



Developer Guide

AWS Marketplace



AWS Marketplace: Developer Guide

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

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

.....	viii
Welcome	1
About the APIs	2
Using the APIs	2
Using the Catalog API	2
Using the Agreement API	18
Using the Metering API	26
Using the Entitlement API	28
Using the Deployment API	28
Using the Discovery API	28
Supported Regions	31
Regions for the Catalog API	31
Regions for the Agreement API	31
Regions for the Metering API	32
Regions for the Entitlement API	34
Regions for the Deployment API	34
Regions for the Discovery API	34
Permissions	34
Access for the Catalog API	35
Access for the Agreement API	52
Access for the Metering API	58
Access for the Entitlement API	58
Access for the Deployment API	58
Access for the Compliance API	59
Access for the Discovery API	73
Access control for the AWS Marketplace Reporting API	76
Quotas	79
Quotas for the Catalog API	79
Quotas for the Agreement API	82
Quotas for the Entitlement API	84
Quotas for the Deployment API	84
Quotas for the Metering API	85
Quotas for the Discovery API	86
Logging	87

Logging the Catalog API	87
Logging the Agreement API	90
Logging the Metering API	96
Logging the Deployment API	103
Logging the Discovery API	106
Notifications	107
Notifications for the Catalog API	108
AWS Marketplace MCP server	109
Available tools	110
Integration guide	110
Claude Connector (recommended)	111
MCP client integration	111
Tool details	115
get_aws_marketplace_report_guidelines	115
search_aws_marketplace_solutions	117
get_aws_marketplace_solution	118
get_aws_marketplace_related_solutions	122
research_aws_marketplace_solution	123
submit_aws_marketplace_feedback	125
Workflows	126
Report generation (recommendations, comparisons, evaluations)	126
Direct catalog lookup	127
Feedback collection	128
End-to-end example	128
Error handling	129
Protocol errors	130
Tool execution errors	130
Validation errors	131
Using the API as a seller	132
Work with seller products	132
Create a product	134
Update product details	137
Add pricing dimensions	145
Update pricing dimensions	152
Restrict pricing dimensions	157
Update targeting configuration	161

Update product visibility	165
Publish a product	169
Find your product ID	171
Change set status and errors	178
Work with AMI-based products	185
Work with EC2 Image Builder component products	218
Work with container-based products	235
Work with SaaS products	275
Work with machine learning products	301
Work with offers	324
Work with private offers	324
Work with resale authorizations	393
Work with channel partner private offers	441
Work with renewals	494
Work with multi-product solutions	496
Work with solutions	497
Work with offer sets	532
Build interfaces with catalog data	556
Architecture overview	556
Build a browse experience	557
Build product detail pages	558
Display pricing information	558
Best practices	559
Using the API as a buyer	560
Discover products and pricing	560
Browse the catalog	560
Search for listings	560
Get listing details	561
Get product details and fulfillment options	561
Find available offers for a product	562
Get detailed pricing terms	562
Get offer set details	562
Access private offer pricing	563
Integrate with procurement tools	564
Work with the Agreement API	564
Prerequisites: Discover products and offers	564

Generate a quote	570
Accept an offer	576
Replace an existing agreement	577
Amend an existing agreement	577
Enable or disable auto-renewal	578
Cancel an agreement	578
Search agreements	578
Describe an agreement	579
Get agreement terms	579
Get agreement entitlements	579
Get a registration token	580
List agreement charges	580
Update purchase orders	580
List cancellation requests	581
Get cancellation request details	581
Accept a cancellation request	581
Reject a cancellation request	582
List payment requests	582
Get payment request details	582
Accept a payment request	583
Reject a payment request	583
Work with the private marketplace	583
Creating a private marketplace	585
Changing the branding of a private marketplace experience	589
Enabling or disabling a private marketplace experience	591
Enabling or disabling user requests	592
Getting a list of products in a private marketplace experience	593
Adding or removing products from a private marketplace	596
Finding products	597
Working with private marketplaces for AWS Organizations	598
Associating principals to experiences	600
Archiving and reactivating a private marketplace experience	602
Errors in the private marketplace API	603
Entity types defined by private marketplace	606
Using the API to share resources	608
Prerequisites to share AWS Marketplace entities	608

Share an AWS Marketplace entity	608
Differences between sharing an entity through AWS RAM and the AWS Marketplace	
Catalog API	610
Attach read-only policy to your resource	611
Attach read and write resource policy to your resource	612
View resource policy set on your resource	613
Delete resource policy on your resource	614
View all resources owned by you and shared with you	614
Working with AWS SDKs	616
Code examples	618
AWS Marketplace Catalog API	623
AMI products	627
Channel partner offers	685
Container products	726
Entities	740
Offers	751
Products	888
Resale authorization	900
SaaS products	997
Utilities	1054
AWS Marketplace Agreement API	1061
Agreements	1063
Document history	1321
Release notes for AWS Marketplace Discovery API	1325
Discovery API release notes for 2026	1325
Discovery API release notes for 2024	1326
Discovery API release notes for 2022	1326

The AWS Marketplace API Reference was restructured. For more information about the supported API operations, see the [AWS Marketplace API Reference](#).

Welcome to the AWS Marketplace Developer Guide

This developer guide describes how AWS Marketplace sellers can use service APIs to integrate and manage product lifecycles and offers. It also describes how buyers can use service APIs to discover third-party software, data, and services as well as how to govern private marketplaces.

