerms()) {
    if (acceptedTerm.paymentScheduleTerm() != null) {
        PaymentScheduleTerm paymentScheduleTerm = acceptedTerm.paymentScheduleTerm();
        if (paymentScheduleTerm.currencyCode() != null) {
            currencyCode = paymentScheduleTerm.currencyCode();
        }
        if (paymentScheduleTerm.hasSchedule()) {
            for (ScheduleItem schedule : paymentScheduleTerm.schedule()) {
                if (schedule.chargeDate() != null) {
                    String chargeDate = schedule.chargeDate().toString();
                    String chargeAmount = schedule.chargeAmount();
                    Map<String, Object> scheduleMap = new HashMap<>();
                    scheduleMap.put(ATTRIBUTE_CURRENCY_CODE, currencyCode);
                    scheduleMap.put(ATTRIBUTE_CHARGE_DATE, chargeDate);
                    scheduleMap.put(ATTRIBUTE_CHARGE_AMOUNT, chargeAmount);
                    paymentScheduleArray.add(scheduleMap);
                }
            }
        }
    }
}
return paymentScheduleArray;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.


```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain the payment schedule I have agreed to with the agreement, including the
invoice date and invoice amount
AG-17
"""

import json
import logging
import os

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# agreement id
AGREEMENT_ID = "agmt-11111111111111111111111111111111"

# to use sample file or not
USE_SAMPLE_FILE = False
SAMPLE_FILE_NAME = "mockup_agreement_terms.json"

# attribute name
ROOT_ELEM = "acceptedTerms"
TERM_NAME = "paymentScheduleTerm"
CONFIG_ELEM = "configuration"
ATTRIBUTE_NAME = "selectorValue"

def get_agreement_information(mp_client, entity_id):
    """
    Returns customer AWS Account id about a given agreement
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """

```

```

"""

try:
    if USE_SAMPLE_FILE:
        sample_file = os.path.join(os.path.dirname(__file__),
SAMPLE_FILE_NAME)
        terms = open_json_file(sample_file)
    else:
        terms = mp_client.get_agreement_terms(agreementId=entity_id)

    paymentScheduleArray = []
    currencyCode = ""
    for term in terms[ROOT_ELEM]:
        if TERM_NAME in term:
            paymentSchedule = term[TERM_NAME]
            if "currencyCode" in paymentSchedule:
                currencyCode = paymentSchedule["currencyCode"]
            if "schedule" in paymentSchedule:
                for sch in paymentSchedule["schedule"]:
                    if "chargeDate" in sch:
                        chargeDate = sch["chargeDate"]
                        chargeAmount = sch["chargeAmount"]
                        # print(f"chargeDate: {chargeDate}, chargeAmount:
{chargeAmount}")

                        schedule = {
                            "currencyCode": currencyCode,
                            "chargeDate": chargeDate,
                            "chargeAmount": chargeAmount,
                        }
                        paymentScheduleArray.append(schedule)

    return paymentScheduleArray

except ClientError as e:
    if e.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Agreement with ID %s not found.", entity_id)
    else:
        logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)

```

```

print("Looking for an agreement in the AWS Marketplace.")
print("-" * 88)

mp_client = boto3.client("marketplace-agreement")

helper.pretty_print_datetime(get_agreement_information(mp_client,
AGREEMENT_ID))

# open json file from path

def open_json_file(filename):
    with open(filename, "r") as f:
        return json.load(f)

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAgreementTerms](#)을 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약에서 차원당 요금 가져오기

다음 코드 예제에서는 계약에서 차원당 요금을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;
```

```
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;

import java.util.ArrayList;
import java.util.List;

import static
    com.example.awsmarketplace.utils.ReferenceCodesConstants.AGREEMENT_ID;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementTermsPricingEachDimension {

    /*
     * Obtain pricing per each dimension in the agreement
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        List<Object> dimensions = getDimensions(agreementId);

        ReferenceCodesUtils.formatOutput(dimensions);
    }

    public static List<Object> getDimensions(String agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementTermsRequest getAgreementTermsRequest =
            GetAgreementTermsRequest.builder().agreementId(agreementId)
                .build();
```

```

GetAgreementTermsResponse getAgreementTermsResponse =
marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);

List<Object> dimensions = new ArrayList<Object>();

for (AcceptedTerm acceptedTerm : getAgreementTermsResponse.acceptedTerms()) {
    List<Object> rateInfo = new ArrayList<Object>();
    if (acceptedTerm.configurableUpfrontPricingTerm() != null) {
        if (acceptedTerm.configurableUpfrontPricingTerm().type() != null) {
            rateInfo.add(acceptedTerm.configurableUpfrontPricingTerm().type());
        }
        if (acceptedTerm.configurableUpfrontPricingTerm().currencyCode() != null) {
            rateInfo.add(acceptedTerm.configurableUpfrontPricingTerm().currencyCode());
        }
        if (acceptedTerm.configurableUpfrontPricingTerm().hasRateCards()) {
            rateInfo.add(acceptedTerm.configurableUpfrontPricingTerm().rateCards());
        }
        dimensions.add(rateInfo);
    }
}
return dimensions;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain pricing per each dimension in the agreement

```

```

AG-29
"""

import json
import logging
import os

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# agreement id
AGREEMENT_ID = "agmt-11111111111111111111111111111111"

# to use sample file or not
USE_SAMPLE_FILE = False
SAMPLE_FILE_NAME = "mockup_agreement_terms.json"

# attribute name
ROOT_ELEM = "acceptedTerms"
TERM_NAME = "configurableUpfrontPricingTerm"
CONFIG_ELEM = "configuration"
ATTRIBUTE_NAME = "selectorValue"

TERMS_TO_SEARCH = [
    "configurableUpfrontPricingTerm",
    "usageBasedPricingTerm",
    "fixedUpfrontPricingTerm",
]

def get_agreement_information(mp_client, entity_id):
    """
    Returns customer AWS Account id about a given agreement
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """

```

```

try:
    if USE_SAMPLE_FILE:
        sample_file = os.path.join(os.path.dirname(__file__),
SAMPLE_FILE_NAME)
        terms = open_json_file(sample_file)
    else:
        terms = mp_client.get_agreement_terms(agreementId=entity_id)

    dimentions = []
    for term in terms[ROOT_ELEM]:
        for t in TERMS_TO_SEARCH:
            rateInfo = []
            if t in term:
                if "type" in term[t]:
                    rateInfo.append(term[t]["type"])
                if "currencyCode" in term[t]:
                    rateInfo.append(term[t]["currencyCode"])
                if "rateCards" in term[t]:
                    rateInfo.append(term[t]["rateCards"])
                dimentions.append(rateInfo)
    return dimentions

except ClientError as e:
    if e.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Agreement with ID %s not found.", entity_id)
    else:
        logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for an agreement in the AWS Marketplace.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-agreement")

    helper.pretty_print_datetime(get_agreement_information(mp_client,
AGREEMENT_ID))

    # open json file from path

```

```
def open_json_file(filename):
    with open(filename, "r") as f:
        return json.load(f)

if __name__ == "__main__":
    usage_demo()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAgreementTerms](#)을 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약의 요금 유형 가져오기

다음 코드 예제에서는 계약의 요금 유형을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AgreementViewSummary;
import software.amazon.awssdk.services.marketplaceagreement.model.Filter;
```



```
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsResponse;

import com.fasterxml.jackson.annotation.JsonAutoDetect.Visibility;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import java.util.Objects;
import java.util.Set;

import org.apache.commons.lang3.tuple.Tuple;

import
    software.amazon.awssdk.services.marketplacecatalog.MarketplaceCatalogClient;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityRequest;
import
    software.amazon.awssdk.services.marketplacecatalog.model.DescribeEntityResponse;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.fasterxml.jackson.annotation.PropertyAccessor;
import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.databind.ObjectWriter;
import com.fasterxml.jackson.datatype.jsr310.JavaTimeModule;

/*
 * Obtain the pricing type of the agreement (contract, FPS, metered, free etc.)
 */
public class GetAgreementPricingType {

    private static final String FILTER_NAME = "OfferId";
```

```

private static final String FILTER_VALUE = OFFER_ID;

// Product types
private static final String SAAS_PRODUCT = "SaaSProduct";
private static final String AMI_PRODUCT = "AmiProduct";
private static final String ML_PRODUCT = "MachineLearningProduct";
private static final String CONTAINER_PRODUCT = "ContainerProduct";
private static final String DATA_PRODUCT = "DataProduct";
private static final String PROSERVICE_PRODUCT = "ProfessionalServicesProduct";
private static final String AIQ_PRODUCT = "AiqProduct";

// Pricing types
private static final String CCP = "CCP";
private static final String ANNUAL = "Annual";
private static final String CONTRACT = "Contract";
private static final String SFT = "SaaS Free Trial";
private static final String HMA = "Hourly and Monthly Agreements";
private static final String HOURLY = "Hourly";
private static final String MONTHLY = "Monthly";
private static final String AFPS = "Annual FPS";
private static final String CFPS = "Contract FPS";
private static final String CCPFPS = "CCP with FPS";
private static final String BYOL = "BYOL";
private static final String FREE = "Free";
private static final String FTH = "Free Trials and Hourly";

// Agreement term pricing types
private static final Set<String> LEGAL = Set.of("LegalTerm");
private static final Set<String> CONFIGURABLE_UPFRONT =
Set.of("ConfigurableUpfrontPricingTerm");
private static final Set<String> USAGE_BASED = Set.of("UsageBasedPricingTerm");
private static final Set<String> CONFIGURABLE_UPFRONT_AND_USAGE_BASED =
Set.of("ConfigurableUpfrontPricingTerm", "UsageBasedPricingTerm");
private static final Set<String> FREE_TRIAL = Set.of("FreeTrialPricingTerm");
private static final Set<String> RECURRING_PAYMENT =
Set.of("RecurringPaymentTerm");
private static final Set<String> USAGE_BASED_AND_RECURRING_PAYMENT =
Set.of("UsageBasedPricingTerm", "RecurringPaymentTerm");
private static final Set<String> FIXED_UPFRONT_AND_PAYMENT_SCHEDULE =
Set.of("FixedUpfrontPricingTerm", "PaymentScheduleTerm");
private static final Set<String>
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED =

```

```

Set.of("FixedUpfrontPricingTerm", "PaymentScheduleTerm",
"UsageBasedPricingTerm");
private static final Set<String> BYOL_PRICING = Set.of("ByolPricingTerm");
private static final Set<String> FREE_TRIAL_AND_USAGE_BASED =
Set.of("FreeTrialPricingTerm", "UsageBasedPricingTerm");

private static final List<Set<String>> ALL_AGREEMENT_TERM_TYPES_COMBINATION
= Arrays.asList(LEGAL, CONFIGURABLE_UPFRONT, USAGE_BASED,
CONFIGURABLE_UPFRONT_AND_USAGE_BASED,
    FREE_TRIAL, RECURRING_PAYMENT, USAGE_BASED_AND_RECURRING_PAYMENT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED, BYOL_PRICING,
FREE_TRIAL_AND_USAGE_BASED);

private static MarketplaceAgreementClient marketplaceAgreementClient =
    MarketplaceAgreementClient.builder()
        .httpClient(ApacheHttpClient.builder().build())
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();

private static MarketplaceCatalogClient marketplaceCatalogClient =
    MarketplaceCatalogClient.builder()
        .httpClient(ApacheHttpClient.builder().build())
        .credentialsProvider(ProfileCredentialsProvider.create())
        .build();

/*
 * Get agreement Pricing Type given product type, agreement term types and
offer types if needed
 */
public static String getPricingType(String productType, Set<String>
agreementTermType, Set<String> offerType) {
    Map<Triple<String, Set<String>, Set<String>>, String> pricingTypes = new
HashMap<>();

    pricingTypes.put(Triple.of(SAAS_PRODUCT, CONFIGURABLE_UPFRONT_AND_USAGE_BASED,
new HashSet<>()), CCP);
    pricingTypes.put(Triple.of(DATA_PRODUCT, CONFIGURABLE_UPFRONT_AND_USAGE_BASED,
new HashSet<>()), CCP);
    pricingTypes.put(Triple.of(CONTAINER_PRODUCT, CONFIGURABLE_UPFRONT,
CONFIGURABLE_UPFRONT_AND_USAGE_BASED), ANNUAL);
    pricingTypes.put(Triple.of(AMI_PRODUCT, CONFIGURABLE_UPFRONT,
CONFIGURABLE_UPFRONT_AND_USAGE_BASED), ANNUAL);

```

```

pricingTypes.put(Triple.of(ML_PRODUCT, CONFIGURABLE_UPFRONT,
CONFIGURABLE_UPFRONT_AND_USAGE_BASED), ANNUAL);
pricingTypes.put(Triple.of(CONTAINER_PRODUCT, CONFIGURABLE_UPFRONT,
CONFIGURABLE_UPFRONT), CONTRACT);
pricingTypes.put(Triple.of(AMI_PRODUCT, CONFIGURABLE_UPFRONT,
CONFIGURABLE_UPFRONT), CONTRACT);
pricingTypes.put(Triple.of(SAAS_PRODUCT, CONFIGURABLE_UPFRONT, new
HashSet<>()), CONTRACT);
pricingTypes.put(Triple.of(DATA_PRODUCT, CONFIGURABLE_UPFRONT, new
HashSet<>()), CONTRACT);
pricingTypes.put(Triple.of(AIQ_PRODUCT, CONFIGURABLE_UPFRONT, new HashSet<>()),
CONTRACT);
pricingTypes.put(Triple.of(PROSERVICE_PRODUCT, CONFIGURABLE_UPFRONT, new
HashSet<>()), CONTRACT);
pricingTypes.put(Triple.of(SAAS_PRODUCT, FREE_TRIAL, new HashSet<>()), SFT);
pricingTypes.put(Triple.of(AMI_PRODUCT, USAGE_BASED_AND_RECURRING_PAYMENT, new
HashSet<>()), HMA);
pricingTypes.put(Triple.of(SAAS_PRODUCT, USAGE_BASED, new HashSet<>()),
HOURLY);
pricingTypes.put(Triple.of(AMI_PRODUCT, USAGE_BASED, new HashSet<>()), HOURLY);
pricingTypes.put(Triple.of(ML_PRODUCT, USAGE_BASED, new HashSet<>()), HOURLY);
pricingTypes.put(Triple.of(CONTAINER_PRODUCT, RECURRING_PAYMENT, new
HashSet<>()), MONTHLY);
pricingTypes.put(Triple.of(AMI_PRODUCT, RECURRING_PAYMENT, new HashSet<>()),
MONTHLY);
pricingTypes.put(Triple.of(CONTAINER_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED), AFPS);
pricingTypes.put(Triple.of(AMI_PRODUCT, FIXED_UPFRONT_AND_PAYMENT_SCHEDULE,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED), AFPS);
pricingTypes.put(Triple.of(ML_PRODUCT, FIXED_UPFRONT_AND_PAYMENT_SCHEDULE, new
HashSet<>()), AFPS);
pricingTypes.put(Triple.of(CONTAINER_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE, new HashSet<>()), CFPS);
pricingTypes.put(Triple.of(AMI_PRODUCT, FIXED_UPFRONT_AND_PAYMENT_SCHEDULE,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE), CFPS);
pricingTypes.put(Triple.of(SAAS_PRODUCT, FIXED_UPFRONT_AND_PAYMENT_SCHEDULE,
new HashSet<>()), CFPS);
pricingTypes.put(Triple.of(DATA_PRODUCT, FIXED_UPFRONT_AND_PAYMENT_SCHEDULE,
new HashSet<>()), CFPS);
pricingTypes.put(Triple.of(AIQ_PRODUCT, FIXED_UPFRONT_AND_PAYMENT_SCHEDULE, new
HashSet<>()), CFPS);
pricingTypes.put(Triple.of(PROSERVICE_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE, new HashSet<>()), CFPS);

```

```

pricingTypes.put(Triple.of(SAAS_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED, new HashSet<>()), CCPFPS);
pricingTypes.put(Triple.of(DATA_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED, new HashSet<>()), CCPFPS);
pricingTypes.put(Triple.of(AIQ_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED, new HashSet<>()), CCPFPS);
pricingTypes.put(Triple.of(PROSERVICE_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE_AND_USAGE_BASED, new HashSet<>()), CCPFPS);
pricingTypes.put(Triple.of(AMI_PRODUCT, BYOL_PRICING, new HashSet<>()), BYOL);
pricingTypes.put(Triple.of(SAAS_PRODUCT, BYOL_PRICING, new HashSet<>()), BYOL);
pricingTypes.put(Triple.of(PROSERVICE_PRODUCT, BYOL_PRICING, new HashSet<>()),
BYOL);
pricingTypes.put(Triple.of(AIQ_PRODUCT, BYOL_PRICING, new HashSet<>()), BYOL);
pricingTypes.put(Triple.of(ML_PRODUCT, BYOL_PRICING, new HashSet<>()), BYOL);
pricingTypes.put(Triple.of(CONTAINER_PRODUCT, BYOL_PRICING, new HashSet<>()),
BYOL);
pricingTypes.put(Triple.of(DATA_PRODUCT, BYOL_PRICING, new HashSet<>()), BYOL);
pricingTypes.put(Triple.of(CONTAINER_PRODUCT, LEGAL, new HashSet<>()), FREE);
pricingTypes.put(Triple.of(AMI_PRODUCT, FREE_TRIAL_AND_USAGE_BASED, new
HashSet<>()), FTH);
pricingTypes.put(Triple.of(CONTAINER_PRODUCT, FREE_TRIAL_AND_USAGE_BASED, new
HashSet<>()), FTH);
pricingTypes.put(Triple.of(ML_PRODUCT, FREE_TRIAL_AND_USAGE_BASED, new
HashSet<>()), FTH);

Triple<String, Set<String>, Set<String>> key = Triple.of(productType,
agreementTermType, offerType);

if (pricingTypes.containsKey(key)) {
    return pricingTypes.get(key);
} else {
    return "Unknown";
}
}

/*
 * Given product type and agreement term types, some combinations need to check
offer term types as well.
 */
public static String needToCheckOfferTermsType(String productType, Set<String>
agreementTermTypes) {
    Map<KeyPair, String> offerTermTypes = new HashMap<>();
    offerTermTypes.put(new KeyPair(CONTAINER_PRODUCT, CONFIGURABLE_UPFRONT), "Y");
    offerTermTypes.put(new KeyPair(AMI_PRODUCT, CONFIGURABLE_UPFRONT), "Y");

```

```

offerTermTypes.put(new KeyPair(CONTAINER_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE), "Y");
offerTermTypes.put(new KeyPair(AMI_PRODUCT,
FIXED_UPFRONT_AND_PAYMENT_SCHEDULE), "Y");

KeyPair key = new KeyPair(productType, agreementTermTypes);
if (offerTermTypes.containsKey(key)) {
    return offerTermTypes.get(key);
} else {
    return null;
}
}

public static List<AgreementViewSummary> getAgreementsById() {

    List<AgreementViewSummary> agreementSummaryList = new
    ArrayList<AgreementViewSummary>();

    // Set PartyType filter to PARTY_TYPE_FILTER_VALUE_PROPOSER to return
    agreements where you are the proposer.
    // Change to PARTY_TYPE_FILTER_VALUE_ACCEPTOR to return agreements where you
    are the acceptor.
    Filter partyType =
    Filter.builder().name(PARTY_TYPE_FILTER_NAME).values(PARTY_TYPE_FILTER_VALUE_PROPOSER).b

    Filter agreementType =
    Filter.builder().name(AGREEMENT_TYPE_FILTER_NAME).values(AGREEMENT_TYPE_FILTER_VALUE_PUR

    Filter customizeFilter =
    Filter.builder().name(FILTER_NAME).values(FILTER_VALUE).build();

    SearchAgreementsRequest searchAgreementsRequest =
        SearchAgreementsRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .filters(partyType, agreementType, customizeFilter).build();

    SearchAgreementsResponse searchResultResponse =
    marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

    agreementSummaryList.addAll(searchResultResponse.agreementViewSummaries());

    while (searchResultResponse.nextToken() != null &&
    searchResultResponse.nextToken().length() > 0) {

```

```

        searchAgreementsRequest =
SearchAgreementsRequest.builder().catalog(AWS_MP_CATALOG)
        .filters(partyType,
agreementType).nextToken(searchResultResponse.nextToken()).build();
        searchResultResponse =
marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);
        agreementSummaryList.addAll(searchResultResponse.agreementViewSummaries());
    }
    return agreementSummaryList;
}

static class KeyPair {
    private final String first;
    private final Set<String> second;

    public KeyPair(String productType, Set<String> second) {
        this.first = productType;
        this.second = second;
    }

    @Override
    public int hashCode() {
        return Objects.hash(first, second);
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null || getClass() != obj.getClass())
            return false;
        KeyPair other = (KeyPair) obj;
        return Objects.equals(first, other.first) && Objects.equals(second,
other.second);
    }
}

/*
 * Get all the term types for the offer
 */
public static Set<String> getOfferTermTypes(String offerId) {

    Set<String> offerTermTypes = new HashSet<String>();

```

```

DescribeEntityRequest request =
    DescribeEntityRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .entityId(offerId)
        .build();

DescribeEntityResponse result =
marketplaceCatalogClient.describeEntity(request);

String details = result.details();

try {
    ObjectMapper objectMapper = new ObjectMapper();
    JsonNode rootNode = objectMapper.readTree(details);
    JsonNode termsNode = rootNode.get(ATTRIBUTE_TERMS);

    for (JsonNode termNode : termsNode) {
        if (termNode.get(ATTRIBUTE_TYPE_ENTITY) != null ) {
            offerTermTypes.add(termNode.get(ATTRIBUTE_TYPE_ENTITY).asText());
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}

return offerTermTypes;
}

/*
 * Get all the agreement term types
 */
public static Set<String> getAgreementTermTypes(GetAgreementTermsResponse
agreementTerm) {
    Set<String> agreementTermTypes = new HashSet<String>();
    try {
        for (AcceptedTerm term : agreementTerm.acceptedTerms()) {
            ObjectMapper objectMapper = new ObjectMapper();
            JsonNode termNode = objectMapper.readTree(getJson(term));
            Iterator<Map.Entry<String, JsonNode>> fieldsIterator = termNode.fields();
            while (fieldsIterator.hasNext()) {
                Map.Entry<String, JsonNode> entry = fieldsIterator.next();
                JsonNode value = entry.getValue();
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return agreementTermTypes;
}