About the AWS Marketplace APIs

The following sections provide general information on the APIs.

Topics

- [Using the AWS Marketplace API](#)
- [Supported AWS Regions for the AWS Marketplace API](#)
- [Permissions for the AWS Marketplace API](#)
- [Quotas for the AWS Marketplace API](#)
- [Logging for the AWS Marketplace API](#)
- [Notifications for the AWS Marketplace API](#)

Using the AWS Marketplace API

The following sections provide information about using the AWS Marketplace APIs.

Topics

- [Using the AWS Marketplace Catalog API](#)
- [Using the AWS Marketplace Agreement API](#)
- [Using the AWS Marketplace Metering API](#)
- [Using the AWS Marketplace Entitlement API](#)
- [Using the AWS Marketplace Deployment API](#)
- [Using the AWS Marketplace Discovery API](#)

Using the AWS Marketplace Catalog API

The AWS Marketplace Catalog API service provides an API interface to manage AWS Marketplace for your AWS organization or AWS account. For approved sellers, you can manage your products programmatically, including the self-service publishing capabilities on the [AWS Marketplace Management Portal](#). For private marketplace administrators, you can manage your private marketplace programmatically.

With Catalog API actions, you can view and update your existing product programmatically. You can automate your product update process by integrating the AWS Marketplace Catalog API

with your AWS Marketplace product build or deployment pipelines. You can also create your own applications on top of the Catalog API to manage your products in AWS Marketplace. You can manage the products that users in your AWS account or AWS organization can see and purchase through your private marketplace.

The AWS Marketplace Catalog API service provides standard AWS API functionality. You can directly use the REST API actions described in [Actions](#), or you can use an AWS SDK to access an API that's tailored to the programming language or platform that you're using. For more information about AWS application development, see [Getting Started with AWS](#). For more information about using AWS SDKs, see [AWS SDKs](#).

Catalog API entities

AWS Marketplace entities are containers of data which serve different business purposes, such as a product or offer. Entities are categorized by types. Each entity type encapsulates data related to a specific business domain (for example, a product or a seller account).

To simplify this paradigm, entities are designed with some level of commonality in their structures. As a result, introducing a new business domain doesn't require that you learn a completely new structure.

General structure

The general structure of any entity is:

- A named type with a version
- An identifier for the specific instance of the type
- One or more facets that include the attributes of the entity

Type versioning

Every named type has a type and version associated with it, for example, *Entity*Product@1.0. The *type* (*Entity*Product) represents the classification of the content. The *version* (1.0) represents the structure of *Entity*Product.

The version gives you details about the structure of the entity. The following describes when a version will be changed:

- Existing entities won't be restructured without changing the version. Additions of optional new fields will result in a minor version update.

- Any feature that fundamentally changes the structure of a type leads to a major version update. Examples include:
 - Removing a field
 - Renaming a field (different name for the same semantic)
 - Changing the semantic of an existing field (for example, changing the expected type)
- A major version update can retain a subset of facets from the previous version.
- Users are provided notifications and documentation for new versions.

Identifier

Each entity represents a unique *thing* within a business domain. To identify the unique thing, we use an identifier associating an `EntityId` with a `RevisionId`, for example, *prod-ad8EXAMPLE651@3*. In this example, the `EntityId` is `prod-ad8EXAMPLE651` and the `RevisionId` is `3`. Every successful change request to the entity will update the revision.

The following are important details about the identifier:

- Each entity is uniquely identified by its `EntityId`, which is the key to globally distinguish one entity from another.
- Each published revision of an entity has a `RevisionId`. The `RevisionId`, along with the `EntityId`, distinguish one published revision from another.
- AWS Marketplace generates `EntityIds` and `RevisionIds`.

You can use the `DescribeEntity` action to find the details and the Identifier with the most recent `revisionId`.

The `RevisionId` is an optional part of requests to `StartChangeSet` (see [Working with change sets](#)). If you include a `RevisionId`, then the request to `StartChangeSet` will fail with a `ValidationException` if the `RevisionId` is not the latest revision of the entity. This allows you to implement optimistic locking in your application.

Note

When you include a `RevisionId` that is not the latest revision, the `ValidationException` message includes the latest `RevisionId`.

If you omit the `RevisionId`, the request is performed on the latest revision of the entity automatically.

Warning

Two requests to change the same object could result with one request overwriting the changes of the other request, as the second request rewrites data changed by the first request. Using `RevisionIds` in your requests prevents this issue by not allowing a change to an earlier revision to overwrite the current revision.

Facets

A facet is a logical grouping of attributes. An entity usually includes several facets which represent different aspects of the entity. The attributes within a facet have the following properties:

- Each attribute has a unique name within the scope of the container it belongs to.
- Attributes can be of a simple type (string, integer, or floating number).
- Attributes can be of a complex type (container/structure or array).

Entity type

The entity type defines what the entity represents. An entity can be a seller product in AWS Marketplace or a private marketplace. For more information, see [Working with seller products](#) and [Working with a private marketplace](#).

Working with change sets

When using the Catalog API, requests are created and updated through entities and completed by using change requests. Every change specifies the entity to be changed, the type of change to be performed, and details of the change. The type of change to be performed is called a `ChangeType`. A collection of `ChangeTypes` is called a `ChangeSet`.

There are four actions that allow you to work with change sets:

- `StartChangeSet` – Requests a set of changes. The changes are added to a queue and processed. For more information, see [Working with seller products](#) and [Working with a private marketplace](#).

- **DescribeChangeSet** – Gets the details of a set of changes, including the status of the request. The statuses include:
 - **PREPARING** – Getting ready to apply the changes.
 - **APPLYING** – In the process of making the requested changes.
 - **SUCCEEDED** – Request was completed successfully.
 - **CANCELLED** – Request was canceled by the user.
 - **FAILED** – Request was completed unsuccessfully. Further details are available in the response.
- **ListChangeSets** – Gets a list of the change sets that are currently in process.
- **CancelChangeSet** – Requests a change set be canceled. Changes can only be canceled while in the **PREPARING** status.

A typical workflow is to request a change with **StartChangeSet**, and then use the returned **ChangeSetId** to poll the **DescribeChangeSet** action until the change is complete.

The following is an example of the **DescribeChangeSet** response.

```
{
  "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

When polling or working with change sets programmatically, you must adhere to the service limits. For more information, see [Service quotas for AWS Marketplace Catalog API](#).

After your change is complete, you can use `ListEntities` to find the entity that you created or modified (and its associated `EntityID`). You can then use `DescribeEntity` with the `EntityID` to get details about it.

For more information about working with change requests in the console for sellers, see [Creating a change request](#) in the *AWS Marketplace Seller Guide*.

Making multiple change requests simultaneously

Within a **single change set**, you can bundle all change types and they are run together. Catalog API is built to make multiple changes simultaneously to provide the best performance. Sellers and Channel Partners can invoke changes with multiple `ChangeTypes` bundled into a `ChangeSet`. You

can invoke multiple changes on single or different entities in the same ChangeSet. Catalog API evaluates which order the changes need to be applied and makes those changes.

However, if the requests are made as **separate change sets**, AWS Marketplace can't initiate conflicting change requests on the same product. In these cases, AWS Marketplace returns a `ResourceInUseException` error.

- For modifying AMI and container products, most changes can be made without error, with the following exceptions:
 - If two requests are the same `ChangeType` on the same product, the second request returns an error.
 - If one request is to update the version information, and the other request is to restrict or add a version, then the second request returns an error.
 - If a request is `PREPARING`, another request can be made on the same product. However, a change that is currently `APPLYING` may block other requests, returning an error.
- For other product types and private marketplaces, you can only have a single request for a product at a time. If a different request to update the same product is made while a first request is ongoing, the second returns an error.
- If there is a request for any product that is pending with the AWS Marketplace Seller Operations team, then any other requests on that product return an error.

If you receive a `ResourceInUseException` error for a change request, you can retry the request later. Depending on the state of the ongoing request, you can also cancel the first request, to allow the resubmitted second request to complete sooner.

Invoking multiple change types in one change set

You can use the Catalog API to combine and chain up to 20 changes in one `StartChangeSet` request targeting one or multiple different entities.

A typical use case is to create a `SaaSProduct@1.0` draft product, an `Offer@1.0` draft offer, and also filling in the metadata information of the product and offer. This is done by including the following four change types in one change set:

- `CreateProduct` on `SaaSProduct@1.0`

Specify the `ChangeName` parameter. Then, the product created in this change type can be referenced in the same change set by subsequent changes.

For example, `CreateProductChange`.

- `UpdateInformation` on the `SaaSProduct@1.0` created in the same change set

In the `Entity.Identifier` field, you can refer to the product created by `CreateProduct` change type using the change name in this format:

```
${ChangeName}.Entity.Identifier
```

For example, `$CreateProductChange.Entity.Identifier`.

- `CreateOffer` on `Offer@1.0` tied to the `SaaSProduct@1.0` created in the same change set

Specify the `ChangeName` parameter. Then, the product created in this change type can be referenced in the same change set by subsequent changes. For example, `CreateOfferChange`.

For the `ProductId` parameter in the payload of `CreateOffer` change type, you can also refer to the SaaS product created in `CreateProduct` change type by using `${ChangeName}.Entity.Identifier` syntax.

For example, `{"ProductId": "$CreateProductChange.Entity.Identifier"}`.

- `UpdateInformation` on the `Offer@1.0` created in the same change set

In the `Entity.Identifier` field, you can refer to the offer created by the `CreateOffer` change type using the change name in this format:

```
${ChangeName}.Entity.Identifier
```

For example, `$CreateOfferChange.Entity.Identifier`.

The following is an example of a combined change set.

```
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"
    }
  }
]
}

```

Working with the Details attribute (Legacy)

Note

This section describes the legacy Details attribute in your change request, which requires additional formatting for your change details. We recommend using the alternative DetailsDocument attribute. It doesn't require additional formatting and the change details don't need to be changed. For examples of the DetailsDocument attribute, see [Working with seller products](#) and [Working with a private marketplace](#).

The Details attribute of the StartChangeSet operation is a string value. Its contents are JSON objects. To put a JSON object into a string attribute, you must convert the object to a single-line string by escaping all JSON control characters, and removing line breaks.

For example, if you are using the StartChangeSet operation with UpdateProcurementPolicy to disable requests from users in your private marketplace, make a request like the following.