```



```

        if (value.isObject() && value.has(ATTRIBUTE_TYPE_AGREEMENT)) {
            agreementTermTypes.add(value.get(ATTRIBUTE_TYPE_AGREEMENT).asText());
        }
    }
} catch (Exception e) {
    e.printStackTrace();
}
return agreementTermTypes;
}

/*
 * make sure all elements in array2 exist in array1
 */
public static boolean allElementsExist(Set<String> array1, Set<String> array2) {
    for (String element : array2) {
        boolean found = false;
        for (String str : array1) {
            if (element.equals(str)) {
                found = true;
                break;
            }
        }
        if (!found) {
            return false;
        }
    }
    return true;
}

/*
 * Find the combinations of the agreement term types for the agreement
 */
public static Set<String> getMatchedTermTypesCombination(Set<String>
agreementTermTypes) {
    Set<String> matchedCombination = new HashSet<String>();
    for (Set<String> element : ALL_AGREEMENT_TERM_TYPES_COMBINATION) {
        if (allElementsExist(agreementTermTypes, element)) {
            matchedCombination = element;
        }
    }
    return matchedCombination;
}

```

```
public static void main(String[] args) {

    List<AgreementViewSummary> agreements = getAgreementsById();

    for (AgreementViewSummary summary : agreements) {
        String pricingType = "";
        String agreementId = summary.agreementId();
        System.out.println(agreementId);
        String offerId = summary.proposalSummary().offerId();

        //get all pricing term types for the offer in the agreement
        Set<String> offerTermTypes = getOfferTermTypes(offerId);
        String productType = summary.proposalSummary().resources().get(0).type();

        //get all pricing term types for the agreement
        GetAgreementTermsRequest getAgreementTermsRequest =
            GetAgreementTermsRequest.builder().agreementId(agreementId)
                .build();
        GetAgreementTermsResponse getAgreementTermsResponse =
            marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);
        Set<String> agreementTermTypes =
            getAgreementTermTypes(getAgreementTermsResponse);

        //get matched pricing term type combination set
        Set<String> agreementMatchedTermType =
            getMatchedTermTypesCombination(agreementTermTypes);

        //check to see if this agreement pricing term combination needs additional
        check on offer pricing terms
        String needToCheckOfferType = needToCheckOfferTermsType(productType,
            agreementMatchedTermType);

        // get the pricing type for the agreement based on the product type, agreement
        term types and offer term types if needed
        if (needToCheckOfferType != null) {
            Set<String> offerMatchedTermType =
                getMatchedTermTypesCombination(offerTermTypes);
            pricingType = getPricingType(productType, agreementMatchedTermType,
                offerMatchedTermType);
        } else if (agreementMatchedTermType == LEGAL) {
            pricingType = FREE;
        } else {
```

```

        pricingType = getPricingType(productType, agreementMatchedTermType, new
HashSet());
    }
    System.out.println("Pricing type is " + pricingType);
}
}

private static String getJson(Object result) {
    String json = "";

    try {
        ObjectMapper om = new ObjectMapper();
        om.setVisibility(PropertyAccessor.FIELD, Visibility.ANY);
        om.registerModule(new JavaTimeModule());
        ObjectWriter ow = om.writer().withDefaultPrettyPrinter();

        json = ow.writeValueAsString(result);
    } catch (JsonProcessingException e) {
        e.printStackTrace();
    }
    return json;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose

```

```
Obtain the pricing type of the agreement (contract, FPS, metered, free etc.)
AG-16
"""

import json
import logging

import boto3
from botocore.exceptions import ClientError

# To search by offer id: OfferId; by product id: ResourceIdentifier; by product
  type: ResourceType
idType = "OfferId"

# replace id value as needed
idValue = "offer-1111111111111"

MAX_PAGE_RESULTS = 10

# catalog; switch to AWSMarketplace for release
AWSMPCATALOG = "AWSMarketplace"

# product types

SaaSProduct = "SaaSProduct"
AmiProduct = "AmiProduct"
MLProduct = "MachineLearningProduct"
ContainerProduct = "ContainerProduct"
DataProduct = "DataProduct"
ProServiceProduct = "ProfessionalServicesProduct"
AiqProduct = "AiqProduct"

# Define pricing types
CCP = "CCP"
Annual = "Annual"
Contract = "Contract"
SFT = "SaaS Free Trial"
HMA = "Hourly and Monthly Agreements"
Hourly = "Hourly"
Monthly = "Monthly"
AFPS = "Annual FPS"
CFPS = "Contract FPS"
CCPFPS = "CCP with FPS"
BYOL = "BYOL"
```

```

Free = "Free"
FTH = "Free Trials and Hourly"

# Define Agreement Term Types
legal = ["LegalTerm"]
config = ["ConfigurableUpfrontPricingTerm"]
usage = ["UsageBasedPricingTerm"]
config_usage = ["ConfigurableUpfrontPricingTerm", "UsageBasedPricingTerm"]
freeTrial = ["FreeTrialPricingTerm"]
recur = ["RecurringPaymentTerm"]
usage_recur = ("UsageBasedPricingTerm", "RecurringPaymentTerm")
fixed_payment = ["FixedUpfrontPricingTerm", "PaymentScheduleTerm"]
fixed_payment_usage = [
    "FixedUpfrontPricingTerm",
    "PaymentScheduleTerm",
    "UsageBasedPricingTerm",
]
byol = ["ByolPricingTerm"]
freeTrial_usage = ("FreeTrialPricingTerm", "UsageBasedPricingTerm")
all_agreement_types_combination = (
    legal,
    config,
    usage,
    config_usage,
    freeTrial,
    recur,
    usage_recur,
    fixed_payment,
    fixed_payment_usage,
    byol,
    freeTrial_usage,
)

# get pricing type method given product type, agreement term type and offer type
if needed
def get_pricing_type(product_type, agreement_term_type, offer_type):
    pricing_types = {
        (SaaSProduct, frozenset(config_usage), frozenset("")): CCP,
        (DataProduct, frozenset(config_usage), frozenset("")): CCP,
        (ContainerProduct, frozenset(config), frozenset(config_usage)): Annual,
        (AmiProduct, frozenset(config), frozenset(config_usage)): Annual,
        (MLProduct, frozenset(config), frozenset(config_usage)): Annual,
        (ContainerProduct, frozenset(config), frozenset(config)): Contract,
    }

```

```

(AmiProduct, frozenset(config), frozenset(config)): Contract,
(SaaSProduct, frozenset(config), frozenset(")): Contract,
(DataProduct, frozenset(config), frozenset(")): Contract,
(AiqProduct, frozenset(config), frozenset(")): Contract,
(ProServiceProduct, frozenset(config), frozenset(")): Contract,
(SaaSProduct, frozenset(freeTrial), frozenset(")): SFT,
(AmiProduct, frozenset(usage_recur), frozenset(")): HMA,
(SaaSProduct, frozenset(usage), frozenset(")): Hourly,
(AmiProduct, frozenset(usage), frozenset(")): Hourly,
(ContainerProduct, frozenset(usage), frozenset(")): Hourly,
(MLProduct, frozenset(usage), frozenset(")): Hourly,
(ContainerProduct, frozenset(recur), frozenset(")): Monthly,
(AmiProduct, frozenset(recur), frozenset(")): Monthly,
(
    ContainerProduct,
    frozenset(fixed_payment),
    frozenset(fixed_payment_usage),
): AFPS,
(AmiProduct, frozenset(fixed_payment), frozenset(fixed_payment_usage)):
AFPS,
(MLProduct, frozenset(fixed_payment), frozenset(")): AFPS,
(ContainerProduct, frozenset(fixed_payment), frozenset(fixed_payment)):
CFPS,
(AmiProduct, frozenset(fixed_payment), frozenset(fixed_payment)): CFPS,
(SaaSProduct, frozenset(fixed_payment), frozenset(")): CFPS,
(DataProduct, frozenset(fixed_payment), frozenset(")): CFPS,
(AiqProduct, frozenset(fixed_payment), frozenset(")): CFPS,
(ProServiceProduct, frozenset(fixed_payment), frozenset(")): CFPS,
(SaaSProduct, frozenset(fixed_payment_usage), frozenset(")): CCPFPS,
(DataProduct, frozenset(fixed_payment_usage), frozenset(")): CCPFPS,
(AiqProduct, frozenset(fixed_payment_usage), frozenset(")): CCPFPS,
(ProServiceProduct, frozenset(fixed_payment_usage), frozenset(")):
CCPFPS,
(AmiProduct, frozenset(byol), frozenset(")): BYOL,
(SaaSProduct, frozenset(byol), frozenset(")): BYOL,
(ProServiceProduct, frozenset(byol), frozenset(")): BYOL,
(AiqProduct, frozenset(byol), frozenset(")): BYOL,
(MLProduct, frozenset(byol), frozenset(")): BYOL,
(ContainerProduct, frozenset(byol), frozenset(")): BYOL,
(DataProduct, frozenset(byol), frozenset(")): BYOL,
(ContainerProduct, frozenset(legal), frozenset(")): Free,
(AmiProduct, frozenset(freeTrial_usage), frozenset(")): FTH,
(ContainerProduct, frozenset(freeTrial_usage), frozenset(")): FTH,
(MLProduct, frozenset(freeTrial_usage), frozenset(")): FTH,

```

```

    }

    key = (product_type, agreement_term_type, offer_type)
    if key in pricing_types:
        return pricing_types[key]
    else:
        return "Unknown"

# Example usage for testing purpose
"""
product_type = SaaSProduct
agreement_term_type = frozenset(config_usage)
offer_type = frozenset('')
pricing_type = get_pricing_type(product_type, agreement_term_type, offer_type)
print("pricing type = " + pricing_type) # Output: CCP
"""

# check if offer term types are needed; if Y, needed
def get_offer_term_type(product_type, agreement_term_type):
    offer_term_types = {
        (ContainerProduct, frozenset(config)): "Y",
        (AmiProduct, frozenset(config)): "Y",
        (ContainerProduct, frozenset(fixed_payment)): "Y",
        (AmiProduct, frozenset(fixed_payment)): "Y",
        (AmiProduct, frozenset(fixed_payment), frozenset(fixed_payment)): "Y",
    }

    key = (product_type, agreement_term_type)
    if key in offer_term_types:
        return offer_term_types[key]
    else:
        return

logger = logging.getLogger(__name__)

def get_agreements(mp_client):
    AgreementSummaryList = []
    # Set PartyType to "Proposer" to return agreements where you are the
    proposer.
    # Change to "Acceptor" to return agreements where you are the acceptor.

```

```

partyTypes = ["Proposer"]
for value in partyTypes:
    try:
        agreement = mp_client.search_agreements(
            catalog=AWSMPCATALOG,
            maxResults=MAX_PAGE_RESULTS,
            filters=[
                {"name": "PartyType", "values": [value]},
                {"name": idType, "values": [idValue]},
                {"name": "AgreementType", "values": ["PurchaseAgreement"]},
            ],
        )
    except ClientError as e:
        logger.error("Could not complete search_agreements request.")
        raise

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    while "nextToken" in agreement and agreement["nextToken"] is not None:
        try:
            agreement = mp_client.search_agreements(
                catalog=AWSMPCATALOG,
                maxResults=MAX_PAGE_RESULTS,
                nextToken=agreement["nextToken"],
                filters=[
                    {"name": "PartyType", "values": [value]},
                    {"name": idType, "values": [idValue]},
                    {"name": "AgreementType", "values":
["PurchaseAgreement"]},
                ],
            )
        except ClientError as e:
            logger.error("Could not complete search_agreements request.")
            raise

        AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    return AgreementSummaryList

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)

```



```
print("Looking for an agreement in the AWS Marketplace Catalog.")
print("-" * 88)

mp_client = boto3.client("marketplace-agreement")

# find all agreements matching the specified idType and idValue
agreements = get_agreements(mp_client)

for item in agreements:
    pricingType = ""
    agreement_id = item["agreementId"]

    # get term types inside offer
    offer_term_types = get_offer_term_types(item)

    # even though multiple product types are allowed for one agreement, only
    need the first one
    productType = item["proposalSummary"]["resources"][0]["type"]

    # get agreement terms types
    agreementTerm = mp_client.get_agreement_terms(agreementId=agreement_id)

    agreementTermTypes = get_agreement_term_types(agreementTerm)

    # match with agreement term type group
    matchedTermType = getMatchedTermTypesCombination(agreementTermTypes)

    # check if offer term type is needed.
    offer_term_type_needed = get_offer_term_type(
        productType, frozenset(matchedTermType)
    )

    # get pricing type given product type, agreement term types and offer
    type if needed;
    # one exception is Container with Legal term. LegalTerm needs to be the
    only term present
    if offer_term_type_needed is not None:
        matchedOfferTermTypes =
getMatchedTermTypesCombination(offer_term_types)
        print(f"matchedOfferTermType = {matchedOfferTermTypes}")
        pricingType = get_pricing_type(
            productType,
            frozenset(matchedTermType),
            frozenset(matchedOfferTermTypes),
```

```

        )
    elif set(matchedTermType) == set(legal):
        pricingType = Free
    else:
        pricingType = get_pricing_type(
            productType, frozenset(matchedTermType), frozenset("")
        )

    print(
        f"agreementId={agreement_id};productType={productType};
        agreementTermTypes={agreementTermTypes}; matchedTermType={matchedTermType};
        offerTermTypeNeeded={offer_term_type_needed};
        offer_term_types={offer_term_types}"
    )
    print(f"pricing type={pricingType}")

def getMatchedTermTypesCombination(agreementTermTypes):
    matchedCombination = ()
    for element in all_agreement_types_combination:
        if check_elements(agreementTermTypes, element):
            matchedCombination = element
    return matchedCombination

def get_offer_term_types(item):
    offer_id = item["proposalSummary"]["offerId"]
    mp_catalogAPI_client = boto3.client("marketplace-catalog")
    offer_document = get_entity_information(mp_catalogAPI_client, offer_id)
    offerDetail = offer_document["Details"]
    offerDetail_json_object = json.loads(offerDetail)
    offer_term_types = [term["Type"] for term in
        offerDetail_json_object["Terms"]]
    return offer_term_types

# make sure all elements in array2 exist in array1
def check_elements(array1, array2):
    for element in array2:
        if element not in array1:
            return False
    return True

```

```

def get_entity_information(mp_client, entity_id):
    """
    Returns information about a given entity
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of entity information
    """

    try:
        response = mp_client.describe_entity(
            Catalog="AWSMarketplace",
            EntityId=entity_id,
        )

        return response

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Entity with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def get_agreement_term_types(agreementTerm):
    types = []
    for term in agreementTerm["acceptedTerms"]:
        for value in term.values():
            if isinstance(value, dict) and "type" in value:
                types.append(value["type"])
    return types

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약의 제품 유형 가져오기

다음 코드 예제에서는 계약의 제품 유형을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Resource;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;

import java.util.ArrayList;
import java.util.List;

import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementProductType {

    /*
     * Obtain the Product Type of the product the agreement was created on
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;
```

```
List<String> productIds = getProducts(agreementId);

ReferenceCodesUtils.formatOutput(productIds);
}

public static List<String> getProducts(String agreementId) {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    DescribeAgreementRequest describeAgreementRequest =
        DescribeAgreementRequest.builder()
            .agreementId(agreementId)
            .build();

    DescribeAgreementResponse describeAgreementResponse =
        marketplaceAgreementClient.describeAgreement(describeAgreementRequest);

    List<String> productIds = new ArrayList<String>();
    for (Resource resource :
        describeAgreementResponse.proposalSummary().resources()) {
        productIds.add(resource.id() + ":" + resource.type());
    }
    return productIds;
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain the Product Type of the product the agreement was created on
AG-11
"""

import logging

import boto3
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

# agreement id
AGREEMENT_ID = "agmt-11111111111111111111111111111111"

def get_agreement_information(mp_client, entity_id):
    """
    Returns information about a given agreement
    Args: entity_id str: Entity to return
    Returns: dict: Dictionary of agreement information
    """

    try:
        agreement = mp_client.describe_agreement(agreementId=entity_id)

        return agreement

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", entity_id)
        else:
            logger.error("Unexpected error: %s", e)

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
```

```

    print("Looking for offer and product details in a given agreement by
    agreement id.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-agreement")

    agreement = get_agreement_information(mp_client, AGREEMENT_ID)

    if agreement is not None:
        for resource in agreement["proposalSummary"]["resources"]:
            print(f"Product ID: {resource['id']} | Product Type:
            {resource['type']}")
        else:
            print("Agreement with ID " + AGREEMENT_ID + " is not found")

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 상태 가져오기

다음 코드 예제에서는 계약 상태를 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;
```

```
import static
    com.example.awsmarketplace.utils.ReferenceCodesConstants.AGREEMENT_ID;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.DescribeAgreementResponse;

public class GetAgreementStatus {

    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        DescribeAgreementResponse describeAgreementResponse =
            getDescribeAgreementResponse(agreementId);

        System.out.println("Agreement status is " +
            describeAgreementResponse.status());

    }

    public static DescribeAgreementResponse getDescribeAgreementResponse(String
        agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        DescribeAgreementRequest describeAgreementRequest =
            DescribeAgreementRequest.builder()
                .agreementId(agreementId)
                .build();

        DescribeAgreementResponse describeAgreementResponse =
            marketplaceAgreementClient.describeAgreement(describeAgreementRequest);
        return describeAgreementResponse;
    }
}
```



```
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [DescribeAgreement](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to get all agreement status
AG-13

Example Usage: python3 get_agreement_status.py --agreement-id <agreement-id>
"""

import argparse
import logging

import boto3
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-agreement")

logger = logging.getLogger(__name__)

def get_agreement(agreement_id):
    try:
        response = mp_client.describe_agreement(agreementId=agreement_id)
        return response
    except ClientError as e:
```

```

        logger.error(f"Could not complete search_agreements request. {e}")

    return None

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--agreement-id",
        "-aid",
        help="Provide agreement ID to describe agreement status",
        required=True,
    )
    args = parser.parse_args()

    response = get_agreement(agreement_id=args.agreement_id)

    if response is not None:
        print(f"Agreement status: {response['status']}")
    else:
        print(f"No agreement found for {args.agreement_id}")

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [DescribeAgreement](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약의 지원 조건 가져오기

다음 코드 예제에서는 계약의 지원 조건을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import software.amazon.awssdk.services.marketplaceagreement.model.AcceptedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.SupportTerm;

import java.util.ArrayList;
import java.util.List;

import static
    com.example.awsmarketplace.utils.ReferenceCodesConstants.AGREEMENT_ID;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class GetAgreementTermsSupportTerm {

    /*
     * Obtain the support and refund policy I have provided to the customer
     */
    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        List<SupportTerm> supportTerms = getSupportTerms(agreementId);

        ReferenceCodesUtils.formatOutput(supportTerms);
    }

    public static List<SupportTerm> getSupportTerms(String agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementTermsRequest getAgreementTermsRequest =
            GetAgreementTermsRequest.builder().agreementId(agreementId)
```

```

        .build();

        GetAgreementTermsResponse getAgreementTermsResponse =
marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);

        List<SupportTerm> supportTerms = new ArrayList<>();

        for (AcceptedTerm acceptedTerm : getAgreementTermsResponse.acceptedTerms()) {
            if (acceptedTerm.supportTerm() != null) {
                supportTerms.add(acceptedTerm.supportTerm());
            }
        }
        return supportTerms;
    }
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)을 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Obtain the support and refund policy I have provided to the customer for an
agreement
AG-19

Example Usage: python3 get_agreement_support_terms.py --agreement-id <agreement-
id>
"""

```

```
import argparse
import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

logger = logging.getLogger(__name__)

mp_client = boto3.client("marketplace-agreement")

def get_agreement_terms(agreement_id):
    try:
        agreement = mp_client.get_agreement_terms(agreementId=agreement_id)
        return agreement

    except ClientError as e:
        if e.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Agreement with ID %s not found.", agreement_id)

        else:
            logger.error("Unexpected error: %s", e)

    return None

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--agreement-id",
        "-aid",
        help="Provide agreement ID to describe agreement status",
        required=True,
    )
    args = parser.parse_args()

    agreement = get_agreement_terms(agreement_id=args.agreement_id)
```

```

if agreement is not None:
    support_found = False

    for term in agreement["acceptedTerms"]:
        if "supportTerm" in term.keys():
            helper.pretty_print_datetime(term)
            support_found = True

    if not support_found:
        print(f"No support term found for agreement: {args.agreement_id}")

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [GetAgreementTerms](#)을 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 조건 가져오기

다음 코드 예제에서는 계약 조건을 가져오는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.seller;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;

```

```

import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementTermsResponse;

public class GetAgreementTerms {

    public static void main(String[] args) {

        String agreementId = args.length > 0 ? args[0] : AGREEMENT_ID;

        GetAgreementTermsResponse getAgreementTermsResponse =
            getAgreementTermsResponse(agreementId);

        ReferenceCodesUtils.formatOutput(getAgreementTermsResponse);

    }

    public static GetAgreementTermsResponse getAgreementTermsResponse(String
        agreementId) {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        GetAgreementTermsRequest getAgreementTermsRequest =
            GetAgreementTermsRequest.builder()
                .agreementId(agreementId)
                .build();

        GetAgreementTermsResponse getAgreementTermsResponse =
            marketplaceAgreementClient.getAgreementTerms(getAgreementTermsRequest);
        return getAgreementTermsResponse;
    }

}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [GetAgreementTerms](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서 이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 취소 요청 나열

다음 코드 예제에서는 수락자로 참여하는 계약에 대한 계약 취소 요청을 나열하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer.agreementCancellation;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AgreementCancellationRequestS
import
    software.amazon.awssdk.services.marketplaceagreement.model.ListAgreementCancellationRequ
import
    software.amazon.awssdk.services.marketplaceagreement.model.ListAgreementCancellationRequ

public class ListAgreementCancellationRequests {

    private static final String PARTY_TYPE = "Proposer";

    public static void main(String[] args) {
        listAgreementCancellationRequests();
    }

    private static void listAgreementCancellationRequests() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();
```



```

String nextToken = null;

do {
    ListAgreementCancellationRequestsRequest request =
        ListAgreementCancellationRequestsRequest.builder()
            .partyType(PARTY_TYPE)
            .nextToken(nextToken)
            .build();

    ListAgreementCancellationRequestsResponse response =
marketplaceAgreementClient.listAgreementCancellationRequests(request);

    for (AgreementCancellationRequestSummary summary : response.items())
    {
        System.out.println("Cancellation Request ID: " +
summary.agreementCancellationRequestId());
        System.out.println("Agreement ID: " + summary.agreementId());
        System.out.println("Status: " + summary.statusAsString());
        System.out.println("Reason Code: " +
summary.reasonCodeAsString());
        System.out.println("Agreement Type: " + summary.agreementType());
        System.out.println("Catalog: " + summary.catalog());
        System.out.println("Created At: " + summary.createdAt());
        System.out.println("Updated At: " + summary.updatedAt());
        System.out.println("---");
    }

    nextToken = response.nextToken();
} while (nextToken != null);
}
}