```

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"
      }
    }
  ]
}

```

In this case, the JSON object that you use for the `Details` attribute looks like the following (before conversion to a string).

```

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

```

But the `Details` attribute requires a string, not JSON. After converting this JSON object to a single line string, it looks like the following.

```

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

```

With this string, you can create the full change set request, as follows.

```

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"
      }
    }
  ]
}

```

```
}  
]  
}
```

Generally, examples in this API reference show the JSON object already converted to a string. In some cases, more complicated samples with new lines are included to enhance understanding.

Automate converting JSON to a string

Converting a JSON object to a string can be automated using tools such as [jq](#), a lightweight command-line JSON processor. The following example shows using jq to convert a JSON object to a string that can be used in the Details attribute.

```
DETAILS_JSON='{  
  "ProductTitle": "My Product Title",  
  "ShortDescription": "My product short description.",  
  "LongDescription": "My product long description."  
}';  
  
DETAILS_JSON_STRING="$(echo "${DETAILS_JSON}" | jq 'tostring');";
```

If you echo "\${DETAILS_JSON_STRING}", the result is the following string with JSON properly escaped: {"ProductTitle": "My Product Title", "ShortDescription": "My product short description.", "LongDescription": "My product long description."}

Using DescribeEntity to get information about your entities

You can programmatically get information about your existing entities, including products and private marketplace, through the Catalog API.

The `ListEntities` action returns a list of entities. Then, you can use the `DescribeEntity` action to get details about an individual entity. This can be directly useful, for example, to catalog the products you sell. It can also be useful when updating entities, because you can get the current state of the entity before updating just the parts that you want to update.

The following example shows using `ListEntities` to get a list of container products, and then using `DescribeEntity` to get information about one of the specific products.

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

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

For the entity type, you must use the entity type without the version. It returns all entities of that type (and doesn't filter on version).

Here is a sample of the response to the `ListEntities` action.

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

To get the details of one of these products, use the `DescribeEntity` action. The following example shows how to get details about the first product returned above.

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

The following shows the response to `DescribeEntity`.

```
{
  "EntityType": "ContainerProduct@1.0",
  "EntityIdentifier": "example1-abcd-1234-5ef6-7890abcdef12@9",
}
```

```

"EntityArn": "arn:aws:aws-marketplace:[exampleARN]",
"LastModifiedDate": "2021-03-02T20:19:14Z",
"Details": "{\\\"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\\\": {\\\"AWSservices\\\": [\\\"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\\\"}]]}",
"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-ccccccccccccc",
            "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

The `DetailsDocument` attribute contains the entity details as a JSON object. The legacy `Details` attribute contains the same JSON object as a string.

Using the AWS Marketplace Agreement API

AWS Marketplace is a curated digital catalog that customers can use to find, buy, deploy, and manage third-party software, data, and services to build solutions and run their businesses. The AWS Marketplace Agreement Service provides an API interface that helps AWS Marketplace buyers and sellers manage their agreements, including listing, searching, and filtering agreements.

Key concepts

The following table describes key concepts used in the AWS Marketplace Agreement API.

Concept	Description
Acceptor	The party that accepts all or a subset of the terms extended by the proposer in an agreement. For most common use cases, acceptors are the buyers of the product.
Agreement	A document that binds two parties, including the proposer (commonly , the seller/ISV or channel partner) and the acceptor (commonly, the buyer) and defines the terms and conditions applicable between them.
Agreement Cancellation Request	A request initiated by the seller (proposer) to cancel an active agreement. The buyer (acceptor) can accept or reject the cancellation request. If accepted, the agreement cancellation workflow executes asynchronously and the agreement status changes to cancelled. If rejected, the agreement remains active and the seller can submit a new cancellation request.
Agreement Proposal	Entity referring to the set of terms/offer proposed by the seller/proposer.
Agreement Request	Entity referring to resource created as a result of customer action, outlining the agreement creation/modification parameters, request lifecycle and estimated outcome upon acceptance of the request.
Billable dimension	Elements related to how the product is priced. For example, Amazon Elastic Compute Cloud (Amazon EC2) instances for Amazon Machine Image (AMI) products, or premium compared to read-only users for software as a service (SaaS) products.
Channel Partners (CP)	Organizations that are authorized by the product owner (ISV) to resell their products on their behalf on AWS Marketplace.
Independent Software Vendors (ISVs)	Sellers that list and manage their own products in AWS Marketplace.

Concept	Description
Intent	Intent represents the motivation the buyer has while creating the Agreement. Currently supported intents are New, Amend, and Replace during agreement creation flow.
Payment Request	A request initiated by the seller (proposer) for payment associated with an agreement. The buyer (acceptor) can accept or reject the payment request.
Proposer	The party that extends the initial set of terms in an agreement, most commonly by using an offer.
Purchase Orders	Organizations require buyers to use purchase orders to buy from approved suppliers, track what they buy, and ensure spending is forecasted, budgeted, approved, and assigned to the responsible cost center(s).
Resource	A unit or resource that sellers intend to sell in AWS Marketplace, often referred to as a base product. A base product is not complete for buyer consumption until product information, deployment attributes, and billing information are added. A product describes the product information, software deployment attributes, and billing mechanism of the listing that a seller intends to sell. The product must be paired with an offer to become a transactable unit that can be sold and be used by buyers in AWS Marketplace.
Term	A unit within an agreement that governs how an agreement is enforced (for example, pricing).