```

- API 세부 정보는 API 참조의 [ListAgreementCancellationRequests](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
    MarketplaceAgreementClient,
    ListAgreementCancellationRequestsCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const PARTY_TYPE = "Proposer";

async function listAgreementCancellationRequests() {
    const client = new MarketplaceAgreementClient();

    let nextToken = null;

    do {
        const response = await client.send(
            new ListAgreementCancellationRequestsCommand({
                partyType: PARTY_TYPE,
                nextToken: nextToken,
            })
        );

        for (const summary of response.items) {
            console.log("Cancellation Request ID: " +
                summary.agreementCancellationRequestId);
            console.log("Agreement ID: " + summary.agreementId);
            console.log("Status: " + summary.status);
            console.log("Reason Code: " + summary.reasonCode);
            console.log("Agreement Type: " + summary.agreementType);
            console.log("Catalog: " + summary.catalog);
            console.log("Created At: " + summary.createdAt);
            console.log("Updated At: " + summary.updatedAt);
            console.log("---");
        }
    }
}
```

```

        nextToken = response.nextToken;
    } while (nextToken != null);
}

listAgreementCancellationRequests();

```

- API 세부 정보는 API 참조의 [ListAgreementCancellationRequests](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class ListAgreementCancellationRequests:

    PARTY_TYPE = "Proposer"

    @staticmethod
    def list_agreement_cancellation_requests():
        client = boto3.client("marketplace-agreement")

        next_token = None

        while True:
            kwargs = {"partyType": ListAgreementCancellationRequests.PARTY_TYPE}

```

```

        if next_token:
            kwargs["nextToken"] = next_token

        response = client.list_agreement_cancellation_requests(**kwargs)

        for summary in response.get("items", []):
            print("Cancellation Request ID: " +
                  summary["agreementCancellationRequestId"])
            print("Agreement ID: " + summary["agreementId"])
            print("Status: " + str(summary.get("status", "")))
            print("Reason Code: " + str(summary.get("reasonCode", "")))
            print("Agreement Type: " + str(summary.get("agreementType", "")))
            print("Catalog: " + str(summary.get("catalog", "")))
            print("Created At: " + str(summary.get("createdAt", "")))
            print("Updated At: " + str(summary.get("updatedAt", "")))
            print("---")

        next_token = response.get("nextToken")
        if not next_token:
            break

if __name__ == "__main__":
    ListAgreementCancellationRequests.list_agreement_cancellation_requests()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [ListAgreementCancellationRequests](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 결제 요청 나열

다음 코드 예제에서는 수락자로 참여하는 계약에 대한 계약 결제 요청을 나열하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer.paymentRequest;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.ListAgreementPaymentRequestsR
import
    software.amazon.awssdk.services.marketplaceagreement.model.ListAgreementPaymentRequestsR
import
    software.amazon.awssdk.services.marketplaceagreement.model.PaymentRequestSummary;

public class ListAgreementPaymentRequests {

    private static final String PARTY_TYPE = "Proposer";

    public static void main(String[] args) {
        listAgreementPaymentRequests();
    }

    private static void listAgreementPaymentRequests() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        String nextToken = null;

        do {
            ListAgreementPaymentRequestsRequest request =
```

```

        ListAgreementPaymentRequestsRequest.builder()
            .partyType(PARTY_TYPE)
            .nextToken(nextToken)
            .build();

        ListAgreementPaymentRequestsResponse response =
marketplaceAgreementClient.listAgreementPaymentRequests(request);

        for (PaymentRequestSummary summary : response.items()) {
            System.out.println("Payment Request ID: " +
summary.paymentRequestId());
            System.out.println("Agreement ID: " + summary.agreementId());
            System.out.println("Status: " + summary.statusAsString());
            System.out.println("Name: " + summary.name());
            System.out.println("Charge ID: " + summary.chargeId());
            System.out.println("Charge Amount: " + summary.chargeAmount());
            System.out.println("Currency Code: " + summary.currencyCode());
            System.out.println("Created At: " + summary.createdAt());
            System.out.println("Updated At: " + summary.updatedAt());
            System.out.println("---");
        }

        nextToken = response.nextToken();
    } while (nextToken != null);
}
}

```

- API 세부 정보는 API 참조의 [ListAgreementPaymentRequests](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
  MarketplaceAgreementClient,
  ListAgreementPaymentRequestsCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const PARTY_TYPE = "Proposer";

async function listAgreementPaymentRequests() {
  const client = new MarketplaceAgreementClient();

  let nextToken = null;

  do {
    const response = await client.send(
      new ListAgreementPaymentRequestsCommand({
        partyType: PARTY_TYPE,
        nextToken: nextToken,
      })
    );

    for (const summary of response.items) {
      console.log("Payment Request ID: " + summary.paymentRequestId);
      console.log("Agreement ID: " + summary.agreementId);
      console.log("Status: " + summary.status);
      console.log("Name: " + summary.name);
      console.log("Charge ID: " + summary.chargeId);
      console.log("Charge Amount: " + summary.chargeAmount);
      console.log("Currency Code: " + summary.currencyCode);
      console.log("Created At: " + summary.createdAt);
      console.log("Updated At: " + summary.updatedAt);
      console.log("---");
    }

    nextToken = response.nextToken;
  } while (nextToken != null);
}

listAgreementPaymentRequests();
```

- API 세부 정보는 API 참조의 [ListAgreementPaymentRequests](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class ListAgreementPaymentRequests:

    PARTY_TYPE = "Proposer"

    @staticmethod
    def list_agreement_payment_requests():
        client = boto3.client("marketplace-agreement")

        next_token = None

        while True:
            kwargs = {"partyType": ListAgreementPaymentRequests.PARTY_TYPE}
            if next_token:
                kwargs["nextToken"] = next_token

            response = client.list_agreement_payment_requests(**kwargs)

            for summary in response.get("items", []):
                print("Payment Request ID: " + summary["paymentRequestId"])
                print("Agreement ID: " + summary["agreementId"])
                print("Status: " + str(summary.get("status", "")))
                print("Name: " + str(summary.get("name", "")))
                print("Charge ID: " + str(summary.get("chargeId", "")))
                print("Charge Amount: " + str(summary.get("chargeAmount", "")))
```



```

        print("Currency Code: " + str(summary.get("currencyCode", "")))
        print("Created At: " + str(summary.get("createdAt", "")))
        print("Updated At: " + str(summary.get("updatedAt", "")))
        print("---")

    next_token = response.get("nextToken")
    if not next_token:
        break

if __name__ == "__main__":
    ListAgreementPaymentRequests.list_agreement_payment_requests()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [ListAgreementPaymentRequests](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 취소 요청 거부

다음 코드 예제에서는 판매자가 시작한 계약 취소 요청을 거부하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.buyer.agreementCancellation;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;

```

```

import
software.amazon.awssdk.services.marketplaceagreement.model.RejectAgreementCancellationRe
import
software.amazon.awssdk.services.marketplaceagreement.model.RejectAgreementCancellationRe

public class RejectAgreementCancellationRequest {

    private static final String AGREEMENT_ID = "<AGREEMENT ID HERE>";
    private static final String AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT
CANCELLATION REQUEST ID HERE>";
    private static final String REJECTION_REASON = "<REJECTION REASON HERE>";

    public static void main(String[] args) {
        rejectAgreementCancellationRequest();
    }

    private static void rejectAgreementCancellationRequest() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        RejectAgreementCancellationRequestRequest request =
            RejectAgreementCancellationRequestRequest.builder()
                .agreementId(AGREEMENT_ID)

.agreementCancellationRequestId(AGREEMENT_CANCELLATION_REQUEST_ID)
                .rejectionReason(REJECTION_REASON)
                .build();

        RejectAgreementCancellationRequestResponse response =

marketplaceAgreementClient.rejectAgreementCancellationRequest(request);

        System.out.println("Agreement ID: " + response.agreementId());
        System.out.println("Cancellation Request ID: " +
response.agreementCancellationRequestId());
        System.out.println("Status: " + response.statusAsString());
        System.out.println("Status Message: " + response.statusMessage());
        System.out.println("Reason Code: " + response.reasonCodeAsString());
        System.out.println("Created At: " + response.createdAt());
        System.out.println("Updated At: " + response.updatedAt());
    }
}

```

```
}

```

- API 세부 정보는 API 참조의 [RejectAgreementCancellationRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
    MarketplaceAgreementClient,
    RejectAgreementCancellationRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const AGREEMENT_ID = "<AGREEMENT ID HERE>";
const AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT CANCELLATION REQUEST ID HERE>";
const REJECTION_REASON = "<REJECTION REASON HERE>";

async function rejectAgreementCancellationRequest() {
    const client = new MarketplaceAgreementClient();

    const response = await client.send(
        new RejectAgreementCancellationRequestCommand({
            agreementId: AGREEMENT_ID,
            agreementCancellationRequestId: AGREEMENT_CANCELLATION_REQUEST_ID,
            rejectionReason: REJECTION_REASON,
        })
    );

    console.log("Agreement ID: " + response.agreementId);
    console.log("Cancellation Request ID: " + response.agreementCancellationRequestId);
    console.log("Status: " + response.status);
}
```

```

    console.log("Status Message: " + response.statusMessage);
    console.log("Reason Code: " + response.reasonCode);
    console.log("Created At: " + response.createdAt);
    console.log("Updated At: " + response.updatedAt);
  }

  rejectAgreementCancellationRequest();

```

- API 세부 정보는 API 참조의 [RejectAgreementCancellationRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class RejectAgreementCancellationRequest:

    AGREEMENT_ID = "<AGREEMENT ID HERE>"
    AGREEMENT_CANCELLATION_REQUEST_ID = "<AGREEMENT CANCELLATION REQUEST ID HERE>"
    REJECTION_REASON = "<REJECTION REASON HERE>"

    @staticmethod
    def reject_agreement_cancellation_request():
        client = boto3.client("marketplace-agreement")

```

```

        response = client.reject_agreement_cancellation_request(
            agreementId=RejectAgreementCancellationRequest.AGREEMENT_ID,
            agreementCancellationRequestId=RejectAgreementCancellationRequest.AGREEMENT_CANCELLATION_ID,
            rejectionReason=RejectAgreementCancellationRequest.REJECTION_REASON,
        )

        print("Agreement ID: " + response["agreementId"])
        print("Cancellation Request ID: " +
              response["agreementCancellationRequestId"])
        print("Status: " + str(response.get("status", "")))
        print("Status Message: " + str(response.get("statusMessage", "")))
        print("Reason Code: " + str(response.get("reasonCode", "")))
        print("Created At: " + str(response.get("createdAt", "")))
        print("Updated At: " + str(response.get("updatedAt", "")))

if __name__ == "__main__":
    RejectAgreementCancellationRequest.reject_agreement_cancellation_request()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [RejectAgreementCancellationRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 결제 요청 거부

다음 코드 예제에서는 판매자가 시작한 계약 결제 요청을 거부하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.buyer.paymentRequest;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RejectAgreementPaymentRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RejectAgreementPaymentRequestResponse;

public class RejectAgreementPaymentRequest {

    private static final String AGREEMENT_ID = "<AGREEMENT ID HERE>";
    private static final String PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>";
    private static final String REJECTION_REASON = "<REJECTION REASON HERE>";

    public static void main(String[] args) {
        rejectAgreementPaymentRequest();
    }

    private static void rejectAgreementPaymentRequest() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        RejectAgreementPaymentRequest request =
            RejectAgreementPaymentRequest.builder()
                .agreementId(AGREEMENT_ID)
                .paymentRequestId(PAYMENT_REQUEST_ID)
                .rejectionReason(REJECTION_REASON)
                .build();

        RejectAgreementPaymentRequestResponse response =
            marketplaceAgreementClient.rejectAgreementPaymentRequest(request);

        System.out.println("Payment Request ID: " + response.paymentRequestId());
        System.out.println("Agreement ID: " + response.agreementId());
        System.out.println("Status: " + response.statusAsString());
        System.out.println("Status Message: " + response.statusMessage());
    }
}

```

```

        System.out.println("Name: " + response.name());
        System.out.println("Charge Amount: " + response.chargeAmount());
        System.out.println("Currency Code: " + response.currencyCode());
        System.out.println("Created At: " + response.createdAt());
        System.out.println("Updated At: " + response.updatedAt());
    }
}

```

- API 세부 정보는 API 참조의 [RejectAgreementPaymentRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

const {
    MarketplaceAgreementClient,
    RejectAgreementPaymentRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");

const AGREEMENT_ID = "<AGREEMENT ID HERE>";
const PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>";
const REJECTION_REASON = "<REJECTION REASON HERE>";

async function rejectAgreementPaymentRequest() {
    const client = new MarketplaceAgreementClient();

    const response = await client.send(
        new RejectAgreementPaymentRequestCommand({
            agreementId: AGREEMENT_ID,
            paymentRequestId: PAYMENT_REQUEST_ID,
            rejectionReason: REJECTION_REASON,
        })
    );
}

```

```

    console.log("Payment Request ID: " + response.paymentRequestId);
    console.log("Agreement ID: " + response.agreementId);
    console.log("Status: " + response.status);
    console.log("Status Message: " + response.statusMessage);
    console.log("Name: " + response.name);
    console.log("Charge Amount: " + response.chargeAmount);
    console.log("Currency Code: " + response.currencyCode);
    console.log("Created At: " + response.createdAt);
    console.log("Updated At: " + response.updatedAt);
}

rejectAgreementPaymentRequest();

```

- API 세부 정보는 API 참조의 [RejectAgreementPaymentRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

import sys
import os

sys.path.insert(0, os.path.join(os.path.dirname(__file__), ".."))

import boto3

class RejectAgreementPaymentRequest:

    AGREEMENT_ID = "<AGREEMENT ID HERE>"
    PAYMENT_REQUEST_ID = "<PAYMENT REQUEST ID HERE>"
    REJECTION_REASON = "<REJECTION REASON HERE>"

```



```

@staticmethod
def reject_agreement_payment_request():
    client = boto3.client("marketplace-agreement")

    response = client.reject_agreement_payment_request(
        agreementId=RejectAgreementPaymentRequest.AGREEMENT_ID,
        paymentRequestId=RejectAgreementPaymentRequest.PAYMENT_REQUEST_ID,
        rejectionReason=RejectAgreementPaymentRequest.REJECTION_REASON,
    )

    print("Payment Request ID: " + response["paymentRequestId"])
    print("Agreement ID: " + response["agreementId"])
    print("Status: " + str(response.get("status", "")))
    print("Status Message: " + str(response.get("statusMessage", "")))
    print("Name: " + str(response.get("name", "")))
    print("Charge Amount: " + str(response.get("chargeAmount", "")))
    print("Currency Code: " + str(response.get("currencyCode", "")))
    print("Created At: " + str(response.get("createdAt", "")))
    print("Updated At: " + str(response.get("updatedAt", "")))

if __name__ == "__main__":
    RejectAgreementPaymentRequest.reject_agreement_payment_request()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [RejectAgreementPaymentRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 계약 계약을 계약 기반 제안으로 대체

다음 코드 예제에서는 SaaS 계약 계약을 새 계약 기반 제안으로 대체하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to replace an existing SaaS agreement with CONTRACT pricing
 * model with a new
 * Agreement-Based Offer (ABO) using the AWS Marketplace Agreement Service APIs.
 *
 * <p>Scenario: A buyer has an active SaaS agreement with CONTRACT pricing model
 * and receives a private
 * Agreement-Based Offer from the seller. The buyer replaces the existing
 * agreement
 * with the new ABO offer, which may include updated pricing terms and payment
 * schedule.
 *
 * <p>Note: Unlike other Replace samples, this example starts from an already-
 * active
```

```

* {@code EXISTING_AGREEMENT_ID} rather than creating a new agreement first. Use
this
* pattern when you already have an agreement ID and want to replace it directly.
*
* <p>Before running this sample, replace the placeholder constants below with
values from
* your AWS Marketplace offers:
* <ul>
*   <li>{@code EXISTING_AGREEMENT_ID} – the agreement ID of the active agreement
to replace (starts with {@code agmt-}).</li>
*   <li>{@code NEW_AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from
the new ABO offer.</li>
*   <li>Term IDs (starting with {@code term-}) – found in the new offer's term
list.</li>
* </ul>
*/
public class ReplaceSaaSContractWithAgreementBasedOffer {

    // The agreement ID of the active SaaS agreement with CONTRACT pricing model
    to replace (starts with "agmt-").
    private static final String EXISTING_AGREEMENT_ID = "<your-existing-
agreement-id>";

    // The agreementProposalId from the new Agreement-Based Offer.
    private static final String NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-
agreement-proposal-identifier>";

    // Term ID for the LegalTerm in the new offer.
    private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

    // Term ID for the ValidityTerm in the new offer.
    private static final String VALIDITY_TERM_ID = "<your-validity-term-id>";

    // Term ID for the FixedUpfrontPricingTerm in the new offer.
    private static final String FIXED_UPFRONT_PRICING_TERM_ID = "<your-fixed-
upfront-pricing-term-id>";

    // Term ID for the PaymentScheduleTerm in the new offer.
    private static final String PAYMENT_SCHEDULE_TERM_ID = "<your-payment-
schedule-term-id>";

    public static void main(String[] args) {
        replaceExistingAgreement();
    }

```

```

/**
 * Replaces an existing SaaS agreement with CONTRACT pricing model with a new
 Agreement-Based Offer.
 * Uses Intent.REPLACE with sourceAgreementIdentifier set to the existing
 agreement ID.
 */
private static void replaceExistingAgreement() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    RequestedTerm legalTerm =
        RequestedTerm.builder().id(LEGAL_TERM_ID).build();
    RequestedTerm validityTerm =
        RequestedTerm.builder().id(VALIDITY_TERM_ID).build();
    RequestedTerm fixedUpfrontPricingTerm =
        RequestedTerm.builder().id(FIXED_UPFRONT_PRICING_TERM_ID).build();
    RequestedTerm paymentScheduleTerm =
        RequestedTerm.builder().id(PAYMENT_SCHEDULE_TERM_ID).build();

    // Replace the agreement with the new offer
    CreateAgreementRequestRequest createAgreementRequestRequest =
        CreateAgreementRequestRequest.builder()
            .clientToken(AgreementApiUtils.generateClientToken())
            .intent(Intent.REPLACE)
            .requestedTerms(fixedUpfrontPricingTerm,
                paymentScheduleTerm, legalTerm, validityTerm)

            .agreementProposalIdentifier(NEW_AGREEMENT_PROPOSAL_IDENTIFIER)
            .sourceAgreementIdentifier(EXISTING_AGREEMENT_ID)
            .build();

    CreateAgreementRequestResponse createAgreementRequestResponse =
        marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
    System.out.println("Replace agreement request created.
        AgreementRequestId: " + createAgreementRequestResponse.agreementRequestId());

    AcceptAgreementRequestRequest acceptAgreementRequestRequest =
        AcceptAgreementRequestRequest.builder()

            .agreementRequestId(createAgreementRequestResponse.agreementRequestId())

```

```

        .build();
        AcceptAgreementRequestResponse acceptAgreementRequestResponse =
marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
        System.out.println("Agreement replaced with AB0. New AgreementId: " +
acceptAgreementRequestResponse.agreementId());
    }
}

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

const {
    MarketplaceAgreementClient,
    CreateAgreementRequestCommand,
    AcceptAgreementRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken } = require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to replace an existing SaaS agreement with CONTRACT pricing
 * model with a new
 * Agreement-Based Offer (ABO) using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: A buyer has an active SaaS agreement with CONTRACT pricing model and
 * receives a private
 * Agreement-Based Offer from the seller. The buyer replaces the existing
 * agreement
 * with the new ABO offer, which may include updated pricing terms and payment
 * schedule.
 *
 * Note: Unlike other Replace samples, this example starts from an already-active

```

```

* EXISTING_AGREEMENT_ID rather than creating a new agreement first. Use this
* pattern when you already have an agreement ID and want to replace it directly.
*
* Before running this sample, replace the placeholder constants below with
values from
* your AWS Marketplace offers:
*   - EXISTING_AGREEMENT_ID – the agreement ID of the active agreement to
replace (starts with "agmt-").
*   - NEW_AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the new
ABO offer.
*   - Term IDs (starting with "term-") – found in the new offer's term list.
*/

// The agreement ID of the active SaaS agreement with CONTRACT pricing model to
replace (starts with "agmt-").
const EXISTING_AGREEMENT_ID = "<your-existing-agreement-id>";

// The agreementProposalId from the new Agreement-Based Offer.
const NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-agreement-proposal-
identifier>";

// Term ID for the LegalTerm in the new offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// Term ID for the ValidityTerm in the new offer.
const VALIDITY_TERM_ID = "<your-validity-term-id>";

// Term ID for the FixedUpfrontPricingTerm in the new offer.
const FIXED_UPFRONT_PRICING_TERM_ID = "<your-fixed-upfront-pricing-term-id>";

// Term ID for the PaymentScheduleTerm in the new offer.
const PAYMENT_SCHEDULE_TERM_ID = "<your-payment-schedule-term-id>";

/**
 * Replaces an existing SaaS agreement with CONTRACT pricing model with a new
Agreement-Based Offer.
 * Uses Intent.REPLACE with sourceAgreementIdentifier set to the existing
agreement ID.
 */
async function replaceExistingAgreement() {
    const client = new MarketplaceAgreementClient();

    const legalTerm = { id: LEGAL_TERM_ID };
    const validityTerm = { id: VALIDITY_TERM_ID };

```

```

const fixedUpfrontPricingTerm = { id: FIXED_UPFRONT_PRICING_TERM_ID };
const paymentScheduleTerm = { id: PAYMENT_SCHEDULE_TERM_ID };

// Replace the agreement with the new offer
const createAgreementRequestResponse = await client.send(
  new CreateAgreementRequestCommand({
    clientToken: generateClientToken(),
    intent: "REPLACE",
    requestedTerms: [fixedUpfrontPricingTerm, paymentScheduleTerm,
legalTerm, validityTerm],
    agreementProposalIdentifier: NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
    sourceAgreementIdentifier: EXISTING_AGREEMENT_ID,
  })
);
console.log("Replace agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId);

const acceptAgreementRequestResponse = await client.send(
  new AcceptAgreementRequestCommand({
    agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
  })
);
console.log("Agreement replaced with AB0. New AgreementId: " +
acceptAgreementRequestResponse.agreementId);
}

replaceExistingAgreement();

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
"""
```

Demonstrates how to replace an existing SaaS agreement with CONTRACT pricing model with a new Agreement-Based Offer (ABO) using the AWS Marketplace Agreement Service APIs.

Scenario: A buyer has an active SaaS agreement with CONTRACT pricing model and receives a private Agreement-Based Offer from the seller. The buyer replaces the existing agreement with the new ABO offer, which may include updated pricing terms and payment schedule.

Note: Unlike other Replace samples, this example starts from an already-active EXISTING_AGREEMENT_ID rather than creating a new agreement first. Use this pattern when you already have an agreement ID and want to replace it directly.