Endpoint

You can access the AWS Marketplace Agreement Service with the following endpoints:

Endpoint	Region	Supported IP Protocols
agreement-marketplace.us-east-1.api.aws	US East (N. Virginia)	IPv4, IPv6

Permissions

The Agreements API uses standard AWS Signature Version 4 (SigV4) authentication. You must have valid AWS credentials and the appropriate IAM permissions to call the API. For details, see [Access control for the AWS Marketplace Agreement API](#).

Error codes

The following error codes apply to [AWS Marketplace Agreement Service actions](#). A single error code may be returned for multiple use cases. Refer to the descriptions below to identify the specific scenario in which each error code occurs. This list is not exhaustive and may be updated as new features are released.

ValidationException

Error code	Description
INVALID_SOURCE_AGREEMENT_IDENTIFIER	Returned when the source agreement identifier does not match the expected format, or when a source agreement identifier is provided for an intent that does not require one (such as NEW).
MISSING_SOURCE_AGREEMENT_IDENTIFIER	Returned when the source agreement identifier is not provided for an intent that requires one (such as REPLACE or AMEND).
INVALID_REQUESTED_TERM_CONFIGURATION	Returned when the requested term configuration is not valid. Possible scenarios: • A configuration is provided for a term that does not accept one. • A required configuration is missing.

Error code	Description
	• The selector value does not belong to the agreement proposal. • The dimensions can only contain one dimension when multiple dimension selection is not allowed. • A dimension key is not recognized in the agreement proposal or accepted terms. • A dimension value can only be 1 when quantity configuration is not enabled for the dimension. • The configuration contains duplicate dimension keys. • The configuration does not differ from the source agreement .
INVALID_AGREEMENT_PROPOSAL_IDENTIFIER	Returned when the agreement proposal identifier does not match the expected format, or when an agreement proposal identifier is provided for an intent that does not require one (such as AMEND).
MISSING_AGREEMENT_PROPOSAL_IDENTIFIER	Returned when the agreement proposal identifier is not provided for an intent that requires one (such as NEW or REPLACE).
INVALID_FILTER_NAME	Returned when the provided filter name is not a recognized filter, or when duplicate filter names are included in the request.
INVALID_FILTER_VALUES	Returned when filter values are invalid, duplicated, empty, exceed the maximum allowed count, or when a date range filter specifies a BeforeEndTime value that precedes the AfterEndTime value.
INVALID_NEXT_TOKEN	Returned when the pagination token is malformed, expired, cannot be decrypted, or when request parameters have changed between paginated calls.

Error code	Description
DUPLICATE_CHARGES	Returned when the request contains duplicate charge identifiers, or when different purchase order values are provided for the same charge identifier.
UNSUPPORTED_FILTERS	Returned when the provided combination of filters is not supported. Supported filter combinations vary based on party type. Check the supported combinations in the public documentation.
INVALID_PURCHASE_ORDER_REFERENCE	Returned when the purchase order reference does not meet format requirements, or when a required purchase order is not provided as configured by the account administrator.
INVALID_CHARGE_AMOUNT	Returned when the charge amount is not valid. Possible scenarios: • The charge amount exceeds the maximum total charge amount for the term. • The charge amount format does not match the currency requirements (for example, too many decimal places). • No remaining amount is available because the agreement has reached the maximum total charge amount.

Error code	Description
UNSUPPORTED_ACTION	Returned when the requested operation is not supported for the current state of the resource. The specific scenarios vary by API: <code>UpdatePurchaseOrders</code> – The charge date is in the past and auto renewal is disabled, the charge date is within 3 hours, the charge is a refund charge, the charge revision is not the latest, the purchase orders do not all belong to the same agreement, or the agreement is not active. <code>CreateAgreementRequest</code> – Amending a future-dated agreement is not supported, the resource type and term type combination does not support the requested intent, replacing an active agreement with a future-dated agreement is not allowed, the agreement duration cannot be changed during amendment, the agreement cannot be updated at this time (for example, pending entitlements exist), the agreement proposal must refer to the same resource as the source agreement, the agreement proposal must differ from the source agreement, or the agreement type does not support this action. <code>AcceptAgreementRequest</code> – The resource type and term type combination does not support this action, the agreement type does not support this action, an active agreement already exists on the same resource, or the agreement cannot be updated at this time. <code>CancelAgreement</code> – The agreement has pending entitlements, the agreement type does not support cancellation, an agreement with <code>UsageBasedPricingTerm</code> cannot be cancelled while an agreement with <code>ConfigurableUpfrontPricingTerm</code> or <code>PaymentScheduleTerm</code> is still active on the same resource, or the resource type and term type combination does not support cancellation.

Error code	Description
	<code>GetBillingAdjustmentRequest</code> – The agreement is a non-purchase agreement type, or the caller is the acceptor of the agreement. <code>SendAgreementCancellationRequest</code> – The agreement type does not support this action. <code>AcceptAgreementCancellationRequest</code> – The agreement type does not support this action. <code>RejectAgreementCancellationRequest</code> – The agreement type does not support this action. <code>CancelAgreementCancellationRequest</code> – The agreement type does not support this action. <code>GetAgreementCancellationRequest</code> – The agreement is a non-purchase agreement type.

ResourceNotFoundException

The `ResourceNotFoundException` includes `resourceType` and `resourceId` fields that identify the resource that could not be found. The following table describes the scenarios that can cause this exception for each resource type.

Note

For security reasons, the same error is returned whether a resource does not exist or the caller does not have access to it. This prevents enumeration of valid resource identifiers.

Resource type	Possible scenarios
<code>AgreementProposal</code>	- No agreement proposal found for the given identifier. - The offer is not targeted to the acceptor. - The offer is not available in the acceptor's country.
<code>Agreement</code>	No agreement found for the given identifier.

Resource type	Possible scenarios
	The caller is neither the acceptor nor the proposer of the agreement.
AgreementRequest	- No agreement request found for the given identifier. - The agreement request belongs to another caller. - The agreement request has expired. - The agreement request has been superseded by a subsequent accepted request on the same product or offer. - The agreement proposal became inactive between <code>CreateAgreementRequest</code> and <code>AcceptAgreementRequest</code> calls due to offer republishing or offer expiration.
Charge	- No charge found for the given identifier. - The charge does not belong to the specified agreement.
BillingAdjustmentRequest	- No billing adjustment request found for the given identifier. - The caller is neither the acceptor nor the proposer of the agreement associated with the adjustment.
AgreementCancellationRequest	- No agreement cancellation request found for the given identifier. - The caller is neither the acceptor nor the proposer of the agreement associated with the cancellation request.
PaymentRequest	- No payment request found for the given identifier. - The caller is neither the acceptor nor the proposer of the agreement associated with the payment request.

Using the AWS Marketplace Metering API

This reference provides descriptions of the AWS Marketplace Metering Service API. AWS Marketplace sellers can use this API to submit data for custom usage dimensions. For more

information about the necessary permissions to use this API, see [AWS Marketplace metering and entitlement API permissions](#) in the *AWS Marketplace Seller Guide*.

Submitting metering records

[*MeterUsage*](#)

- Submits the metering record for an AWS Marketplace product
- Called from: Amazon Elastic Compute Cloud (Amazon EC2) instance or a container running on either Amazon Elastic Kubernetes Service (Amazon EKS) or Amazon Elastic Container Service (Amazon ECS)
- Supported product types: Amazon Machine Images (AMIs) and containers
- Vendor-metered tagging: supported allocation tagging

[*BatchMeterUsage*](#)

- Submits the metering record for a set of customers. AWS CloudTrail captures BatchMeterUsage API calls. Use CloudTrail to verify that the software as a subscription (SaaS) metering records that you sent are accurate by searching for records using the eventName of BatchMeterUsage. You can also use CloudTrail to audit records over time. For more information, see [CloudTrail concepts](#).
- Called from: SaaS applications
- Supported product type: SaaS
- Vendor-metered tagging: supports allocation tagging

Accepting new customers

[*ResolveCustomer*](#)

- Resolves the registration token that the buyer submits through the browser during the registration process. Obtains a CustomerIdentifier along with the CustomerAWSAccountId and ProductCode.
- Called from: SaaS application during the registration process
- Supported product type: SaaS
- Vendor-metered tagging: not applicable

Entitlement and metering for paid container products

RegisteredUsage

- Provides software entitlement and metering. Paid container software products sold through AWS Marketplace must integrate with the AWS Marketplace Metering Service and call the RegisterUsage operation. Free and Bring Your Own License model (BYOL) products for Amazon ECS or Amazon EKS aren't required to call RegisterUsage. However, you can do so if you want to receive usage data in your seller reports. For more information about using the RegisterUsage operation, see [Container-based products on AWS Marketplace](#).
- Called from: paid container software products
- Supported product type: containers
- Vendor-metered tagging: not applicable

Using the AWS Marketplace Entitlement API

The AWS Marketplace Entitlement Service is used to determine the entitlement of a customer to a given product. An entitlement represents capacity in a product owned by the customer. For example, a customer might own some number of users or seats in an SaaS application or some amount of data capacity in a multi-tenant database.

Using the AWS Marketplace Deployment API

The AWS Marketplace Deployment Service supports Quick Launch, which is a deployment option for software as a service (SaaS) products. Quick Launch can help you configure, deploy, and launch products. The AWS Marketplace Deployment Service provides sellers with a secure method for passing deployment parameters (for example, API keys and external IDs) to buyers.

Note

For information about Quick Launch for sellers, see [Configure Quick Launch](#).

Using the AWS Marketplace Discovery API

The AWS Marketplace Discovery API provides programmatic access to the AWS Marketplace catalog. You can use it to retrieve product and pricing information, build integrated procurement experiences, and create custom storefronts.

Service endpoint

The Discovery API uses the following endpoint format:

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

For example, to call the API in US East (N. Virginia):

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

API version

The current API version is 2026-02-05.