Before running this sample, replace the placeholder constants below with values from

your AWS Marketplace offers:

- EXISTING_AGREEMENT_ID – the agreement ID of the active agreement to replace (starts with agmt-).
- NEW_AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the new ABO offer.
- Term IDs (starting with term-) – found in the new offer's term list.

```
"""
```

```
import boto3
```

```
from utils.agreement_api_utils import generate_client_token
```

```
class ReplaceSaaSContractWithAgreementBasedOffer:
```

```
    # The agreement ID of the active SaaS agreement with CONTRACT pricing model
    # to replace (starts with "agmt-").
```

```
    EXISTING_AGREEMENT_ID = "<your-existing-agreement-id>"
```

```
    # The agreementProposalId from the new Agreement-Based Offer.
```

```
    NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-agreement-proposal-
    identifier>"
```

```
    # Term ID for the LegalTerm in the new offer.
```

```
    LEGAL_TERM_ID = "<your-legal-term-id>"
```



```

# Term ID for the ValidityTerm in the new offer.
VALIDITY_TERM_ID = "<your-validity-term-id>"

# Term ID for the FixedUpfrontPricingTerm in the new offer.
FIXED_UPFRONT_PRICING_TERM_ID = "<your-fixed-upfront-pricing-term-id>"

# Term ID for the PaymentScheduleTerm in the new offer.
PAYMENT_SCHEDULE_TERM_ID = "<your-payment-schedule-term-id>"

@staticmethod
def replace_existing_agreement():
    """
    Replace an existing SaaS agreement with CONTRACT pricing model with a new
    Agreement-Based Offer.

    Uses Intent.REPLACE with sourceAgreementIdentifier set to the existing
    agreement ID.
    """
    client = boto3.client("marketplace-agreement")
    cls = ReplaceSaaSContractWithAgreementBasedOffer

    legal_term = {"id": cls.LEGAL_TERM_ID}
    validity_term = {"id": cls.VALIDITY_TERM_ID}
    fixed_upfront_pricing_term = {"id": cls.FIXED_UPFRONT_PRICING_TERM_ID}
    payment_schedule_term = {"id": cls.PAYMENT_SCHEDULE_TERM_ID}

    # Replace the agreement with the new offer
    create_response = client.create_agreement_request(
        clientToken=generate_client_token(),
        intent="REPLACE",
        requestedTerms=[fixed_upfront_pricing_term, payment_schedule_term,
legal_term, validity_term],
        agreementProposalIdentifier=cls.NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
        sourceAgreementIdentifier=cls.EXISTING_AGREEMENT_ID,
    )
    agreement_request_id = create_response["agreementRequestId"]
    print("Replace agreement request created. AgreementRequestId: " +
agreement_request_id)

    accept_response = client.accept_agreement_request(
        agreementRequestId=agreement_request_id
    )
    print("Agreement replaced with AB0. New AgreementId: " +
accept_response["agreementId"])

```

```
if __name__ == "__main__":
    ReplaceSaaSContractWithAgreementBasedOffer.replace_existing_agreement()
```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 무료 평가판을 소비 요금으로 계약으로 교체

다음 코드 예제에서는 SaaS 무료 평가판을 사용 요금 계약(CCP)으로 대체하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse
```

```

import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementEntitlementsResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to create a SaaS free trial agreement and then replace it
 * with a
 * paid Contract with Consumption Pricing (CCP) offer using the AWS Marketplace
 * Agreement Service APIs.
 *
 * <p>Scenario: A buyer first starts a free trial on a SaaS product. Once the
 * trial is active,
 * they decide to convert by replacing it with a paid offer that includes a
 * FixedUpfrontPricingTerm, PaymentScheduleTerm, and UsageBasedPricingTerm.
 *
 * <p>Flow:
 * <ol>
 *   <li>Create a SaaS free trial agreement with freeTrialPricingTerm.</li>
 *   <li>Wait for freeTrialPricingTerm agreement entitlements to become active.</li>
 *   <li>Replace the free trial agreement with the paid CCP offer using
 * Intent.REPLACE.</li>
 * </ol>
 *
 * <p>Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offers:
 * <ul>
 *   <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
 * free trial offer.</li>
 *   <li>{@code NEW_AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from
 * the paid CCP offer.</li>
 *   <li>Term IDs (starting with {@code term-}) – found in each offer's term
 * list.</li>
 * </ul>
 */
public class ReplaceSaaSFreeTrialWithCCP {

    // The agreementProposalId from the free trial offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-free-
trial-agreement-proposal-identifier>";

```

```

// Term ID for the FreeTrialPricingTerm in the free trial offer.
private static final String FREE_TRIAL_PRICING_TERM_ID = "<your-free-trial-pricing-term-id>";

// Term ID for the LegalTerm in the free trial offer.
private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

// The agreementProposalId from the paid CCP offer to convert to.
private static final String NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-ccp-agreement-proposal-identifier>";

// Term ID for the FixedUpfrontPricingTerm in the CCP offer.
private static final String FIXED_UPFRONT_PRICING_TERM_ID = "<your-fixed-upfront-pricing-term-id>";

// Term ID for the PaymentScheduleTerm in the CCP offer.
private static final String PAYMENT_SCHEDULE_TERM_ID = "<your-payment-schedule-term-id>";

// Term ID for the UsageBasedPricingTerm in the CCP offer.
private static final String USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-pricing-term-id>";

// Term ID for the ValidityTerm in the CCP offer.
private static final String VALIDITY_TERM_ID = "<your-validity-term-id>";

// Term ID for the LegalTerm in the CCP offer.
private static final String NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>";

public static void main(String[] args) {
    createSaaSFreeTrialAndReplaceWithCCP();
}

/**
 * Full end-to-end flow:
 * 1. Create a SaaS free trial agreement with freeTrialPricingTerm.
 * 2. Wait for freeTrialPricingTerm agreement entitlements to become active.
 * 3. Replace the free trial agreement with the paid CCP offer.
 */
private static void createSaaSFreeTrialAndReplaceWithCCP() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())

```

```

        .build();

        RequestedTerm legalTerm =
        RequestedTerm.builder().id(LEGAL_TERM_ID).build();
        RequestedTerm freeTrialPricingTerm =
        RequestedTerm.builder().id(FREE_TRIAL_PRICING_TERM_ID).build();

        // --- Step 1: Agreement with freeTrialPricingTerm ---
        CreateAgreementRequestRequest createAgreementRequestRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.NEW)
                .requestedTerms(freeTrialPricingTerm, legalTerm)

        .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
            .build();
        CreateAgreementRequestResponse createAgreementRequestResponse =

        marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
        System.out.println("Agreement request with freeTrialPricingTerm created.
        AgreementRequestId: " + createAgreementRequestResponse.agreementRequestId());

        AcceptAgreementRequestRequest acceptAgreementRequestRequest =
            AcceptAgreementRequestRequest.builder()

        .agreementRequestId(createAgreementRequestResponse.agreementRequestId())
            .build();
        AcceptAgreementRequestResponse acceptAgreementRequestResponse =

        marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
        System.out.println("Agreement request with freeTrialPricingTerm accepted.
        AgreementId: " + acceptAgreementRequestResponse.agreementId());

        // Wait for freeTrialPricingTerm agreement entitlements to become active
        before replacing.
        System.out.println("Waiting for freeTrialPricingTerm agreement
        entitlements to become active...");
        GetAgreementEntitlementsResponse entitlementsResponse =
        AgreementApiUtils.pollUntilEntitlementsAvailable(
            marketplaceAgreementClient,
            acceptAgreementRequestResponse.agreementId());
        System.out.println("freeTrialPricingTerm agreement entitlements are now
        active.");
        AgreementApiUtils.formatOutput(entitlementsResponse);

```

```

        // --- Step 2: Replace Agreement with freeTrialPricingTerm with Paid CCP
        Offer ---
        // Use Intent.REPLACE and sourceAgreementIdentifier to replace the free
        trial agreement.
        RequestedTerm usageBasedPricingTerm =
        RequestedTerm.builder().id(USAGE_BASED_PRICING_TERM_ID).build();
        RequestedTerm fixedUpfrontPricingTerm =
        RequestedTerm.builder().id(FIXED_UPFRONT_PRICING_TERM_ID).build();
        RequestedTerm paymentScheduleTerm =
        RequestedTerm.builder().id(PAYMENT_SCHEDULE_TERM_ID).build();
        RequestedTerm validityTerm =
        RequestedTerm.builder().id(VALIDITY_TERM_ID).build();
        RequestedTerm newLegalTerm =
        RequestedTerm.builder().id(NEW_LEGAL_TERM_ID).build();

        CreateAgreementRequestRequest carRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.REPLACE)
                .requestedTerms(usageBasedPricingTerm,
fixedUpfrontPricingTerm, paymentScheduleTerm, validityTerm, newLegalTerm)

                .agreementProposalIdentifier(NEW_AGREEMENT_PROPOSAL_IDENTIFIER)

                .sourceAgreementIdentifier(acceptAgreementRequestResponse.agreementId())
                    .build();
        CreateAgreementRequestResponse carResponse =
            marketplaceAgreementClient.createAgreementRequest(carRequest);
        System.out.println("Replace agreement request created.
AgreementRequestId: " + carResponse.agreementRequestId());

        AcceptAgreementRequestRequest aarRequest =
            AcceptAgreementRequestRequest.builder()
                .agreementRequestId(carResponse.agreementRequestId())
                    .build();
        AcceptAgreementRequestResponse aarResponse =
            marketplaceAgreementClient.acceptAgreementRequest(aarRequest);
        System.out.println("Agreement with freeTrialPricingTerm replaced with
paid CCP offer. New AgreementId: " + aarResponse.agreementId());
    }
}

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
    MarketplaceAgreementClient,
    CreateAgreementRequestCommand,
    AcceptAgreementRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken, formatOutput, pollUntilEntitlementsAvailable } =
    require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to create a SaaS free trial agreement and then replace it
 * with a
 * paid Contract with Consumption Pricing (CCP) offer using the AWS Marketplace
 * Agreement Service APIs.
 *
 * Scenario: A buyer first starts a free trial on a SaaS product. Once the trial
 * is active,
 * they decide to convert by replacing it with a paid offer that includes a
 * FixedUpfrontPricingTerm, PaymentScheduleTerm, and UsageBasedPricingTerm.
 *
 * Flow:
 * 1. Create a SaaS free trial agreement with freeTrialPricingTerm.
 * 2. Wait for freeTrialPricingTerm agreement entitlements to become active.
 * 3. Replace the free trial agreement with the paid CCP offer using
 * Intent.REPLACE.
 *
 * Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offers:
 * - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the free
 * trial offer.
```

```
*   - NEW_AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the paid
    CCP offer.
*   - Term IDs (starting with "term-") – found in each offer's term list.
*/

// The agreementProposalId from the free trial offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-free-trial-agreement-proposal-
identifier>";

// Term ID for the FreeTrialPricingTerm in the free trial offer.
const FREE_TRIAL_PRICING_TERM_ID = "<your-free-trial-pricing-term-id>";

// Term ID for the LegalTerm in the free trial offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// The agreementProposalId from the paid CCP offer to convert to.
const NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-ccp-agreement-proposal-
identifier>";

// Term ID for the FixedUpfrontPricingTerm in the CCP offer.
const FIXED_UPFRONT_PRICING_TERM_ID = "<your-fixed-upfront-pricing-term-id>";

// Term ID for the PaymentScheduleTerm in the CCP offer.
const PAYMENT_SCHEDULE_TERM_ID = "<your-payment-schedule-term-id>";

// Term ID for the UsageBasedPricingTerm in the CCP offer.
const USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-pricing-term-id>";

// Term ID for the ValidityTerm in the CCP offer.
const VALIDITY_TERM_ID = "<your-validity-term-id>";

// Term ID for the LegalTerm in the CCP offer.
const NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>";

/**
 * Full end-to-end flow:
 * 1. Create a SaaS free trial agreement with freeTrialPricingTerm.
 * 2. Wait for freeTrialPricingTerm agreement entitlements to become active.
 * 3. Replace the free trial agreement with the paid CCP offer.
 */
async function createSaaSFreeTrialAndReplaceWithCCP() {
    const client = new MarketplaceAgreementClient();

    const legalTerm = { id: LEGAL_TERM_ID };
```



```
const freeTrialPricingTerm = { id: FREE_TRIAL_PRICING_TERM_ID };

// --- Step 1: Agreement with freeTrialPricingTerm ---
const createAgreementRequestResponse = await client.send(
  new CreateAgreementRequestCommand({
    clientToken: generateClientToken(),
    intent: "NEW",
    requestedTerms: [freeTrialPricingTerm, legalTerm],
    agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
  })
);
console.log("Agreement request with freeTrialPricingTerm created.
AgreementRequestId: " + createAgreementRequestResponse.agreementRequestId);

const acceptAgreementRequestResponse = await client.send(
  new AcceptAgreementRequestCommand({
    agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
  })
);
console.log("Agreement request with freeTrialPricingTerm accepted.
AgreementId: " + acceptAgreementRequestResponse.agreementId);

// Wait for freeTrialPricingTerm agreement entitlements to become active
before replacing.
console.log("Waiting for freeTrialPricingTerm agreement entitlements to
become active...");
const entitlementsResponse = await pollUntilEntitlementsAvailable(client,
acceptAgreementRequestResponse.agreementId);
console.log("freeTrialPricingTerm agreement entitlements are now active.");
formatOutput(entitlementsResponse);

// --- Step 2: Replace Agreement with freeTrialPricingTerm with Paid CCP
Offer ---
// Use Intent.REPLACE and sourceAgreementIdentifier to replace the free trial
agreement.
const usageBasedPricingTerm = { id: USAGE_BASED_PRICING_TERM_ID };
const fixedUpfrontPricingTerm = { id: FIXED_UPFRONT_PRICING_TERM_ID };
const paymentScheduleTerm = { id: PAYMENT_SCHEDULE_TERM_ID };
const validityTerm = { id: VALIDITY_TERM_ID };
const newLegalTerm = { id: NEW_LEGAL_TERM_ID };

const carResponse = await client.send(
  new CreateAgreementRequestCommand({
```

```

        clientToken: generateClientToken(),
        intent: "REPLACE",
        requestedTerms: [usageBasedPricingTerm, fixedUpfrontPricingTerm,
paymentScheduleTerm, validityTerm, newLegalTerm],
        agreementProposalIdentifier: NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
        sourceAgreementIdentifier:
acceptAgreementRequestResponse.agreementId,
    })
);
console.log("Replace agreement request created. AgreementRequestId: " +
carResponse.agreementRequestId);

const aarResponse = await client.send(
    new AcceptAgreementRequestCommand({
        agreementRequestId: carResponse.agreementRequestId,
    })
);
console.log("Agreement with freeTrialPricingTerm replaced with paid CCP
offer. New AgreementId: " + aarResponse.agreementId);
}

createSaaSFreeTrialAndReplaceWithCCP();

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
"""
```

```
Demonstrates how to create a SaaS free trial agreement and then replace it with a
paid Contract with Consumption Pricing (CCP) offer using the AWS Marketplace
Agreement Service APIs.
```

Scenario: A buyer first starts a free trial on a SaaS product. Once the trial is active, they decide to convert by replacing it with a paid offer that includes a `FixedUpfrontPricingTerm`, `PaymentScheduleTerm`, and `UsageBasedPricingTerm`.

Flow:

1. Create a SaaS free trial agreement with `freeTrialPricingTerm`.
2. Wait for `freeTrialPricingTerm` agreement entitlements to become active.
3. Replace the free trial agreement with the paid CCP offer using `Intent.REPLACE`.

Before running this sample, replace the placeholder constants below with values from

your AWS Marketplace offers:

- `AGREEMENT_PROPOSAL_IDENTIFIER` – the `agreementProposalId` from the free trial offer.
- `NEW_AGREEMENT_PROPOSAL_IDENTIFIER` – the `agreementProposalId` from the paid CCP offer.
- Term IDs (starting with `term-`) – found in each offer's term list.

"""

```
import boto3
```

```
from utils.agreement_api_utils import (
    format_output,
    generate_client_token,
    poll_until_entitlements_available,
)
```

```
class ReplaceSaaSFreeTrialWithCCP:
```

```
    # The agreementProposalId from the free trial offer.
    AGREEMENT_PROPOSAL_IDENTIFIER = "<your-free-trial-agreement-proposal-identifier>"
```

```
    # Term ID for the FreeTrialPricingTerm in the free trial offer.
    FREE_TRIAL_PRICING_TERM_ID = "<your-free-trial-pricing-term-id>"
```

```
    # Term ID for the LegalTerm in the free trial offer.
    LEGAL_TERM_ID = "<your-legal-term-id>"
```

```
    # The agreementProposalId from the paid CCP offer to convert to.
```

```

NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-ccp-agreement-proposal-
identifier>"

# Term ID for the FixedUpfrontPricingTerm in the CCP offer.
FIXED_UPFRONT_PRICING_TERM_ID = "<your-fixed-upfront-pricing-term-id>"

# Term ID for the PaymentScheduleTerm in the CCP offer.
PAYMENT_SCHEDULE_TERM_ID = "<your-payment-schedule-term-id>"

# Term ID for the UsageBasedPricingTerm in the CCP offer.
USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-pricing-term-id>"

# Term ID for the ValidityTerm in the CCP offer.
VALIDITY_TERM_ID = "<your-validity-term-id>"

# Term ID for the LegalTerm in the CCP offer.
NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>"

@staticmethod
def create_saas_free_trial_and_replace_with_ccp():
    """
    Full end-to-end flow:
    1. Create a SaaS free trial agreement with freeTrialPricingTerm.
    2. Wait for freeTrialPricingTerm agreement entitlements to become active.
    3. Replace the free trial agreement with the paid CCP offer.
    """
    client = boto3.client("marketplace-agreement")
    cls = ReplaceSaaSFreeTrialWithCCP

    legal_term = {"id": cls.LEGAL_TERM_ID}
    free_trial_pricing_term = {"id": cls.FREE_TRIAL_PRICING_TERM_ID}

    # --- Step 1: Agreement with freeTrialPricingTerm ---
    create_response = client.create_agreement_request(
        clientToken=generate_client_token(),
        intent="NEW",
        requestedTerms=[free_trial_pricing_term, legal_term],
        agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
    )
    agreement_request_id = create_response["agreementRequestId"]
    print("Agreement request with freeTrialPricingTerm created.
    AgreementRequestId: " + agreement_request_id)

    accept_response = client.accept_agreement_request(

```

```

        agreementRequestId=agreement_request_id
    )
    agreement_id = accept_response["agreementId"]
    print("Agreement request with freeTrialPricingTerm accepted. AgreementId:
" + agreement_id)

    # Wait for freeTrialPricingTerm agreement entitlements to become active
    before replacing.
    print("Waiting for freeTrialPricingTerm agreement entitlements to become
    active...")
    entitlements_response = poll_until_entitlements_available(client,
    agreement_id)
    print("freeTrialPricingTerm agreement entitlements are now active.")
    format_output(entitlements_response)

    # --- Step 2: Replace Agreement with freeTrialPricingTerm with Paid CCP
    Offer ---
    # Use Intent.REPLACE and sourceAgreementIdentifier to replace the free
    trial agreement.
    usage_based_pricing_term = {"id": cls.USAGE_BASED_PRICING_TERM_ID}
    fixed_upfront_pricing_term = {"id": cls.FIXED_UPFRONT_PRICING_TERM_ID}
    payment_schedule_term = {"id": cls.PAYMENT_SCHEDULE_TERM_ID}
    validity_term = {"id": cls.VALIDITY_TERM_ID}
    new_legal_term = {"id": cls.NEW_LEGAL_TERM_ID}

    car_response = client.create_agreement_request(
        clientToken=generate_client_token(),
        intent="REPLACE",
        requestedTerms=[
            usage_based_pricing_term, fixed_upfront_pricing_term,
            payment_schedule_term, validity_term, new_legal_term,
        ],
        agreementProposalIdentifier=cls.NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
        sourceAgreementIdentifier=agreement_id,
    )
    print("Replace agreement request created. AgreementRequestId: " +
    car_response["agreementRequestId"])

    aar_response = client.accept_agreement_request(
        agreementRequestId=car_response["agreementRequestId"]
    )
    print(
        "Agreement with freeTrialPricingTerm replaced with paid CCP offer.
    New AgreementId: "

```

```

        + aar_response["agreementId"]
    )

if __name__ == "__main__":
    ReplaceSaaSFreeTrialWithCCP.create_saas_free_trial_and_replace_with_ccp()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 SaaS 사용량 기반 요금 기간 대체 및 이전 계약 취소

다음 코드 예제에서는 SaaS 사용량 기반 요금 기간을 대체하고 이전 계약을 취소하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest

```

```

import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse;
import
    software.amazon.awssdk.services.marketplaceagreement.model.CancelAgreementRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementEntitlementsResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to create a SaaS agreement with usageBasedPricingTerm (UBPT)
 * and then replace it
 * with a new offer using the AWS Marketplace Agreement Service APIs.
 *
 * <p>Scenario: A buyer subscribes to a SaaS product with UsageBasedPricingTerm.
 * The buyer then converts to a different offer by replacing the existing
 * agreement.
 *
 * <p>Flow:
 * <ol>
 * <li>Create and accept the initial agreement request with UBPT.</li>
 * <li>Wait for entitlements to become active.</li>
 * <li>Replace the agreement with a new offer using Intent.REPLACE.</li>
 * <li>Cancel the new agreement using CancelAgreement.</li>
 * </ol>
 *
 * <p>Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offers:
 * <ul>
 * <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
 * initial offer.</li>
 * <li>{@code NEW_AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from
 * the new offer to replace to.</li>
 * <li>Term IDs (starting with {@code term-}) – found in each offer's term
 * list.</li>
 * </ul>
 */
public class ReplaceSaaSUsageBasedPricingTermAndCancel {

    // The agreementProposalId from the initial offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-
proposal-identifier>";

```

```

// Term ID for the UsageBasedPricingTerm in the initial offer.
private static final String USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-
pricing-term-id>";

// Term ID for the ValidityTerm in the initial offer.
private static final String VALIDITY_TERM_ID = "<your-validity-term-id>";

// Term ID for the LegalTerm in the initial offer.
private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

// The agreementProposalId from the new offer to replace to.
private static final String NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-
agreement-proposal-identifier>";

// Term ID for the UsageBasedPricingTerm in the new offer.
private static final String NEW_USAGE_BASED_PRICING_TERM_ID = "<your-new-
usage-based-pricing-term-id>";

// Term ID for the ValidityTerm in the new offer.
private static final String NEW_VALIDITY_TERM_ID = "<your-new-validity-term-
id>";

// Term ID for the LegalTerm in the new offer.
private static final String NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>";

public static void main(String[] args) {
    replaceSaaSbptAndCancel();
}

/**
 * Full end-to-end flow:
 * 1. Create and accept the initial agreement request with
UsageBasedPricingTerm.
 * 2. Wait for entitlements to become active.
 * 3. Replace the agreement with a new offer.
 * 4. Cancel the new agreement.
 */
private static void replaceSaaSbptAndCancel() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

```



```

        RequestedTerm usageBasedPricingTerm =
RequestedTerm.builder().id(USAGE_BASED_PRICING_TERM_ID).build();
        RequestedTerm validityTerm =
RequestedTerm.builder().id(VALIDITY_TERM_ID).build();
        RequestedTerm legalTerm =
RequestedTerm.builder().id(LEGAL_TERM_ID).build();

// --- Step 1: Create and accept the initial UBPT agreement request ---
CreateAgreementRequestRequest createAgreementRequestRequest =
    CreateAgreementRequestRequest.builder()
        .clientToken(AgreementApiUtils.generateClientToken())
        .intent(Intent.NEW)
        .requestedTerms(usageBasedPricingTerm, validityTerm,
legalTerm)

        .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
            .build();
    CreateAgreementRequestResponse createAgreementRequestResponse =

marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
    System.out.println("Agreement request with UBPT created.
AgreementRequestId: " + createAgreementRequestResponse.agreementRequestId());

    AcceptAgreementRequestRequest acceptAgreementRequestRequest =
        AcceptAgreementRequestRequest.builder()

            .agreementRequestId(createAgreementRequestResponse.agreementRequestId())
                .build();
    AcceptAgreementRequestResponse acceptAgreementRequestResponse =

marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
    System.out.println("Agreement request with UBPT accepted. AgreementId: "
+ acceptAgreementRequestResponse.agreementId());

    // Wait for entitlements to become active before replacing.
    System.out.println("Waiting for entitlements to become active...");
    GetAgreementEntitlementsResponse entitlementsResponse =
AgreementApiUtils.pollUntilEntitlementsAvailable(
        marketplaceAgreementClient,
acceptAgreementRequestResponse.agreementId());
    System.out.println("Entitlements are now active.");
    AgreementApiUtils.formatOutput(entitlementsResponse);

```

```

// --- Step 2: Replace the UBPT agreement with a new offer ---
// Use Intent.REPLACE and sourceAgreementIdentifier to replace the
existing agreement.
    RequestedTerm newUsageBasedPricingTerm =
RequestedTerm.builder().id(NEW_USAGE_BASED_PRICING_TERM_ID).build();
    RequestedTerm newValidityTerm =
RequestedTerm.builder().id(NEW_VALIDITY_TERM_ID).build();
    RequestedTerm newLegalTerm =
RequestedTerm.builder().id(NEW_LEGAL_TERM_ID).build();

    CreateAgreementRequestRequest carRequest =
        CreateAgreementRequestRequest.builder()
            .clientToken(AgreementApiUtils.generateClientToken())
            .intent(Intent.REPLACE)
            .requestedTerms(newUsageBasedPricingTerm,
newValidityTerm, newLegalTerm)

.agreementProposalIdentifier(NEW_AGREEMENT_PROPOSAL_IDENTIFIER)

.sourceAgreementIdentifier(acceptAgreementRequestResponse.agreementId())
        .build();
    CreateAgreementRequestResponse carResponse =
        marketplaceAgreementClient.createAgreementRequest(carRequest);
    System.out.println("Replace agreement request created.
AgreementRequestId: " + carResponse.agreementRequestId());

    AcceptAgreementRequestRequest aarRequest =
        AcceptAgreementRequestRequest.builder()
            .agreementRequestId(carResponse.agreementRequestId())
            .build();
    AcceptAgreementRequestResponse aarResponse =
        marketplaceAgreementClient.acceptAgreementRequest(aarRequest);
    final String agreementIdWithTheNewOffer = aarResponse.agreementId();
    System.out.println("UBPT agreement replaced. New AgreementId: " +
agreementIdWithTheNewOffer);

// --- Step 3: Cancel the new agreement ---
    CancelAgreementRequest cancelAgreementRequest =
        CancelAgreementRequest.builder()
            .agreementId(agreementIdWithTheNewOffer)
            .build();
    marketplaceAgreementClient.cancelAgreement(cancelAgreementRequest);
    System.out.println("The new agreement has been cancelled. AgreementId: "
+ agreementIdWithTheNewOffer);

```

```
    }
}
```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
  MarketplaceAgreementClient,
  CreateAgreementRequestCommand,
  AcceptAgreementRequestCommand,
  CancelAgreementCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken, formatOutput, pollUntilEntitlementsAvailable } =
  require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to create a SaaS agreement with usageBasedPricingTerm (UBPT)
 * and then replace it
 * with a new offer using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: A buyer subscribes to a SaaS product with UsageBasedPricingTerm.
 * The buyer then converts to a different offer by replacing the existing
 * agreement.
 *
 * Flow:
 * 1. Create and accept the initial agreement request with UBPT.
 * 2. Wait for entitlements to become active.
 * 3. Replace the agreement with a new offer using Intent.REPLACE.
 * 4. Cancel the new agreement using CancelAgreement.
 *
 * Before running this sample, replace the placeholder constants below with
 * values from
```

```

* your AWS Marketplace offers:
*   - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the initial
  offer.
*   - NEW_AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the new
  offer to replace to.
*   - Term IDs (starting with "term-") – found in each offer's term list.
*/

// The agreementProposalId from the initial offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>";

// Term ID for the UsageBasedPricingTerm in the initial offer.
const USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-pricing-term-id>";

// Term ID for the ValidityTerm in the initial offer.
const VALIDITY_TERM_ID = "<your-validity-term-id>";

// Term ID for the LegalTerm in the initial offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// The agreementProposalId from the new offer to replace to.
const NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-agreement-proposal-
identifier>";

// Term ID for the UsageBasedPricingTerm in the new offer.
const NEW_USAGE_BASED_PRICING_TERM_ID = "<your-new-usage-based-pricing-term-id>";

// Term ID for the ValidityTerm in the new offer.
const NEW_VALIDITY_TERM_ID = "<your-new-validity-term-id>";

// Term ID for the LegalTerm in the new offer.
const NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>";

/**
 * Full end-to-end flow:
 * 1. Create and accept the initial agreement request with UsageBasedPricingTerm.
 * 2. Wait for entitlements to become active.
 * 3. Replace the agreement with a new offer.
 * 4. Cancel the new agreement.
 */
async function replaceSaaSUBptAndCancel() {
    const client = new MarketplaceAgreementClient();

    const usageBasedPricingTerm = { id: USAGE_BASED_PRICING_TERM_ID };

```

```
const validityTerm = { id: VALIDITY_TERM_ID };
const legalTerm = { id: LEGAL_TERM_ID };

// --- Step 1: Create and accept the initial UBPT agreement request ---
const createAgreementRequestResponse = await client.send(
  new CreateAgreementRequestCommand({
    clientToken: generateClientToken(),
    intent: "NEW",
    requestedTerms: [usageBasedPricingTerm, validityTerm, legalTerm],
    agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
  })
);
console.log("Agreement request with UBPT created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId);

const acceptAgreementRequestResponse = await client.send(
  new AcceptAgreementRequestCommand({
    agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
  })
);
console.log("Agreement request with UBPT accepted. AgreementId: " +
acceptAgreementRequestResponse.agreementId);

// Wait for entitlements to become active before replacing.
console.log("Waiting for entitlements to become active...");
const entitlementsResponse = await pollUntilEntitlementsAvailable(client,
acceptAgreementRequestResponse.agreementId);
console.log("Entitlements are now active.");
formatOutput(entitlementsResponse);

// --- Step 2: Replace the UBPT agreement with a new offer ---
// Use Intent.REPLACE and sourceAgreementIdentifier to replace the existing
agreement.
const newUsageBasedPricingTerm = { id: NEW_USAGE_BASED_PRICING_TERM_ID };
const newValidityTerm = { id: NEW_VALIDITY_TERM_ID };
const newLegalTerm = { id: NEW_LEGAL_TERM_ID };

const carResponse = await client.send(
  new CreateAgreementRequestCommand({
    clientToken: generateClientToken(),
    intent: "REPLACE",
    requestedTerms: [newUsageBasedPricingTerm, newValidityTerm,
newLegalTerm],
```

```

        agreementProposalIdentifier: NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
        sourceAgreementIdentifier:
    acceptAgreementRequestResponse.agreementId,
    })
    );
    console.log("Replace agreement request created. AgreementRequestId: " +
    carResponse.agreementRequestId);

    const aarResponse = await client.send(
        new AcceptAgreementRequestCommand({
            agreementRequestId: carResponse.agreementRequestId,
        })
    );
    const agreementIdWithTheNewOffer = aarResponse.agreementId;
    console.log("UBPT agreement replaced. New AgreementId: " +
    agreementIdWithTheNewOffer);

    // --- Step 3: Cancel the new agreement ---
    await client.send(
        new CancelAgreementCommand({
            agreementId: agreementIdWithTheNewOffer,
        })
    );
    console.log("The new agreement has been cancelled. AgreementId: " +
    agreementIdWithTheNewOffer);
}

replaceSaaSUBPTAndCancel();

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

"""

Demonstrates how to create a SaaS agreement with usageBasedPricingTerm (UBPT) and then replace it with a new offer using the AWS Marketplace Agreement Service APIs.

Scenario: A buyer subscribes to a SaaS product with UsageBasedPricingTerm. The buyer then converts to a different offer by replacing the existing agreement.

Flow:

1. Create and accept the initial agreement request with UBPT.
2. Wait for entitlements to become active.
3. Replace the agreement with a new offer using Intent.REPLACE.
4. Cancel the new agreement using CancelAgreement.

Before running this sample, replace the placeholder constants below with values from

your AWS Marketplace offers:

- AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the initial offer.
- NEW_AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the new offer to replace to.
- Term IDs (starting with term-) – found in each offer's term list.

"""

```
import boto3
```

```
from utils.agreement_api_utils import (
    format_output,
    generate_client_token,
    poll_until_entitlements_available,
)
```

```
class ReplaceSaaSUsageBasedPricingTermAndCancel:
```

```
    # The agreementProposalId from the initial offer.
    AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>"

    # Term ID for the UsageBasedPricingTerm in the initial offer.
    USAGE_BASED_PRICING_TERM_ID = "<your-usage-based-pricing-term-id>"

    # Term ID for the ValidityTerm in the initial offer.
    VALIDITY_TERM_ID = "<your-validity-term-id>"
```

```

# Term ID for the LegalTerm in the initial offer.
LEGAL_TERM_ID = "<your-legal-term-id>"

# The agreementProposalId from the new offer to replace to.
NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-agreement-proposal-
identifier>"

# Term ID for the UsageBasedPricingTerm in the new offer.
NEW_USAGE_BASED_PRICING_TERM_ID = "<your-new-usage-based-pricing-term-id>"

# Term ID for the ValidityTerm in the new offer.
NEW_VALIDITY_TERM_ID = "<your-new-validity-term-id>"

# Term ID for the LegalTerm in the new offer.
NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>"

@staticmethod
def replace_saas_ubpt_and_cancel():
    """
    Full end-to-end flow:
    1. Create and accept the initial agreement request with
UsageBasedPricingTerm.
    2. Wait for entitlements to become active.
    3. Replace the agreement with a new offer.
    4. Cancel the new agreement.
    """
    client = boto3.client("marketplace-agreement")
    cls = ReplaceSaaSUsageBasedPricingTermAndCancel

    usage_based_pricing_term = {"id": cls.USAGE_BASED_PRICING_TERM_ID}
    validity_term = {"id": cls.VALIDITY_TERM_ID}
    legal_term = {"id": cls.LEGAL_TERM_ID}

    # --- Step 1: Create and accept the initial UBPT agreement request ---
    create_response = client.create_agreement_request(
        clientToken=generate_client_token(),
        intent="NEW",
        requestedTerms=[usage_based_pricing_term, validity_term, legal_term],
        agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
    )
    agreement_request_id = create_response["agreementRequestId"]
    print("Agreement request with UBPT created. AgreementRequestId: " +
agreement_request_id)

```



```

    accept_response = client.accept_agreement_request(
        agreementRequestId=agreement_request_id
    )
    agreement_id = accept_response["agreementId"]
    print("Agreement request with UBPT accepted. AgreementId: " +
agreement_id)

    # Wait for entitlements to become active before replacing.
    print("Waiting for entitlements to become active...")
    entitlements_response = poll_until_entitlements_available(client,
agreement_id)
    print("Entitlements are now active.")
    format_output(entitlements_response)

    # --- Step 2: Replace the UBPT agreement with a new offer ---
    # Use Intent.REPLACE and sourceAgreementIdentifier to replace the
existing agreement.
    new_usage_based_pricing_term = {"id":
cls.NEW_USAGE_BASED_PRICING_TERM_ID}
    new_validity_term = {"id": cls.NEW_VALIDITY_TERM_ID}
    new_legal_term = {"id": cls.NEW_LEGAL_TERM_ID}

    car_response = client.create_agreement_request(
        clientToken=generate_client_token(),
        intent="REPLACE",
        requestedTerms=[new_usage_based_pricing_term, new_validity_term,
new_legal_term],
        agreementProposalIdentifier=cls.NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
        sourceAgreementIdentifier=agreement_id,
    )
    print("Replace agreement request created. AgreementRequestId: " +
car_response["agreementRequestId"])

    aar_response = client.accept_agreement_request(
        agreementRequestId=car_response["agreementRequestId"]
    )
    agreement_id_with_new_offer = aar_response["agreementId"]
    print("UBPT agreement replaced. New AgreementId: " +
agreement_id_with_new_offer)

    # --- Step 3: Cancel the new agreement ---
    client.cancel_agreement(agreementId=agreement_id_with_new_offer)

```

```

        print("The new agreement has been cancelled. AgreementId: " +
              agreement_id_with_new_offer)

if __name__ == "__main__":
    ReplaceSaaSUsageBasedPricingTermAndCancel.replace_saas_ubpt_and_cancel()

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 AMI 사용량 기반 요금 기간을 바꾸고 기존 ConfigurableUpfrontPricingTerm 유지

다음 코드 예제에서는 기존 ConfigurableUpfrontPricingTerm을 유지하면서 AMI 사용량 기반 요금 조건을 대체하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse

```

```

import
    software.amazon.awssdk.services.marketplaceagreement.model.ConfigurableUpfrontPricingTerm
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse
import software.amazon.awssdk.services.marketplaceagreement.model.Dimension;
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementEntitlementsResponse
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RequestedTermConfiguration;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to replace an AMI agreement with usageBasedPricingTerm with a
 * new offer while
 * the agreement with ConfigurableUpfrontPricingTerm (CUPT) is still active,
 * using the AWS Marketplace Agreement Service APIs.
 *
 * <p>Scenario: An AMI product with USAGE pricing requires two agreements:
 * <ol>
 * <li>An agreement with <b>usageBasedPricingTerm</b> – accepted first to
 * establish the base agreement.</li>
 * <li>An agreement with <b>configurableUpfrontPricingTerm (CUPT)</b> –
 * accepted after the usageBasedPricingTerm agreement entitlements are active.</li>
 * </ol>
 * <p> This sample shows how to replace only the agreement with
 * usageBasedPricingTerm with a
 * new offer without touching the existing agreement with
 * configurableUpfrontPricingTerm (CUPT).
 *
 * <p>Flow:
 * <ol>
 * <li>Create and accept the initial agreement request with
 * usageBasedPricingTerm.</li>
 * <li>Create and accept the agreement request with
 * configurableUpfrontPricingTerm (CUPT) (after usageBasedPricingTerm agreement
 * entitlements are active).</li>
 * <li>Replace the agreement with usageBasedPricingTerm with a new offer using
 * Intent.REPLACE.</li>
 * </ol>
 *

```

```

* <p>Before running this sample, replace the placeholder constants below with
values from
* your AWS Marketplace offers:
* <ul>
*   <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
initial offer.</li>
*   <li>{@code NEW_AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from
the new offer to replace to.</li>
*   <li>Term IDs (starting with {@code term-}) – found in each offer's term
list.</li>
*   <li>{@code SELECTOR_VALUE} – duration for the agreement (e.g., {@code
P365D}).</li>
*   <li>{@code DIMENSION_1_KEY} – dimension key defined in the CUPT term.</li>
* </ul>
*/
public class ReplaceAmiUsageBasedPricingTermButNotCupt {

    // The agreementProposalId from the initial offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-
proposal-identifier>";

    // Term ID for the ConfigurableUpfrontPricingTerm in the initial offer.
    private static final String CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-
configurable-upfront-pricing-term-id>";

    // Duration for the agreement (e.g., "P365D" for 365 days).
    private static final String SELECTOR_VALUE = "<your-selector-value>";

    // Dimension key defined in the CUPT term.
    private static final String DIMENSION_1_KEY = "<your-dimension-key>";

    // Quantity for the dimension.
    private static final int DIMENSION_1_VALUE = 1;

    // Term ID for the UsageBasedPricingTerm in the initial offer.
    private static final String USAGE_TERM_ID = "<your-usage-term-id>";

    // Term ID for the LegalTerm in the initial offer.
    private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

    // Term ID for the ValidityTerm in the initial offer.
    private static final String VALIDITY_TERM_ID = "<your-validity-term-id>";

    // The agreementProposalId from the new offer to replace to.

```

```

private static final String NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-
agreement-proposal-identifier>";

// Term ID for the UsageBasedPricingTerm in the new offer.
private static final String NEW_USAGE_TERM_ID = "<your-new-usage-term-id>";

// Term ID for the LegalTerm in the new offer.
private static final String NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>";

// Term ID for the ValidityTerm in the new offer.
private static final String NEW_VALIDITY_TERM_ID = "<your-new-validity-term-
id>";

public static void main(String[] args) {
    replaceAmiUsageWhenCuptExists();
}

/**
 * Full end-to-end flow:
 * 1. Create and accept an agreement request with usageBasedPricingTerm, then
wait for entitlements.
 * 2. Create and accept an agreement request with CUPT, then wait for
entitlements.
 * 3. Replace the agreement with usageBasedPricingTerm with a new offer
(agreement with CUPT is unaffected).
 */
private static void replaceAmiUsageWhenCuptExists() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    RequestedTerm usageTerm =
        RequestedTerm.builder().id(USAGE_TERM_ID).build();
    RequestedTerm legalTerm =
        RequestedTerm.builder().id(LLEGAL_TERM_ID).build();
    RequestedTerm validityTerm =
        RequestedTerm.builder().id(VALIDITY_TERM_ID).build();

    // --- Step 1: Agreement with UBPT ---
    CreateAgreementRequestRequest createAgreementRequestRequest =
        CreateAgreementRequestRequest.builder()
            .clientToken(AgreementApiUtils.generateClientToken())

```

```

        .intent(Intent.NEW)
        .requestedTerms(usageTerm, legalTerm, validityTerm)

        .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
        .build();
        CreateAgreementRequestResponse createAgreementRequestResponse =
marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
        System.out.println("Agreement request with UBPT created.
AgreementRequestId: " + createAgreementRequestResponse.agreementRequestId());

        AcceptAgreementRequestRequest acceptAgreementRequestRequest =
            AcceptAgreementRequestRequest.builder()

        .agreementRequestId(createAgreementRequestResponse.agreementRequestId())
        .build();
        AcceptAgreementRequestResponse acceptAgreementRequestResponse =
marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
        final String usageAgreementId =
acceptAgreementRequestResponse.agreementId();
        System.out.println("Agreement request with UBPT accepted. AgreementId: "
+ usageAgreementId);

        // Wait for UBPT agreement entitlements to become active before creating
the agreement with CUPT.
        System.out.println("Waiting for UBPT agreement entitlements to become
active...");
        GetAgreementEntitlementsResponse entitlementsResponse =
AgreementApiUtils.pollUntilEntitlementsAvailable(
            marketplaceAgreementClient, usageAgreementId);
        System.out.println("UBPT agreement entitlements are now active.");
        AgreementApiUtils.formatOutput(entitlementsResponse);

        // --- Step 2: Agreement with configurableUpfrontPricingTerm (CUPT) ---
        RequestedTerm configurableUpfrontPricingTerm = RequestedTerm.builder()
            .id(CONFIGURABLE_UPFRONT_PRICING_TERM_ID)

        .configuration(RequestedTermConfiguration.fromConfigurableUpfrontPricingTermConfiguration(
            ConfigurableUpfrontPricingTermConfiguration.builder()
                .selectorValue(SELECTOR_VALUE)
                .dimensions(Dimension.builder()

        .dimensionKey(DIMENSION_1_KEY)

```

```

        .dimensionValue(DIMENSION_1_VALUE)
                                .build())
        .build()))
    .build();

    CreateAgreementRequestRequest carRequest =
        CreateAgreementRequestRequest.builder()
            .clientToken(AgreementApiUtils.generateClientToken())
            .intent(Intent.NEW)
            .requestedTerms(configurableUpfrontPricingTerm,
legalTerm, validityTerm)

        .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
            .build();
    CreateAgreementRequestResponse carResponse =
        marketplaceAgreementClient.createAgreementRequest(carRequest);
    System.out.println("Agreement request with CUPT created.
AgreementRequestId: " + carResponse.agreementRequestId());

    AcceptAgreementRequestRequest aarRequest =
        AcceptAgreementRequestRequest.builder()
            .agreementRequestId(carResponse.agreementRequestId())
            .build();
    AcceptAgreementRequestResponse aarResponse =
        marketplaceAgreementClient.acceptAgreementRequest(aarRequest);
    final String cuptAgreementId = aarResponse.agreementId();
    System.out.println("Agreement request with CUPT accepted. AgreementId: "
+ cuptAgreementId);

    // Wait for CUPT agreement entitlements to become active before replacing
usage.
    System.out.println("Waiting for CUPT agreement entitlements to become
active...");
    GetAgreementEntitlementsResponse cuptEntitlementsResponse =
AgreementApiUtils.pollUntilEntitlementsAvailable(
        marketplaceAgreementClient, cuptAgreementId);
    System.out.println("CUPT agreement entitlements are now active.");
    AgreementApiUtils.formatOutput(cuptEntitlementsResponse);

    // --- Step 3: Replace Agreement with usageBasedPricingTerm with a new
offer ---
    RequestedTerm newUsageTerm =
RequestedTerm.builder().id(NEW_USAGE_TERM_ID).build();

```

```

        RequestedTerm newLegalTerm =
RequestedTerm.builder().id(NEW_LEGAL_TERM_ID).build();
        RequestedTerm newValidityTerm =
RequestedTerm.builder().id(NEW_VALIDITY_TERM_ID).build();