Data model

The Discovery API organizes the AWS Marketplace catalog into the following entities:

- **Listing** — A product or multi-product solution as it appears to buyers. A listing includes descriptions, highlights, categories, badges, pricing models, pricing units, reviews, promotional media, seller engagements, fulfillment option types, and references to associated products and offers. Use `GetListing` to retrieve a listing, or `SearchListings` to search across listings.
- **Product** — The underlying software or service being sold. A product includes descriptions, highlights, categories, promotional media, seller engagements, and fulfillment options that describe how a buyer can deploy or access the product (such as AMI, SaaS, Container, or Helm). Use `GetProduct` to retrieve product details and `ListFulfillmentOptions` to retrieve detailed fulfillment options for a product.
- **Offer** — A pricing arrangement for a product, including the pricing model, seller of record, availability dates, and badges. An offer contains commercial terms such as usage-based pricing, fixed upfront pricing, free trial periods, legal documents, payment schedules, and renewal terms. Use `ListPurchaseOptions` to find all available offers for a product, `GetOffer` to retrieve the details of an offer, and `GetOfferTerms` to retrieve the specific terms of the offer.
- **Offer set** — A grouped collection of private offers for each product in a multi-product solution. An offer set lets buyers review all offers together and accept them simultaneously with a single action. Use `ListPurchaseOptions` to find all available offer sets for a product, `GetOfferSet` to retrieve the details of an offer set, `GetOffer` to retrieve the details of an offer, and `GetOfferTerms` to retrieve the specific terms of the offer.

Authentication

The Discovery API uses standard AWS Signature Version 4 (SigV4) authentication. You must have valid AWS credentials and the appropriate IAM permissions to call the API. For details, see [Access control for the AWS Marketplace Discovery API](#).

Making requests

All Discovery API operations use the HTTP POST method with a JSON request body. The operation name is specified in the URL path.

Response format

All responses are returned in JSON format. Successful responses return HTTP status code 200. Error responses include an error type and message. For details, see [Common Errors](#).

Using the AWS SDK

The recommended way to call the Discovery API is through the AWS SDK. The SDK handles authentication, request signing, serialization, and error handling automatically.

```
# 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);
```

Pagination

Operations that return lists (such as `ListPurchaseOptions` and `SearchFacets`) support pagination using `nextToken`. If the response includes a `nextToken` value, pass it in the next request to retrieve additional results.

Throttling

The Discovery API enforces request rate limits to ensure service availability. If you exceed the rate limit, the API returns a `ThrottlingException` (HTTP 429). Implement exponential backoff and retry logic in your application.

Supported AWS Regions for the AWS Marketplace API

The following sections provide information about supported AWS Regions.

Topics

- [Supported AWS Regions for the AWS Marketplace Catalog API](#)
- [Supported AWS Regions for the AWS Marketplace Agreement API](#)
- [Supported AWS Regions for the AWS Marketplace Metering API](#)
- [Supported AWS Regions for the AWS Marketplace Entitlement API](#)
- [Supported AWS Regions for the AWS Marketplace Deployment API](#)
- [Supported AWS Regions for the AWS Marketplace Discovery API](#)

Supported AWS Regions for the AWS Marketplace Catalog API

You can access the AWS Marketplace Catalog API from the US East (N. Virginia) AWS Region with the following endpoint.

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

Supported AWS Regions for the AWS Marketplace Agreement API

You can access the AWS Marketplace Agreement Service from the US East (N. Virginia) AWS Region with the following endpoint.

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

Supported AWS Regions for the AWS Marketplace Metering API

BatchMeterUsage Region support

BatchMeterUsage is supported in the following AWS Regions

Commercial Regions

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-southeast-2, il-central-1, ca-central-1, eu-south-2, eu-central-2

China Regions

cn-northwest-1

MeterUsage Region support for Amazon EC2

MeterUsage is supported in the following AWS Regions

Commercial Regions

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-southeast-2, il-central-1, ca-central-1, eu-south-2, eu-central-2

AWS GovCloud (US) Regions

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

Note

To add AWS Regions for metering, contact [AWS Marketplace Seller Operations](#).

MeterUsage Region support for Amazon ECS and Amazon EKS

MeterUsage is supported for Amazon ECS and Amazon EKS in the following AWS Regions

Commercial Regions

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

MeterUsage is unavailable in AWS GovCloud (US) for Amazon ECS and Amazon EKS.

MeterUsage Region support for Amazon Bedrock AgentCore Runtime

MeterUsage is supported for Amazon Bedrock AgentCore Runtime in the following AWS Regions

Commercial Regions

us-east-1

Note

MeterUsage is unavailable in AWS GovCloud (US) for Amazon Bedrock AgentCore Runtime.

RegisterUsage Region support

RegisterUsage is supported in the following AWS Regions

Commercial Regions

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 Region support

ResolveCustomer is supported in the following AWS Regions

Commercial Regions

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-southeast-2, il-central-1, ca-central-1, eu-south-2, eu-central-2

China Regions

cn-northwest-1

Supported AWS Regions for the AWS Marketplace Entitlement API

You can access the AWS Marketplace Entitlement Service from the US East (N. Virginia) AWS Region with the following endpoints.

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

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

Supported AWS Regions for the AWS Marketplace Deployment API

You can access the AWS Marketplace Deployment Service from the US East (N. Virginia) AWS Region with the following endpoint.

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

Supported AWS Regions for the AWS Marketplace Discovery API

The AWS Marketplace Discovery API is available in the following AWS Regions:

- US East (N. Virginia) — us-east-1
- US West (Oregon) — us-west-2
- Europe (Ireland) — eu-west-1

Permissions for the AWS Marketplace API

The following sections provide information about permissions.