        // Use Intent.REPLACE and sourceAgreementIdentifier pointing to the
usageBasedPricingTerm agreement only.
        // The agreement with configurableUpfrontPricingTerm (CUPT) is NOT
affected by this replacement.
        CreateAgreementRequestRequest carReplaceRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.REPLACE)
                .requestedTerms(newUsageTerm, newLegalTerm,
newValidityTerm)

            .agreementProposalIdentifier(NEW_AGREEMENT_PROPOSAL_IDENTIFIER)
                .sourceAgreementIdentifier(usageAgreementId)
                .build();
        CreateAgreementRequestResponse carReplaceResponse =
marketplaceAgreementClient.createAgreementRequest(carReplaceRequest);
        System.out.println("Replace agreement request created.
AgreementRequestId: " + carReplaceResponse.agreementRequestId());

        AcceptAgreementRequestRequest aarReplaceRequest =
            AcceptAgreementRequestRequest.builder()

            .agreementRequestId(carReplaceResponse.agreementRequestId())
                .build();
        AcceptAgreementRequestResponse aarReplaceResponse =
marketplaceAgreementClient.acceptAgreementRequest(aarReplaceRequest);
        System.out.println("Agreement with UBPT replaced. New AgreementId: " +
aarReplaceResponse.agreementId());
    }
}

```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
    MarketplaceAgreementClient,
    CreateAgreementRequestCommand,
    AcceptAgreementRequestCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken, formatOutput, pollUntilEntitlementsAvailable } =
    require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to replace an AMI agreement with usageBasedPricingTerm with a
 * new offer while
 * the agreement with ConfigurableUpfrontPricingTerm (CUPT) is still active,
 * using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: An AMI product with USAGE pricing requires two agreements:
 * 1. An agreement with usageBasedPricingTerm – accepted first to establish the
 * base agreement.
 * 2. An agreement with configurableUpfrontPricingTerm (CUPT) – accepted after
 * the usageBasedPricingTerm agreement entitlements are active.
 *
 * This sample shows how to replace only the agreement with usageBasedPricingTerm
 * with a
 * new offer without touching the existing agreement with
 * configurableUpfrontPricingTerm (CUPT).
 *
 * Flow:
 * 1. Create and accept the initial agreement request with
 * usageBasedPricingTerm.
 * 2. Create and accept the agreement request with
 * configurableUpfrontPricingTerm (CUPT) (after usageBasedPricingTerm agreement
 * entitlements are active).
```

```

*   3. Replace the agreement with usageBasedPricingTerm with a new offer using
Intent.REPLACE.
*
* Before running this sample, replace the placeholder constants below with
values from
* your AWS Marketplace offers:
*   - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the initial
offer.
*   - NEW_AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the new
offer to replace to.
*   - Term IDs (starting with "term-") – found in each offer's term list.
*   - SELECTOR_VALUE – duration for the agreement (e.g., "P365D").
*   - DIMENSION_1_KEY – dimension key defined in the CUPT term.
*/

// The agreementProposalId from the initial offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>";

// Term ID for the ConfigurableUpfrontPricingTerm in the initial offer.
const CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-
term-id>";

// Duration for the agreement (e.g., "P365D" for 365 days).
const SELECTOR_VALUE = "<your-selector-value>";

// Dimension key defined in the CUPT term.
const DIMENSION_1_KEY = "<your-dimension-key>";

// Quantity for the dimension.
const DIMENSION_1_VALUE = 1;

// Term ID for the UsageBasedPricingTerm in the initial offer.
const USAGE_TERM_ID = "<your-usage-term-id>";

// Term ID for the LegalTerm in the initial offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// Term ID for the ValidityTerm in the initial offer.
const VALIDITY_TERM_ID = "<your-validity-term-id>";

// The agreementProposalId from the new offer to replace to.
const NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-agreement-proposal-
identifier>";

```

```
// Term ID for the UsageBasedPricingTerm in the new offer.
const NEW_USAGE_TERM_ID = "<your-new-usage-term-id>";

// Term ID for the LegalTerm in the new offer.
const NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>";

// Term ID for the ValidityTerm in the new offer.
const NEW_VALIDITY_TERM_ID = "<your-new-validity-term-id>";

/**
 * Full end-to-end flow:
 * 1. Create and accept an agreement request with usageBasedPricingTerm, then
    wait for entitlements.
 * 2. Create and accept an agreement request with CUPT, then wait for
    entitlements.
 * 3. Replace the agreement with usageBasedPricingTerm with a new offer
    (agreement with CUPT is unaffected).
 */
async function replaceAmiUsageWhenCuptExists() {
    const client = new MarketplaceAgreementClient();

    const usageTerm = { id: USAGE_TERM_ID };
    const legalTerm = { id: LEGAL_TERM_ID };
    const validityTerm = { id: VALIDITY_TERM_ID };

    // --- Step 1: Agreement with UBPT ---
    const createAgreementRequestResponse = await client.send(
        new CreateAgreementRequestCommand({
            clientToken: generateClientToken(),
            intent: "NEW",
            requestedTerms: [usageTerm, legalTerm, validityTerm],
            agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
        })
    );
    console.log("Agreement request with UBPT created. AgreementRequestId: " +
        createAgreementRequestResponse.agreementRequestId);

    const acceptAgreementRequestResponse = await client.send(
        new AcceptAgreementRequestCommand({
            agreementRequestId:
                createAgreementRequestResponse.agreementRequestId,
        })
    );
    const usageAgreementId = acceptAgreementRequestResponse.agreementId;
```

```

    console.log("Agreement request with UBPT accepted. AgreementId: " +
usageAgreementId);

    // Wait for UBPT agreement entitlements to become active before creating the
agreement with CUPT.
    console.log("Waiting for UBPT agreement entitlements to become active...");
    const entitlementsResponse = await pollUntilEntitlementsAvailable(client,
usageAgreementId);
    console.log("UBPT agreement entitlements are now active.");
    formatOutput(entitlementsResponse);

    // --- Step 2: Agreement with configurableUpfrontPricingTerm (CUPT) ---
    const configurableUpfrontPricingTerm = {
        id: CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        configuration: {
            configurableUpfrontPricingTermConfiguration: {
                selectorValue: SELECTOR_VALUE,
                dimensions: [
                    { dimensionKey: DIMENSION_1_KEY, dimensionValue:
DIMENSION_1_VALUE },
                ],
            },
        },
    };

    const carResponse = await client.send(
        new CreateAgreementRequestCommand({
            clientToken: generateClientToken(),
            intent: "NEW",
            requestedTerms: [configurableUpfrontPricingTerm, legalTerm,
validityTerm],
            agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
        })
    );

    console.log("Agreement request with CUPT created. AgreementRequestId: " +
carResponse.agreementRequestId);

    const aarResponse = await client.send(
        new AcceptAgreementRequestCommand({
            agreementRequestId: carResponse.agreementRequestId,
        })
    );

    const cuptAgreementId = aarResponse.agreementId;

```

```
    console.log("Agreement request with CUPT accepted. AgreementId: " +
    cuptAgreementId);

    // Wait for CUPT agreement entitlements to become active before replacing
    usage.
    console.log("Waiting for CUPT agreement entitlements to become active...");
    const cuptEntitlementsResponse = await pollUntilEntitlementsAvailable(client,
    cuptAgreementId);
    console.log("CUPT agreement entitlements are now active.");
    formatOutput(cuptEntitlementsResponse);

    // --- Step 3: Replace Agreement with usageBasedPricingTerm with a new offer
    ---
    const newUsageTerm = { id: NEW_USAGE_TERM_ID };
    const newLegalTerm = { id: NEW_LEGAL_TERM_ID };
    const newValidityTerm = { id: NEW_VALIDITY_TERM_ID };

    // Use Intent.REPLACE and sourceAgreementIdentifier pointing to the
    usageBasedPricingTerm agreement only.
    // The agreement with configurableUpfrontPricingTerm (CUPT) is NOT affected
    by this replacement.
    const carReplaceResponse = await client.send(
        new CreateAgreementRequestCommand({
            clientToken: generateClientToken(),
            intent: "REPLACE",
            requestedTerms: [newUsageTerm, newLegalTerm, newValidityTerm],
            agreementProposalIdentifier: NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
            sourceAgreementIdentifier: usageAgreementId,
        })
    );
    console.log("Replace agreement request created. AgreementRequestId: " +
    carReplaceResponse.agreementRequestId);

    const aarReplaceResponse = await client.send(
        new AcceptAgreementRequestCommand({
            agreementRequestId: carReplaceResponse.agreementRequestId,
        })
    );
    console.log("Agreement with UBPT replaced. New AgreementId: " +
    aarReplaceResponse.agreementId);
}

replaceAmiUsageWhenCuptExists();
```

- API 세부 정보는 API 참조의 [CreateAgreementRequest](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
"""
```

Demonstrates how to replace an AMI agreement with `usageBasedPricingTerm` with a new offer while the agreement with `ConfigurableUpfrontPricingTerm` (CUPT) is still active, using the AWS Marketplace Agreement Service APIs.

Scenario: An AMI product with USAGE pricing requires two agreements:

1. An agreement with `usageBasedPricingTerm` – accepted first to establish the base agreement.
2. An agreement with `configurableUpfrontPricingTerm` (CUPT) – accepted after the `usageBasedPricingTerm` agreement entitlements are active.

This sample shows how to replace only the agreement with `usageBasedPricingTerm` with a new offer without touching the existing agreement with `configurableUpfrontPricingTerm` (CUPT).

Flow:

1. Create and accept the initial agreement request with `usageBasedPricingTerm`.
2. Create and accept the agreement request with `configurableUpfrontPricingTerm` (CUPT) (after `usageBasedPricingTerm` agreement entitlements are active).
3. Replace the agreement with `usageBasedPricingTerm` with a new offer using `Intent.REPLACE`.

```

Before running this sample, replace the placeholder constants below with values
from
your AWS Marketplace offers:
    - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the initial
      offer.
    - NEW_AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the new
      offer to replace to.
    - Term IDs (starting with term-) – found in each offer's term list.
    - SELECTOR_VALUE – duration for the agreement (e.g., P365D).
    - DIMENSION_1_KEY – dimension key defined in the CUPT term.
"""

import boto3

from utils.agreement_api_utils import (
    format_output,
    generate_client_token,
    poll_until_entitlements_available,
)

class ReplaceAmiUsageBasedPricingTermButNotCupt:

    # The agreementProposalId from the initial offer.
    AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>"

    # Term ID for the ConfigurableUpfrontPricingTerm in the initial offer.
    CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-
term-id>"

    # Duration for the agreement (e.g., "P365D" for 365 days).
    SELECTOR_VALUE = "<your-selector-value>"

    # Dimension key defined in the CUPT term.
    DIMENSION_1_KEY = "<your-dimension-key>"

    # Quantity for the dimension.
    DIMENSION_1_VALUE = 1

    # Term ID for the UsageBasedPricingTerm in the initial offer.
    USAGE_TERM_ID = "<your-usage-term-id>"

    # Term ID for the LegalTerm in the initial offer.
    LEGAL_TERM_ID = "<your-legal-term-id>"

```

```

# Term ID for the ValidityTerm in the initial offer.
VALIDITY_TERM_ID = "<your-validity-term-id>"

# The agreementProposalId from the new offer to replace to.
NEW_AGREEMENT_PROPOSAL_IDENTIFIER = "<your-new-agreement-proposal-
identifier>"

# Term ID for the UsageBasedPricingTerm in the new offer.
NEW_USAGE_TERM_ID = "<your-new-usage-term-id>"

# Term ID for the LegalTerm in the new offer.
NEW_LEGAL_TERM_ID = "<your-new-legal-term-id>"

# Term ID for the ValidityTerm in the new offer.
NEW_VALIDITY_TERM_ID = "<your-new-validity-term-id>"

@staticmethod
def replace_ami_usage_when_cupt_exists():
    """
    Full end-to-end flow:
    1. Create and accept an agreement request with usageBasedPricingTerm,
    then wait for entitlements.
    2. Create and accept an agreement request with CUPT, then wait for
    entitlements.
    3. Replace the agreement with usageBasedPricingTerm with a new offer
    (agreement with CUPT is unaffected).
    """
    client = boto3.client("marketplace-agreement")
    cls = ReplaceAmiUsageBasedPricingTermButNotCupt

    usage_term = {"id": cls.USAGE_TERM_ID}
    legal_term = {"id": cls.LEGAL_TERM_ID}
    validity_term = {"id": cls.VALIDITY_TERM_ID}

    # --- Step 1: Agreement with UBPT ---
    create_response = client.create_agreement_request(
        clientToken=generate_client_token(),
        intent="NEW",
        requestedTerms=[usage_term, legal_term, validity_term],
        agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
    )
    agreement_request_id = create_response["agreementRequestId"]

```



```

    print("Agreement request with UBPT created. AgreementRequestId: " +
          agreement_request_id)

    accept_response = client.accept_agreement_request(
        agreementRequestId=agreement_request_id
    )
    usage_agreement_id = accept_response["agreementId"]
    print("Agreement request with UBPT accepted. AgreementId: " +
          usage_agreement_id)

    # Wait for UBPT agreement entitlements to become active before creating
    the agreement with CUPT.
    print("Waiting for UBPT agreement entitlements to become active...")
    entitlements_response = poll_until_entitlements_available(client,
        usage_agreement_id)
    print("UBPT agreement entitlements are now active.")
    format_output(entitlements_response)

    # --- Step 2: Agreement with configurableUpfrontPricingTerm (CUPT) ---
    configurable_upfront_pricing_term = {
        "id": cls.CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        "configuration": {
            "configurableUpfrontPricingTermConfiguration": {
                "selectorValue": cls.SELECTOR_VALUE,
                "dimensions": [
                    {
                        "dimensionKey": cls.DIMENSION_1_KEY,
                        "dimensionValue": cls.DIMENSION_1_VALUE,
                    },
                ],
            },
        },
    },
}

car_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="NEW",
    requestedTerms=[configurable_upfront_pricing_term, legal_term,
        validity_term],
    agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
)
print("Agreement request with CUPT created. AgreementRequestId: " +
      car_response["agreementRequestId"])

```

```

        aar_response = client.accept_agreement_request(
            agreementRequestId=car_response["agreementRequestId"]
        )
        cupt_agreement_id = aar_response["agreementId"]
        print("Agreement request with CUPT accepted. AgreementId: " +
            cupt_agreement_id)

        # Wait for CUPT agreement entitlements to become active before replacing
        usage.
        print("Waiting for CUPT agreement entitlements to become active...")
        cupt_entitlements_response = poll_until_entitlements_available(client,
            cupt_agreement_id)
        print("CUPT agreement entitlements are now active.")
        format_output(cupt_entitlements_response)

        # --- Step 3: Replace Agreement with usageBasedPricingTerm with a new
        offer ---
        new_usage_term = {"id": cls.NEW_USAGE_TERM_ID}
        new_legal_term = {"id": cls.NEW_LEGAL_TERM_ID}
        new_validity_term = {"id": cls.NEW_VALIDITY_TERM_ID}

        # Use Intent.REPLACE and sourceAgreementIdentifier pointing to the
        usageBasedPricingTerm agreement only.
        # The agreement with configurableUpfrontPricingTerm (CUPT) is NOT
        affected by this replacement.
        car_replace_response = client.create_agreement_request(
            clientToken=generate_client_token(),
            intent="REPLACE",
            requestedTerms=[new_usage_term, new_legal_term, new_validity_term],
            agreementProposalIdentifier=cls.NEW_AGREEMENT_PROPOSAL_IDENTIFIER,
            sourceAgreementIdentifier=usage_agreement_id,
        )
        print("Replace agreement request created. AgreementRequestId: " +
            car_replace_response["agreementRequestId"])

        aar_replace_response = client.accept_agreement_request(
            agreementRequestId=car_replace_response["agreementRequestId"]
        )
        print("Agreement with UBPT replaced. New AgreementId: " +
            aar_replace_response["agreementId"])

if __name__ == "__main__":

```

```
ReplaceAmiUsageBasedPricingTermButNotCupt.replace_ami_usage_when_cupt_exists()
```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [CreateAgreementRequest](#)를 참조하세요. AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계정 ID별로 계약 검색

다음 코드 예제에서는 계정 ID로 계약을 검색하는 방법을 보여줍니다.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to get agreement by customer AWS
account ID
AG-02
"""

import argparse
import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))
```

```

import utils.helpers as helper
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-agreement")
logger = logging.getLogger(__name__)

MAX_PAGE_RESULTS = 10

def get_agreements(account_id):
    AgreementSummaryList = []

    try:
        agreement = mp_client.search_agreements(
            catalog="AWSMarketplace",
            maxResults=MAX_PAGE_RESULTS,
            # Set PartyType filter to "Proposer" to return agreements where you
            # are the proposer.
            # Change to "Acceptor" to return agreements where you are the
            # acceptor.
            filters=[
                {"name": "PartyType", "values": ["Proposer"]},
                {"name": "AcceptorId", "values": [account_id]},
                {"name": "AgreementType", "values": ["PurchaseAgreement"]},
            ],
        )
    except ClientError as e:
        logger.error("Could not complete search_agreements request.")
        raise e

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    while "nextToken" in agreement and agreement["nextToken"] is not None:
        try:
            agreement = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                nextToken=agreement["nextToken"],
                filters=[
                    {"name": "PartyType", "values": ["Proposer"]},
                    {"name": "AcceptorId", "values": [account_id]},
                    {"name": "AgreementType", "values": ["PurchaseAgreement"]},
                ],
            )

```

```

    )
    except ClientError as e:
        logger.error("Could not complete search_agreements request.")
        raise e

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    return AgreementSummaryList

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "--account_id",
        "-aid",
        help="Provide accepting account ID to search for agreements",
        required=True,
    )
    args = parser.parse_args()

    response = get_agreements(account_id=args.account_id)

    helper.pretty_print_datetime(response)

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 ID로 계약 검색

다음 코드 예제에서는 계약 ID로 계약을 검색하는 방법을 보여줍니다.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to search for agreements give id
information
AG-02-A
"""

import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

# To search by offer id: OfferId; by product id: ResourceIdentifier; by product
type: ResourceType
idType = "ResourceType"

# replace id value as needed
idValue = "SaaSProduct"

MAX_PAGE_RESULTS = 10

logger = logging.getLogger(__name__)
```

```

def get_agreements(mp_client):
    AgreementSummaryList = []
    # Set PartyType to "Proposer" to return agreements where you are the
    proposer.
    # Change to "Acceptor" to return agreements where you are the acceptor.
    partyTypes = ["Proposer"]
    for value in partyTypes:
        try:
            agreement = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                filters=[
                    {"name": "PartyType", "values": [value]},
                    {"name": idType, "values": [idValue]},
                    {"name": "AgreementType", "values": ["PurchaseAgreement"]},
                ],
            )
        except ClientError as e:
            logger.error("Could not complete search_agreements request.")
            raise e

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    while "nextToken" in agreement and agreement["nextToken"] is not None:
        try:
            agreement = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                nextToken=agreement["nextToken"],
                filters=[
                    {"name": "PartyType", "values": [value]},
                    {"name": idType, "values": [idValue]},
                    {"name": "AgreementType", "values":
["PurchaseAgreement"]},
                ],
            )
        except ClientError as e:
            logger.error("Could not complete search_agreements request.")
            raise e

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    return AgreementSummaryList

```

```
def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for an agreement in the AWS Marketplace Catalog.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-agreement")

    helper.pretty_print_datetime(get_agreements(mp_client))

if __name__ == "__main__":
    usage_demo()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 종료 날짜별로 계약 검색

다음 코드 예제에서는 종료 날짜로 계약을 검색하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
```



```
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AgreementViewSummary;
import software.amazon.awssdk.services.marketplaceagreement.model.Filter;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsResponse;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

public class SearchAgreementsByEndDate {

    static String beforeOrAfterEndtimeFilterName =
        BeforeOrAfterEndTimeFilterName.BeforeEndTime.name();

    static String cutoffDate = "2050-11-18T00:00:00Z";

    static String partyTypeFilterValue = PARTY_TYPE_FILTER_VALUE_PROPOSER;

    public static void main(String[] args) {

        List<AgreementViewSummary> agreementSummaryList = getAgreements();

        ReferenceCodesUtils.formatOutput(agreementSummaryList);
    }

    public static List<AgreementViewSummary> getAgreements() {
        MarketplaceAgreementClient marketplaceAgreementClient =
            MarketplaceAgreementClient.builder()
                .httpClient(ApacheHttpClient.builder().build())
                .credentialsProvider(ProfileCredentialsProvider.create())
                .build();

        // Set PartyType filter to PARTY_TYPE_FILTER_VALUE_PROPOSER to return
        // agreements where you are the proposer.
        // Change to PARTY_TYPE_FILTER_VALUE_ACCEPTOR to return agreements where you
        // are the acceptor.
```

```
Filter partyTypeFilter = Filter.builder().name(PARTY_TYPE_FILTER_NAME)
    .values(PARTY_TYPE_FILTER_VALUE_PROPOSER).build();

Filter agreementTypeFilter = Filter.builder().name(AGREEMENT_TYPE_FILTER_NAME)
    .values(AGREEMENT_TYPE_FILTER_VALUE_PURCHASEAGREEMENT).build();

Filter customizeFilter =
Filter.builder().name(beforeOrAfterEndtimeFilterName).values(cutoffDate).build();

List<Filter> filters = new ArrayList<Filter>();

filters.addAll(Arrays.asList(partyTypeFilter, agreementTypeFilter,
customizeFilter));

// search agreement with filters

SearchAgreementsRequest searchAgreementsRequest =
    SearchAgreementsRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .filters(filters)
        .build();

SearchAgreementsResponse searchAgreementResponse=
marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

List<AgreementViewSummary> agreementSummaryList = new
ArrayList<AgreementViewSummary>();

agreementSummaryList.addAll(searchAgreementResponse.agreementViewSummaries());

while (searchAgreementResponse.nextToken() != null &&
searchAgreementResponse.nextToken().length() > 0) {
    searchAgreementsRequest =
        SearchAgreementsRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .filters(filters)
            .nextToken(searchAgreementResponse.nextToken())
            .build();
    searchAgreementResponse =
marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);
    agreementSummaryList.addAll(searchAgreementResponse.agreementViewSummaries());
}
return agreementSummaryList;
}
```

```
}
```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [SearchAgreements](#)를 참조하세요.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to search for agreement
information before or after end date
AG-03
"""

import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

mp_client = boto3.client("marketplace-agreement")

# change to 'AfterEndTime' if after endtime is desired
beforeOrAfterEndTimeFilterName = "BeforeEndTime"

# Make sure to use the same date format as below
```

```

cutoffDate = "2322-11-18T00:00:00Z"

MAX_PAGE_RESULTS = 10

logger = logging.getLogger(__name__)

def get_agreements():
    AgreementSummaryList = []

    try:
        agreement = mp_client.search_agreements(
            catalog="AWSMarketplace",
            maxResults=MAX_PAGE_RESULTS,
            # Set PartyType filter to "Proposer" to return agreements where you
            # are the proposer.
            # Change to "Acceptor" to return agreements where you are the
            # acceptor.
            filters=[
                {"name": "PartyType", "values": ["Proposer"]},
                {"name": beforeOrAfterEndtimeFilterName, "values": [cutoffDate]},
                {"name": "AgreementType", "values": ["PurchaseAgreement"]},
            ],
        )
    except ClientError as e:
        logger.error("Could not complete search_agreements request.")
        raise

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    while "nextToken" in agreement:
        try:
            agreement = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                nextToken=agreement["nextToken"],
                filters=[
                    {"name": "PartyType", "values": ["Proposer"]},
                    {
                        "name": beforeOrAfterEndtimeFilterName,
                        "values": [cutoffDate],
                    },
                    {"name": "AgreementType", "values": ["PurchaseAgreement"]},
                ],
            )

```

```

    )
    except ClientError as e:
        logger.error("Could not complete search_agreements request.")
        raise

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    return AgreementSummaryList

if __name__ == "__main__":
    agreements = get_agreements()
    helper.pretty_print_datetime(agreements)

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 제안 ID로 계약 검색

다음 코드 예제에서는 제안 ID로 계약을 검색하는 방법을 보여줍니다.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to search for agreements by offer
id
AG-0

```

```
"""

import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

# offer id to search by
offerId = "11111111111111111111111111111111"

MAX_PAGE_RESULTS = 10

logger = logging.getLogger(__name__)

def get_agreements(mp_client):
    AgreementSummaryList = []
    # Set PartyType to "Proposer" to return agreements where you are the
    proposer.
    # Change to "Acceptor" to return agreements where you are the acceptor.
    partyTypes = ["Proposer"]
    for value in partyTypes:
        try:
            agreement = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                filters=[
                    {"name": "PartyType", "values": [value]},
                    {"name": "OfferId", "values": [offerId]},
                    {"name": "AgreementType", "values": ["PurchaseAgreement"]},
                ],
            )
        except ClientError as e:
            logger.error("Could not complete search_agreements request.")
            raise

        AgreementSummaryList.extend(agreement["agreementViewSummaries"])
```

```

        while "nextToken" in agreement and agreement["nextToken"] is not None:
            try:
                agreement = mp_client.search_agreements(
                    catalog="AWSMarketplace",
                    maxResults=MAX_PAGE_RESULTS,
                    nextToken=agreement["nextToken"],
                    filters=[
                        {"name": "PartyType", "values": [value]},
                        {"name": "OfferId", "values": [offerId]},
                        {"name": "AgreementType", "values":
["PurchaseAgreement"]}],
                    ],
                )
            except ClientError as e:
                logger.error("Could not complete search_agreements request.")
                raise

            AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    return AgreementSummaryList

def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for an agreement by offer id.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-agreement")

    helper.pretty_print_datetime(get_agreements(mp_client))

if __name__ == "__main__":
    usage_demo()

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 [섹션을 참조하세요](#)[AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 제품 ID별로 계약 검색

다음 코드 예제에서는 제품 ID로 계약을 검색하는 방법을 보여줍니다.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to search for agreement by
product id
AG-02
"""

import logging

import boto3
import sys
import os

sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))

import utils.helpers as helper
from botocore.exceptions import ClientError

# product id to search by
resourceId = "prod-11111111111111"

MAX_PAGE_RESULTS = 10

logger = logging.getLogger(__name__)
```



```

def get_agreements(mp_client):
    AgreementSummaryList = []
    # Set PartyType to "Proposer" to return agreements where you are the
    proposer.
    # Change to "Acceptor" to return agreements where you are the acceptor.
    partyTypes = ["Proposer"]
    for value in partyTypes:
        try:
            agreement = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                filters=[
                    {"name": "PartyType", "values": [value]},
                    {"name": "ResourceIdentifier", "values": [resourceId]},
                    {"name": "AgreementType", "values": ["PurchaseAgreement"]},
                ],
            )
        except ClientError as e:
            logger.error("Could not complete list_entities request.")
            raise

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    while "nextToken" in agreement:
        try:
            agreement = mp_client.search_agreements(
                catalog="AWSMarketplace",
                maxResults=MAX_PAGE_RESULTS,
                nextToken=agreement["nextToken"],
                filters=[
                    {"name": "PartyType", "values": [value]},
                    {"name": "ResourceIdentifier", "values": [resourceId]},
                    {"name": "AgreementType", "values":
["PurchaseAgreement"]},
                ],
            )
        except ClientError as e:
            logger.error("Could not complete search_agreements request.")
            raise

    AgreementSummaryList.extend(agreement["agreementViewSummaries"])

    return AgreementSummaryList

```

```
def usage_demo():
    logging.basicConfig(level=logging.INFO, format="%(levelname)s: %(message)s")

    print("-" * 88)
    print("Looking for an agreement in the AWS Marketplace Catalog.")
    print("-" * 88)

    mp_client = boto3.client("marketplace-agreement")

    helper.pretty_print_datetime(get_agreements(mp_client))

if __name__ == "__main__":
    usage_demo()
```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 상태별로 계약 검색

다음 코드 예제에서는 상태별로 계약을 검색하는 방법을 보여줍니다.

Python

SDK for Python(Boto3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Shows how to use the AWS SDK for Python (Boto3) to filter agreements by status
```

AG-04

Example Usage: `python3 search_agreements_by_status.py`

"""

`import logging`

`import boto3`

`import sys`

`import os`

`sys.path.append(os.path.join(os.path.dirname(__file__), "..", "..", ".."))`

`import utils.helpers as helper`

`from botocore.exceptions import ClientError`

`mp_client = boto3.client("marketplace-agreement")`

`logger = logging.getLogger(__name__)`

`MAX_PAGE_RESULTS = 10`

`# Set PartyType to "Proposer" to return agreements where you are the proposer.`

`# Change to "Acceptor" to return agreements where you are the acceptor.`

`party_type_list = ["Proposer"]`

`agreement_type_list = ["PurchaseAgreement"]`

`# Accepted values: "ACTIVE", "TERMINATED", "CANCELED", "EXPIRED", "REPLACED",
"RENEWED"`

`status_list = ["ACTIVE"]`

`filter_list = [`

`{"name": "PartyType", "values": party_type_list},`

`{"name": "AgreementType", "values": agreement_type_list},`

`{"name": "Status", "values": status_list},`

`]`

`agreement_results_list = []`

`def get_agreements(filter_list=filter_list):`

`try:`

`agreements = mp_client.search_agreements(
 catalog="AWSMarketplace",`

```

        maxResults=MAX_PAGE_RESULTS,
        filters=filter_list,
    )
except ClientError as e:
    logger.error("Could not complete search_agreements request.")
    raise e

agreement_results_list.extend(agreements["agreementViewSummaries"])

while "nextToken" in agreements and agreements["nextToken"] is not None:
    try:
        agreements = mp_client.search_agreements(
            catalog="AWSMarketplace",
            maxResults=MAX_PAGE_RESULTS,
            nextToken=agreements["nextToken"],
            filters=filter_list,
        )
    except ClientError as e:
        logger.error("Could not complete search_agreements request.")
        raise e

    agreement_results_list.extend(agreements["agreementViewSummaries"])

helper.pretty_print_datetime(agreement_results_list)
return agreement_results_list

if __name__ == "__main__":
    agreements_list = get_agreements(filter_list)

```

- API 세부 정보는 AWS SDK for Python (Boto3) API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 하나의 사용자 지정 필터가 있는 계약 검색

다음 코드 예제에서는 하나의 사용자 지정 필터를 적용하여 계약을 검색하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AgreementViewSummary;
import software.amazon.awssdk.services.marketplaceagreement.model.Filter;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsResponse;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import com.example.awsmarketplace.utils.ReferenceCodesUtils;

/**
 * All filter combinations we support for Proposer and Acceptor:
 * https://docs.aws.amazon.com/marketplace/latest/APIReference/API_marketplace-
 * agreements_SearchAgreements.html
 */

public class SearchAgreementsByOneFilter {

    private static final String FILTER_NAME = "ResourceType";
```

```
private static final String FILTER_VALUE = "SaaSProduct";

/*
 * search agreements by one customize filter
 */
public static void main(String[] args) {

    List<AgreementViewSummary> agreementSummaryList = getAgreements();

    ReferenceCodesUtils.formatOutput(agreementSummaryList);
}

public static List<AgreementViewSummary> getAgreements() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    // Set PartyType filter to PARTY_TYPE_FILTER_VALUE_PROPOSER to return
    agreements where you are the proposer.
    // Change to PARTY_TYPE_FILTER_VALUE_ACCEPTOR to return agreements where you
    are the acceptor.
    Filter partyTypeFilter = Filter.builder().name(PARTY_TYPE_FILTER_NAME)
        .values(PARTY_TYPE_FILTER_VALUE_PROPOSER).build();

    Filter agreementTypeFilter = Filter.builder().name(AGREEMENT_TYPE_FILTER_NAME)
        .values(AGREEMENT_TYPE_FILTER_VALUE_PURCHASEAGREEMENT).build();

    Filter customizeFilter =
    Filter.builder().name(FILTER_NAME).values(FILTER_VALUE).build();

    List<Filter> filters = new ArrayList<Filter>();

    filters.addAll(Arrays.asList(partyTypeFilter, agreementTypeFilter,
    customizeFilter));

    SearchAgreementsRequest searchAgreementsRequest =
        SearchAgreementsRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .filters(filters)
            .build();

    SearchAgreementsResponse searchAgreementsResponse =
    marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);
}
```

```

    List<AgreementViewSummary> agreementSummaryList = new
    ArrayList<AgreementViewSummary>();

    agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());

    while (searchAgreementsResponse.nextToken() != null &&
    searchAgreementsResponse.nextToken().length() > 0) {
        searchAgreementsRequest =
            SearchAgreementsRequest.builder()
                .catalog(AWS_MP_CATALOG)
                .filters(filters)
                .nextToken(searchAgreementsResponse.nextToken())
                .build();
        searchAgreementsResponse =
            marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

        agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());
    }
    return agreementSummaryList;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 두 개의 사용자 지정 필터가 있는 계약 검색

다음 코드 예제에서는 두 개의 사용자 지정 필터를 적용하여 계약을 검색하는 방법을 보여줍니다.

Java

SDK for Java 2.x

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.seller;

import software.amazon.awssdk.auth.credentials.AwsCredentialsProvider;
import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.SdkHttpClient;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AgreementViewSummary;
import software.amazon.awssdk.services.marketplaceagreement.model.Filter;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.SearchAgreementsResponse;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import static com.example.awsmarketplace.utils.ReferenceCodesConstants.*;
import com.example.awsmarketplace.utils.ReferenceCodesUtils;

/**
 * All filter combinations we support for Proposer and Acceptor:
 * https://docs.aws.amazon.com/marketplace/latest/APIReference/API_marketplace-
 * agreements_SearchAgreements.html
 */

public class SearchAgreementsByTwoFilters {

    public static final String FILTER_1_NAME = "ResourceType";
```



```
public static final String FILTER_1_VALUE = "SaaSProduct";

public static final String FILTER_2_NAME = "Status";

public static final String FILTER_2_VALUE = "ACTIVE";

/*
 * search agreements by two customize filter
 */
public static void main(String[] args) {

    List<AgreementViewSummary> agreementSummaryList = getAgreements();

    ReferenceCodesUtils.formatOutput(agreementSummaryList);

}

public static List<AgreementViewSummary> getAgreements() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    // Set PartyType filter to PARTY_TYPE_FILTER_VALUE_PROPOSER to return
    agreements where you are the proposer.
    // Change to PARTY_TYPE_FILTER_VALUE_ACCEPTOR to return agreements where you
    are the acceptor.
    Filter partyTypeFilter = Filter.builder().name(PARTY_TYPE_FILTER_NAME)
        .values(PARTY_TYPE_FILTER_VALUE_PROPOSER).build();

    Filter agreementTypeFilter = Filter.builder().name(AGREEMENT_TYPE_FILTER_NAME)
        .values(AGREEMENT_TYPE_FILTER_VALUE_PURCHASEAGREEMENT).build();

    Filter customizeFilter1 =
        Filter.builder().name(FILTER_1_NAME).values(FILTER_1_VALUE).build();

    Filter customizeFilter2 =
        Filter.builder().name(FILTER_2_NAME).values(FILTER_2_VALUE).build();

    List<Filter> filters = new ArrayList<Filter>();
```

```

filters.addAll(Arrays.asList(partyTypeFilter, agreementTypeFilter,
customizeFilter1, customizeFilter2));

SearchAgreementsRequest searchAgreementsRequest =
    SearchAgreementsRequest.builder()
        .catalog(AWS_MP_CATALOG)
        .filters(filters)
        .build();

SearchAgreementsResponse searchAgreementsResponse =
marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

List<AgreementViewSummary> agreementSummaryList = new
ArrayList<AgreementViewSummary>();

agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());

while (searchAgreementsResponse.nextToken() != null &&
searchAgreementsResponse.nextToken().length() > 0) {
    searchAgreementsRequest =
        SearchAgreementsRequest.builder()
            .catalog(AWS_MP_CATALOG)
            .filters(filters)
            .nextToken(searchAgreementsResponse.nextToken())
            .build();
    searchAgreementsResponse =
marketplaceAgreementClient.searchAgreements(searchAgreementsRequest);

    agreementSummaryList.addAll(searchAgreementsResponse.agreementViewSummaries());
}
return agreementSummaryList;
}
}

```

- API 세부 정보는 AWS SDK for Java 2.x API 참조의 [SearchAgreements](#)를 참조하세요.

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

AWSSDK를 사용하여 계약 수락 후 구매 주문 업데이트

다음 코드 예제에서는 계약이 수락된 후 구매 주문을 업데이트하는 방법을 보여줍니다.

Java

SDK for Java 2.x

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
package com.example.awsmarketplace.agreementapi.buyer;

import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
import software.amazon.awssdk.http.apache.ApacheHttpClient;
import
    software.amazon.awssdk.services.marketplaceagreement.MarketplaceAgreementClient;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementRequestResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Charge;
import
    software.amazon.awssdk.services.marketplaceagreement.model.ConfigurableUpfrontPricingTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.CreateAgreementRequestResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.Dimension;
import software.amazon.awssdk.services.marketplaceagreement.model.Intent;
import
    software.amazon.awssdk.services.marketplaceagreement.model.ListAgreementChargesRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.ListAgreementChargesResponse;
import software.amazon.awssdk.services.marketplaceagreement.model.PurchaseOrder;
import software.amazon.awssdk.services.marketplaceagreement.model.RequestedTerm;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RequestedTermConfiguration;
```

```

import
    software.amazon.awssdk.services.marketplaceagreement.model.UpdatePurchaseOrdersRequest;
import utils.AgreementApiUtils;

/**
 * Demonstrates how to associate a purchase order reference with a SaaS agreement
 * with CONTRACT pricing model
 * using the AWS Marketplace Agreement Service APIs.
 *
 * <p>Scenario: A buyer creates a SaaS agreement request with CONTRACT pricing
 * model and provides a purchase order
 * reference in AcceptAgreementRequest. After acceptance, the buyer lists the
 * resulting
 * charges via ListAgreementCharges and associates the purchase order reference
 * with a
 * specific charge via UpdatePurchaseOrders.
 *
 * <p>Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
 * <ul>
 * <li>{@code AGREEMENT_PROPOSAL_IDENTIFIER} – the agreementProposalId from the
 * offer.</li>
 * <li>Term IDs (starting with {@code term-}) – found in the offer's term
 * list.</li>
 * <li>{@code SELECTOR_VALUE} – duration for the agreement.</li>
 * <li>{@code DIMENSION_1_KEY} – dimension key defined in the offer.</li>
 * <li>{@code PURCHASE_ORDER_REFERENCE} – your internal purchase order number
 * (e.g., {@code po-123456}).</li>
 * </ul>
 */
public class UpdatePurchaseOrdersAfterAgreementAcceptance {

    // Your internal purchase order reference number (e.g., "po-123456").
    private static final String PURCHASE_ORDER_REFERENCE = "po-123456";

    // The agreementProposalId from the offer.
    private static final String AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-
proposal-identifier>";

    // Term ID for the ConfigurableUpfrontPricingTerm in your offer.
    private static final String CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-
configurable-upfront-pricing-term-id>";

```

```

// Duration for the agreement (e.g., "P366D" for 366 days).
private static final String SELECTOR_VALUE = "<your-selector-value>";

// Dimension key and quantity defined in your offer.
private static final String DIMENSION_1_KEY = "<your-dimension-key>";
private static final int DIMENSION_1_VALUE = 1;

// Term ID for the LegalTerm in your offer.
private static final String LEGAL_TERM_ID = "<your-legal-term-id>";

// Term ID for the ValidityTerm in your offer.
private static final String VALIDITY_TERM_ID = "<your-validity-term-id>";

public static void main(String[] args) {
    listAgreementChargesAndUpdatePurchaseOrders();
}

/**
 * Full end-to-end flow:
 * 1. Create a SaaS agreement with CONTRACT pricing model with a purchase
order reference.
 * 2. List charges to retrieve charge IDs and revisions.
 * 3. Associate the purchase order reference with a specific charge via
UpdatePurchaseOrders.
 * 4. List charges again to confirm the update.
 */
private static void listAgreementChargesAndUpdatePurchaseOrders() {
    MarketplaceAgreementClient marketplaceAgreementClient =
        MarketplaceAgreementClient.builder()
            .httpClient(ApacheHttpClient.builder().build())
            .credentialsProvider(ProfileCredentialsProvider.create())
            .build();

    RequestedTerm configurableUpfrontPricingTerm = RequestedTerm.builder()
        .id(CONFIGURABLE_UPFRONT_PRICING_TERM_ID)

        .configuration(RequestedTermConfiguration.fromConfigurableUpfrontPricingTermConfiguration(
            ConfigurableUpfrontPricingTermConfiguration.builder()
                .selectorValue(SELECTOR_VALUE)
                .dimensions(Dimension.builder()

                    .dimensionKey(DIMENSION_1_KEY)

                    .dimensionValue(DIMENSION_1_VALUE)

```

```

        .build()))
        .build()))
        .build();

        RequestedTerm legalTerm =
        RequestedTerm.builder().id(LEGAL_TERM_ID).build();
        RequestedTerm validityTerm =
        RequestedTerm.builder().id(VALIDITY_TERM_ID).build();

        // --- Create Agreement ---
        CreateAgreementRequestRequest createAgreementRequestRequest =
            CreateAgreementRequestRequest.builder()
                .clientToken(AgreementApiUtils.generateClientToken())
                .intent(Intent.NEW)
                .requestedTerms(configurableUpfrontPricingTerm,
legalTerm, validityTerm)

            .agreementProposalIdentifier(AGREEMENT_PROPOSAL_IDENTIFIER)
                .build();
        CreateAgreementRequestResponse createAgreementRequestResponse =

marketplaceAgreementClient.createAgreementRequest(createAgreementRequestRequest);
        System.out.println("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId());

        // --- Accept Agreement Request with Purchase Order ---
        // The chargeId is available from the CAR response's
chargeSummary.expectedCharges.
        String chargeId =
createAgreementRequestResponse.chargeSummary().expectedCharges().get(0).id();
        PurchaseOrder purchaseOrderAtAcceptance = PurchaseOrder.builder()
            .chargeId(chargeId)
            .purchaseOrderReference(PURCHASE_ORDER_REFERENCE)
            .build();
        AcceptAgreementRequestRequest acceptAgreementRequestRequest =
            AcceptAgreementRequestRequest.builder()

            .agreementRequestId(createAgreementRequestResponse.agreementRequestId())
                .purchaseOrders(purchaseOrderAtAcceptance)
                .build();
        AcceptAgreementRequestResponse acceptAgreementRequestResponse =

marketplaceAgreementClient.acceptAgreementRequest(acceptAgreementRequestRequest);
        final String agreementId = acceptAgreementRequestResponse.agreementId();

```

```

        System.out.println("Agreement request accepted with purchase order
reference '" + PURCHASE_ORDER_REFERENCE
        + "'. AgreementId: " + agreementId);

// --- List Agreement Charges ---
ListAgreementChargesRequest listAgreementChargesRequest =
    ListAgreementChargesRequest.builder()
        .agreementId(agreementId)
        .build();
ListAgreementChargesResponse listAgreementChargesResponse =

marketplaceAgreementClient.listAgreementCharges(listAgreementChargesRequest);

System.out.println("All charges for agreement " + agreementId + ":");
AgreementApiUtils.formatOutput(listAgreementChargesResponse);

// --- Update Purchase Order ---
Charge firstCharge = listAgreementChargesResponse.items().get(0);
PurchaseOrder purchaseOrder = PurchaseOrder.builder()
    .agreementId(agreementId)
    .purchaseOrderReference(PURCHASE_ORDER_REFERENCE)
    .chargeRevision(firstCharge.revision())
    .chargeId(firstCharge.id())
    .build();
UpdatePurchaseOrdersRequest updatePurchaseOrdersRequest =
    UpdatePurchaseOrdersRequest.builder()
        .purchaseOrders(purchaseOrder)
        .build();

marketplaceAgreementClient.updatePurchaseOrders(updatePurchaseOrdersRequest);
System.out.println("Purchase order reference '" +
PURCHASE_ORDER_REFERENCE
        + "' updated for ChargeId: " + firstCharge.id());

// --- Verify Update ---
ListAgreementChargesRequest lacRequest =
    ListAgreementChargesRequest.builder()
        .agreementId(agreementId)
        .build();
ListAgreementChargesResponse lacResponse =
    marketplaceAgreementClient.listAgreementCharges(lacRequest);
System.out.println("Verified updated charge:");
AgreementApiUtils.formatOutput(lacResponse.items().get(0));
}

```

```
}

```

- API 세부 정보는 API 참조의 [UpdatePurchaseOrders](#) AWS SDK for Java 2.x를 참조하세요.

JavaScript

SDK for JavaScript (v3)

Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```
const {
  MarketplaceAgreementClient,
  CreateAgreementRequestCommand,
  AcceptAgreementRequestCommand,
  ListAgreementChargesCommand,
  UpdatePurchaseOrdersCommand,
} = require("@aws-sdk/client-marketplace-agreement");
const { generateClientToken, formatOutput } = require("../utils/AgreementApiUtils");

/**
 * Demonstrates how to associate a purchase order reference with a SaaS agreement
 * with CONTRACT pricing model
 * using the AWS Marketplace Agreement Service APIs.
 *
 * Scenario: A buyer creates a SaaS agreement request with CONTRACT pricing model
 * and provides a purchase order
 * reference in AcceptAgreementRequest. After acceptance, the buyer lists the
 * resulting
 * charges via ListAgreementCharges and associates the purchase order reference
 * with a
 * specific charge via UpdatePurchaseOrders.
 *
 * Before running this sample, replace the placeholder constants below with
 * values from
 * your AWS Marketplace offer:
 * - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
```



```

*   - Term IDs (starting with "term-") – found in the offer's term list.
*   - SELECTOR_VALUE – duration for the agreement.
*   - DIMENSION_1_KEY – dimension key defined in the offer.
*   - PURCHASE_ORDER_REFERENCE – your internal purchase order number (e.g.,
    "po-123456").
*/

// Your internal purchase order reference number (e.g., "po-123456").
const PURCHASE_ORDER_REFERENCE = "po-123456";

// The agreementProposalId from the offer.
const AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>";

// Term ID for the ConfigurableUpfrontPricingTerm in your offer.
const CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-
term-id>";

// Duration for the agreement (e.g., "P366D" for 366 days).
const SELECTOR_VALUE = "<your-selector-value>";

// Dimension key and quantity defined in your offer.
const DIMENSION_1_KEY = "<your-dimension-key>";
const DIMENSION_1_VALUE = 1;

// Term ID for the LegalTerm in your offer.
const LEGAL_TERM_ID = "<your-legal-term-id>";

// Term ID for the ValidityTerm in your offer.
const VALIDITY_TERM_ID = "<your-validity-term-id>";

/**
 * Full end-to-end flow:
 * 1. Create a SaaS agreement with CONTRACT pricing model with a purchase order
    reference.
 * 2. List charges to retrieve charge IDs and revisions.
 * 3. Associate the purchase order reference with a specific charge via
    UpdatePurchaseOrders.
 * 4. List charges again to confirm the update.
 */
async function listAgreementChargesAndUpdatePurchaseOrders() {
    const client = new MarketplaceAgreementClient();

    const configurableUpfrontPricingTerm = {
        id: CONFIGURABLE_UPFRONT_PRICING_TERM_ID,

```

```

        configuration: {
            configurableUpfrontPricingTermConfiguration: {
                selectorValue: SELECTOR_VALUE,
                dimensions: [
                    { dimensionKey: DIMENSION_1_KEY, dimensionValue:
DIMENSION_1_VALUE },
                ],
            },
        },
    };

    const legalTerm = { id: LEGAL_TERM_ID };
    const validityTerm = { id: VALIDITY_TERM_ID };

    // --- Create Agreement ---
    const createAgreementRequestResponse = await client.send(
        new CreateAgreementRequestCommand({
            clientToken: generateClientToken(),
            intent: "NEW",
            requestedTerms: [configurableUpfrontPricingTerm, legalTerm,
validityTerm],
            agreementProposalIdentifier: AGREEMENT_PROPOSAL_IDENTIFIER,
        })
    );
    console.log("Agreement request created. AgreementRequestId: " +
createAgreementRequestResponse.agreementRequestId);

    // --- Accept Agreement Request with Purchase Order ---
    // The chargeId is available from the CAR response's
chargeSummary.expectedCharges.
    const chargeId =
createAgreementRequestResponse.chargeSummary.expectedCharges[0].id;
    const purchaseOrderAtAcceptance = {
        chargeId: chargeId,
        purchaseOrderReference: PURCHASE_ORDER_REFERENCE,
    };
    const acceptAgreementRequestResponse = await client.send(
        new AcceptAgreementRequestCommand({
            agreementRequestId:
createAgreementRequestResponse.agreementRequestId,
            purchaseOrders: [purchaseOrderAtAcceptance],
        })
    );
    const agreementId = acceptAgreementRequestResponse.agreementId;

```

```

    console.log("Agreement request accepted with purchase order reference '" +
PURCHASE_ORDER_REFERENCE
        + "'. AgreementId: " + agreementId);

    // --- List Agreement Charges ---
    const listAgreementChargesResponse = await client.send(
        new ListAgreementChargesCommand({
            agreementId: agreementId,
        })
    );

    console.log("All charges for agreement " + agreementId + ":");
    formatOutput(listAgreementChargesResponse);

    // --- Update Purchase Order ---
    const firstCharge = listAgreementChargesResponse.items[0];
    const purchaseOrder = {
        agreementId: agreementId,
        purchaseOrderReference: PURCHASE_ORDER_REFERENCE,
        chargeRevision: firstCharge.revision,
        chargeId: firstCharge.id,
    };
    await client.send(
        new UpdatePurchaseOrdersCommand({
            purchaseOrders: [purchaseOrder],
        })
    );
    console.log("Purchase order reference '" + PURCHASE_ORDER_REFERENCE
        + "' updated for ChargeId: " + firstCharge.id);

    // --- Verify Update ---
    const lacResponse = await client.send(
        new ListAgreementChargesCommand({
            agreementId: agreementId,
        })
    );
    console.log("Verified updated charge:");
    formatOutput(lacResponse.items[0]);
}

listAgreementChargesAndUpdatePurchaseOrders();

```

- API 세부 정보는 API 참조의 [UpdatePurchaseOrders](#) AWS SDK for JavaScript를 참조하세요.

Python

SDK for Python(Boto3)

 Note

GitHub에 더 많은 내용이 있습니다. [AWS Marketplace API 참조 코드 라이브러리](#) 리포지토리에서 전체 예제를 확인하고 설정 및 실행하는 방법을 알아보세요.

```

"""
Demonstrates how to associate a purchase order reference with a SaaS agreement
with CONTRACT
pricing model using the AWS Marketplace Agreement Service APIs.

Scenario: A buyer creates a SaaS agreement request with CONTRACT pricing model
and provides
a purchase order reference in AcceptAgreementRequest. After acceptance, the buyer
lists the
resulting charges via ListAgreementCharges and associates the purchase order
reference with a
specific charge via UpdatePurchaseOrders.

Before running this sample, replace the placeholder constants below with values
from
your AWS Marketplace offer:
    - AGREEMENT_PROPOSAL_IDENTIFIER – the agreementProposalId from the offer.
    - Term IDs (starting with term-) – found in the offer's term list.
    - SELECTOR_VALUE – duration for the agreement.
    - DIMENSION_1_KEY – dimension key defined in the offer.
    - PURCHASE_ORDER_REFERENCE – your internal purchase order number (e.g.,
    po-123456).
"""

import boto3

from utils.agreement_api_utils import format_output, generate_client_token

class UpdatePurchaseOrdersAfterAgreementAcceptance:

    # Your internal purchase order reference number (e.g., "po-123456").

```

```

PURCHASE_ORDER_REFERENCE = "po-123456"

# The agreementProposalId from the offer.
AGREEMENT_PROPOSAL_IDENTIFIER = "<your-agreement-proposal-identifier>"

# Term ID for the ConfigurableUpfrontPricingTerm in your offer.
CONFIGURABLE_UPFRONT_PRICING_TERM_ID = "<your-configurable-upfront-pricing-
term-id>"

# Duration for the agreement (e.g., "P366D" for 366 days).
SELECTOR_VALUE = "<your-selector-value>"

# Dimension key and quantity defined in your offer.
DIMENSION_1_KEY = "<your-dimension-key>"
DIMENSION_1_VALUE = 1

# Term ID for the LegalTerm in your offer.
LEGAL_TERM_ID = "<your-legal-term-id>"

# Term ID for the ValidityTerm in your offer.
VALIDITY_TERM_ID = "<your-validity-term-id>"

@staticmethod
def list_agreement_charges_and_update_purchase_orders():
    """
    Full end-to-end flow:
    1. Create a SaaS agreement with CONTRACT pricing model with a purchase
order reference.
    2. List charges to retrieve charge IDs and revisions.
    3. Associate the purchase order reference with a specific charge via
UpdatePurchaseOrders.
    4. List charges again to confirm the update.
    """
    client = boto3.client("marketplace-agreement")
    cls = UpdatePurchaseOrdersAfterAgreementAcceptance

    configurable_upfront_pricing_term = {
        "id": cls.CONFIGURABLE_UPFRONT_PRICING_TERM_ID,
        "configuration": {
            "configurableUpfrontPricingTermConfiguration": {
                "selectorValue": cls.SELECTOR_VALUE,
                "dimensions": [
                    {
                        "dimensionKey": cls.DIMENSION_1_KEY,

```

```

        "dimensionValue": cls.DIMENSION_1_VALUE,
    },
],
}

legal_term = {"id": cls.LEGAL_TERM_ID}
validity_term = {"id": cls.VALIDITY_TERM_ID}

# --- Create Agreement ---
create_response = client.create_agreement_request(
    clientToken=generate_client_token(),
    intent="NEW",
    requestedTerms=[configurable_upfront_pricing_term, legal_term,
validity_term],
    agreementProposalIdentifier=cls.AGREEMENT_PROPOSAL_IDENTIFIER,
)
agreement_request_id = create_response["agreementRequestId"]
print("Agreement request created. AgreementRequestId: " +
agreement_request_id)

# --- Accept Agreement Request with Purchase Order ---
# The chargeId is available from the CAR response's
chargeSummary.expectedCharges.
charge_id = create_response["chargeSummary"]["expectedCharges"][0]["id"]
accept_response = client.accept_agreement_request(
    agreementRequestId=agreement_request_id,
    purchaseOrders=[
        {
            "chargeId": charge_id,
            "purchaseOrderReference": cls.PURCHASE_ORDER_REFERENCE,
        }
    ],
)
agreement_id = accept_response["agreementId"]
print(
    "Agreement request accepted with purchase order reference '"
    + cls.PURCHASE_ORDER_REFERENCE + "'. AgreementId: " + agreement_id
)

# --- List Agreement Charges ---
list_charges_response =
client.list_agreement_charges(agreementId=agreement_id)

```

```

print("All charges for agreement " + agreement_id + ":")
format_output(list_charges_response)

# --- Update Purchase Order ---
first_charge = list_charges_response["items"][0]
client.update_purchase_orders(
    purchaseOrders=[
        {
            "agreementId": agreement_id,
            "purchaseOrderReference": cls.PURCHASE_ORDER_REFERENCE,
            "chargeRevision": first_charge["revision"],
            "chargeId": first_charge["id"],
        }
    ],
)
print(
    "Purchase order reference '" + cls.PURCHASE_ORDER_REFERENCE
    + "' updated for ChargeId: " + first_charge["id"]
)

# --- Verify Update ---
lac_response = client.list_agreement_charges(agreementId=agreement_id)
print("Verified updated charge:")
format_output(lac_response["items"][0])

if __name__ == "__main__":

    UpdatePurchaseOrdersAfterAgreementAcceptance.list_agreement_charges_and_update_purchase_

```

- API 세부 정보는 SDK for Python (Boto3) API 참조의 [UpdatePurchaseOrders](#)를 참조하세요.
AWS

AWSSDK 개발자 안내서 및 코드 예제의 전체 목록은 섹션을 참조하세요 [AWSSDK에서이 서비스 사용](#). 이 주제에는 시작하기에 대한 정보와 이전 SDK 버전에 대한 세부 정보도 포함되어 있습니다.

문서 기록

다음 표는 이 안내서에 대한 문서 변경 사항을 설명합니다.

변경 사항	설명	날짜
AWS Marketplace Discovery API 퍼블릭 시작	이제 AWS Marketplace Discovery API를 사용할 수 있습니다. GetListing, GetProduct, GetOffer, GetOfferTerms, GetOfferSet, ListPurchaseOptions, ListFulfillmentOptions, SearchFacets 및 SearchListings의 9가지 API 작업이 추가되었습니다. 표준 AWS SDK 지원 및 IAM 기반 액세스.	2026년 4월 8일
ML 제품 업데이트	판매자 제품 기계 학습 정보가 업데이트되었습니다.	2025년 3월 26일
CustomerIdentifier	2026년 3월에 CustomerIdentifier 사용 중단 공지가 추가되었습니다.	2025년 3월 25일
AWS Marketplace Catalog API 주제 업데이트	CPPO 사전 조건을 업데이트했습니다.	2024년 5월 9일
AWS Marketplace Catalog API 주제 업데이트	Details 속성의 인스턴스를 로 업데이트했습니다DetailsDocument .	2024년 4월 30일
AWS Marketplace Catalog API 주제 업데이트	계약 조건을 명확히 하기 위해 제안 작업 에 참고 사항이 추가되었습니다.	2024년 4월 17일

AWS Marketplace Catalog API는 재판매 권한 부여를 위한 서비스 연결 지원	서비스 연결 역할에 대한 재판매 권한 부여 사전 조건을 업데이트했습니다.	2024년 3월 20일
AWS Marketplace Catalog API는 조직 단위를 지원합니다.	조직 단위(OU) 수준에서 프라이빗 마켓플레이스 지원을 활성화하는 콘텐츠가 추가되었습니다.	2024년 2월 16일
AWS Marketplace Catalog API는 요청에 대한 의도 설정을 지원합니다.	이제 판매자를 사용하여 특정 의도의 엔터티에 대한 변경을 요청할 수 있습니다 AWS Marketplace Catalog API.	2024년 2월 9일
AWS Marketplace Catalog API는 와일드카드 필터 검증을 지원합니다.	ListEntities API에 와일드카드 필터 검증이 추가되었습니다.	2024년 2월 5일
AWS Marketplace Catalog API에서 Amazon EKS 추가 기능 지원	AWS Marketplace 컨테이너 기반 제품의 Amazon EKS 추가 기능에 게시와 관련된 콘텐츠 및 오류 메시지가 추가되었습니다.	2024년 1월 29일
AWS Marketplace Catalog API는 엔터티에 대한 세부 정보 나열을 지원합니다.	이제 판매자를 사용하여 엔터티에 대한 세부 정보를 나열할 수 있습니다 AWS Marketplace Catalog API.	2023년 12월 19일
이제 AWS Marketplace 배포 서비스 API 참조를 정식으로 사용할 수 있습니다.	이 서비스는 빠른 시작 경험 중에 배포 파라미터(예: API 키 및 외부 IDs)를 전달하기 위한 안전한 방법을 지원하는 API 인터페이스를 제공합니다.	2023년 11월 29일

[이제 AWS Marketplace 계약 서비스 API 참조를 정식으로 사용할 수 있습니다.](#)

이 서비스는 AWS Marketplace 판매자가 계약 나열, 검색 및 필터링을 포함하여 제품 관련 계약을 관리할 수 있도록 지원하는 API 인터페이스를 제공합니다.

2023년 11월 29일

[AWS Marketplace Catalog API는 제품, 제안, 재판매 권한 부여 및 CPPOs](#)

이제 판매자는 AWS Marketplace Catalog API를 사용하여 [제품](#), [제안](#), [재판매 승인](#) 및 [채널 파트너 비공개 제안\(CPPOs\)](#).

2023년 11월 29일

[AWS Marketplace Catalog API는 향상된 필터링 및 정렬 기능을 지원합니다.](#)

이제 판매자는를 사용하여 제품을 정렬하고 필터링할 수 있습니다 AWS Marketplace Catalog API.

2023년 11월 29일

[AWS Marketplace Catalog API에서 리소스 공유 지원](#)

는 AWS Resource Access Manager(AWS RAM)와 AWS Marketplace Catalog API 통합되어 리소스 공유를 활성화합니다. [리소스 공유 작업을 참조 AWS RAM 하세요.](#)

2023년 8월 12일

[AWS Marketplace Discovery API 주제 업데이트](#)

AWS Marketplace Discovery API는 이제 CloudTrail을 지원합니다. 를 [사용하여 AWS Marketplace Discovery API 호출 로깅 AWS CloudTrail](#)을 참조하세요.

2022년 12월 15일

[AWS Marketplace는 프라이빗 마켓플레이스 경험 아카이빙을 지원합니다.](#)

이제 구매자는 AWS Marketplace에서 프라이빗 마켓플레이스 경험을 보관하고 다시 활성화할 수 있습니다. [프라이빗 마켓플레이스 작업을 참조하세요.](#)

2022년 12월 12일

AWS Marketplace 프라이빗 마켓플레이스 세분화된 권한	이제 구매자는 프라이빗 마켓플레이스 경험을 관리할 수 있는 보다 세분화된 권한을 갖게 되었습니다. 프라이빗 마켓플레이스 작업을 참조하세요.	2022년 9월 8일
AWS Marketplace Discovery API 릴리스 정보	AWS Marketplace Discovery API에 대한 릴리스 정보가 추가되었습니다.	2022년 5월 20일
AWS Marketplace Discovery API 주제 업데이트	AWS Marketplace Discovery API 주제 에 대한 설명서 전용 업데이트입니다.	2022년 1월 14일
컨테이너 기반 제품에 Helm 차트 전송 옵션 및 QuickLaunch 지원	구매자를 위한 QuickLaunch 활성화를 포함하여 컨테이너 기반 제품 버전에서 Helm 차트 전송 옵션을 추가하거나 업데이트하기 위한 설명서가 추가되었습니다. 컨테이너 기반 제품 작업을 참조하세요.	2021년 11월 29일
판매자 제품 관리 지원	프로그래밍 방식으로 AMI 및 컨테이너 제품을 관리하는 기능이 추가되었습니다. 판매자 제품 작업을 참조하세요.	2021년 3월 26일
프라이빗 마켓플레이스 관리 지원	프로그래밍 방식으로 AWS Organizations 용 프라이빗 마켓플레이스를 생성하고 유지 관리하는 기능이 추가되었습니다. 프라이빗 마켓플레이스 작업을 참조하세요.	2020년 12월 3일

[이제 AWS Marketplace Discovery API를 사용할 수 있습니다.](#)

Discovery API는에서 제품을 찾을 수 있는 프로그래밍 방식의 액세스를 제공합니다 AWS Marketplace. 자세한 내용은 [Discovery API](#)를 참조하세요.

2020년 9월 30일

[이제 AWS Marketplace Catalog API 를 정식 버전으로 사용할 수 있습니다.](#)

이 서비스는 승인된 공급자가 의 셀프 서비스 게시 기능에 프로그래밍 방식으로 액세스할 수 있는 API 인터페이스를 제공합니다 AWS Marketplace Management Portal.

2019년 11월 12일

AWS Marketplace Discovery API 릴리스 정보

AWS Marketplace Discovery API에 대한 릴리스 정보입니다. 새로운 기능, 개선 사항, 수정 사항 및 공지 사항에 대한 세부 정보입니다.

2026년 Discovery API 릴리스 정보

2026년 4월 8일

이제 표준 AWS SDK를 통해 AWS Marketplace Discovery API를 공개적으로 사용할 수 있습니다. API는 목록 검색 및 검색, 제품 세부 정보 및 이행 옵션 검색, 공개 및 비공개 제안 요금 및 약관 액세스를 포함하여 AWS Marketplace 카탈로그에 대한 프로그래밍 방식의 액세스를 제공합니다.

출시 발표

- 사용 가능한 9가지 API 작업: GetListing, GetProduct, GetOffer, GetOfferTerms, GetOfferSet, ListPurchaseOptions, ListFulfillmentOptions, SearchFacets 및 SearchListings.
- 표준 AWS SDK는 모든 언어(Java, Python, JavaScript, .NET, Go 등)를 지원합니다. 프라이빗 SDK는 필요하지 않습니다.
- 리소스 수준 권한이 있는 IAM 기반 액세스 제어. 수동 온보딩은 필요하지 않습니다.
- 미국 동부(버지니아 북부), 미국 서부(오레곤) 및 유럽(아일랜드)에서 사용할 수 있습니다.

2024년 Discovery API 릴리스 정보

2024년 8월 15일

Discovery API 고객은 Discovery API 팀이 제공하는 Amazon Simple Storage Service 버킷에서 업데이트된 Discovery API 설명서 및 SDK에 액세스할 수 있습니다. 자세한 내용은 프라이빗 설명서의 변경 로그를 참조하세요.

출시 발표

이제 JavaScript V3에서 Discovery API Private SDK를 사용할 수 있습니다.

- AWS SDK for JavaScript V3 사용에 대한 자세한 내용은 [AWS SDK for JavaScript V3 개발자 안내서](#)를 참조하세요.
- V2에서 V3로 마이그레이션하는 방법에 대한 자세한 내용은 [JavaScript용 SDK V3 개발자 안내서의 JavaScript용 AWS SDK 버전 2.x에서 3.x로 마이그레이션](#)을 참조하세요. AWS JavaScript V3

개선 사항

- AWS 이제 SDK 페이지네이터를 기본적으로 지원하는 지원되는 모든 언어로 SearchListings 및 ListListingViewQueries API 작업에 SDK 페이지네이터를 사용할 수 있습니다.

를 사용하여 수동으로 페이지 매김NextToken할 수 있습니다. 를 벤딩하는 모든 SDKs에서 SearchListingsListListingViewQueries, 및 GetSearchFacets API 작업과 NextToken 함께 사용할 수 있습니다.

- 지원되는 모든 언어에 대해 SDK가 최신 AWS SDK 아티팩트로 업데이트되었습니다.
- JavaScript V3용 SDK 사용 섹션을 포함하도록 프라이빗 설명서가 업데이트되었습니다.

2022년 Discovery API 릴리스 정보

2022년 5월 20일

2022년 5월 20일 게시

Discovery API 고객은 Discovery API 팀이 이전에 공유한 Amazon Simple Storage Service(Amazon S3) 버킷에서 업데이트된 Discovery API 설명서 및 SDK에 액세스할 수 있습니다. 자세한 내용은 프라이빗 설명서의 변경 로그를 참조하세요.

Discovery API는 다음과 같은 출시 및 개선 사항을 발표했습니다.

출시 발표

- Discovery API는 AWS 리전다음 두 가지 추가 기능에서 시작되었습니다.
 - 미국 서부(오리건) – us-west-2
 - 유럽(아일랜드) – eu-west-1
- 이제 Java 2.x에서 Discovery API Private SDK를 사용할 수 있습니다.
 - 사용 방법에 대한 자세한 내용은 [AWS SDK for Java 2.x 개발자 안내서](#)를 AWS SDK for Java 2.x 참조하세요.
 - 마이그레이션에 대한 자세한 내용은 AWS SDK for Java 2.x 개발자 안내서의 [버전 1.x에서 2.x로 마이그레이션 AWS SDK for Java](#)을 참조하세요.
 - 버전 1.11.x와 2.x 간의 변경 사항에 대한 자세한 내용은 GitHub 웹 사이트의 [1.11~2.x Changelog](#)를 AWS SDK for Java 2.x참조하세요.

개선 사항

- 다음에 대한 새로운 옵션을 도입하여 SearchListings API 작업에 대한 정렬 기능을 개선했습니다.
 - SortBy – AVERAGE_CUSTOMER_RATING, CREATION_TIME, LAST_MODIFIED_TIME
 - SortOrder – ASCENDING
- 최신 AWS SDK 아티팩트로 모든 기존 언어에 맞게 SDK가 업데이트되었습니다.
- 모든 언어에 대한 SDK 사용 섹션을 포함하도록 설명서가 업데이트되었습니다.