Topics

- [Access control for the AWS Marketplace Catalog API](#)
- [Access control for the AWS Marketplace Agreement API](#)

- [Access control for the AWS Marketplace Metering API](#)
- [Access control for the AWS Marketplace Entitlement API](#)
- [Access control for the AWS Marketplace Deployment API](#)
- [Access control for the AWS Marketplace Compliance API](#)
- [Access control for the AWS Marketplace Discovery API](#)
- [Access control for the AWS Marketplace Reporting API](#)

Access control for the AWS Marketplace Catalog API

You can use the AWS Marketplace Catalog API to manage [a seller product in AWS Marketplace](#) or an [experience in a private marketplace](#). However, first make sure your user or role can access the API functionality that you want to call.

Use AWS Identity and Access Management (IAM) to create users and roles and assign policies that grant limited permissions to end users. The policies define the actions that the user or role can take on your resources through the AWS Marketplace Catalog API.

For example, you can define roles such as engineering, marketing, and pricing. Then, you can add a user in your organization to the engineering role. In that role, they might be granted permissions to initiate a change request to publish a new version of your seller product. However, the engineering role doesn't allow the user to list all change sets.

Note

To sell products on AWS Marketplace, your AWS account must be set up as a seller account. For more details about becoming an AWS Marketplace seller, see [Getting started as a seller](#) in the *AWS Marketplace Seller Guide*.

You can use AWS managed policies, or you can create your own IAM policies to have more granular control than what's available in AWS managed policies. For details about these approaches, see the following topics.

Topics

- [Allowing actions with AWS managed policies](#)
- [Allowing actions on all resources](#)

- [Allowing actions on specific resources](#)
- [Allowing actions with specific ChangeType condition key](#)
- [Allowing actions with specific aws:ResourceTag condition key](#)
- [Creating a custom IAM role](#)
- [Managing tags on resources](#)
- [Managing tags when requesting changes to resources](#)
- [Granting permission to manage tags on resources](#)
- [Granting permission to manage tags on resources only when those resources have specific tags](#)
- [Granting permission to create entities and change sets only with tags](#)

Allowing actions with AWS managed policies

You can use policies that are managed by AWS to grant permissions to your user or role.

To work with products that you sell on AWS Marketplace, you can use the `AWSMarketplaceSellerFullAccess` IAM managed policy, which has full access to the AWS Marketplace Catalog API in addition to its other permissions. You can grant read-only access for the Catalog API with the `AWSMarketplaceSellerProductsReadOnly` policy. For more information, see [Controlling access to AWS Marketplace Management Portal](#), [Policies and permissions for AWS Marketplace sellers](#), and [AWS managed policies for AWS Marketplace sellers](#) in the *AWS Marketplace Seller Guide*.

To manage a private marketplace, you can use the `AWSPprivateMarketplaceAdminFullAccess` IAM managed policy, which has full access to create and edit the private marketplace for your account or AWS organization. For more information, see [Controlling access to AWS Marketplace subscriptions](#), [Creating a private marketplace administrator](#), and [AWS managed policies for AWS Marketplace buyers](#) in the *AWS Marketplace Buyer Guide*.

Alternatively, you can create your own IAM policies to have more granular control than is available in AWS managed policies. Use the following topics to create your own IAM policies.

Allowing actions on all resources

Resources are objects that the actions can act upon. Not every resource type can be specified with every action. Some resource types work with only certain actions. For more information, see [Actions, resources, and condition keys for the AWS Marketplace Catalog](#) in the *Service Authorization Reference*.

There are two resource types in the Catalog API:

- **Entity** – An entity is a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).
- **ChangeSet** – A change set is returned each time you use Catalog API to make changes to an entity. The change set describes the requested changes and its status. A change set can be canceled if the status is in the PREPARING state.

To allow a user or role the permission to make changes to all entities in an AWS account, you can add the following IAM policy. With this policy, the user or role can use the `StartChangeSet` action on all resources (`"*"`).

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

For information about all actions available for the Catalog API, see [Actions, resources, and condition keys for AWS Marketplace Catalog](#) in the *Service Authorization Reference*.

Allowing actions on specific resources

Note

Resource-level permissions and condition context keys for the `StartChangeSet` action are supported only when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

Instead of allowing changes to all resources, you can use resource-level permissions to allow changes to specific resources.

For example, you can allow changes to a specific seller product in the AWS account instead of to all seller products. You do this by specifying the Amazon Resource Name (ARN) of the seller product in the Resource of the IAM policy.

 Note

To specify granular, resource-level permissions with actions that create new change sets, you need to also include a ChangeSet ARN to the list of resources. The ChangeSet ARN must include the wildcard (/*) to match any new change set ID that's created as shown.

```
{
  "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/*"
      ]
    }
  ]
}
```

For another example, you can allow changes to a specific experience in a private marketplace instead of to all experiences. You do this by specifying the ARN of the experience in the Resource of the IAM policy.

 Note

To specify granular, resource-level permissions with actions that create new change sets, you need to also include a ChangeSet ARN to the list of resources. The ChangeSet ARN must include the wildcard (/*) to match any new change set ID that's created as shown.

```
{
```

```
"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/*"
    ]
  }
]
```

Allowing actions with specific ChangeType condition key

Note

Resource-level permissions and condition context keys for the StartChangeSet action are supported only when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

The Catalog API action StartChangeSet has several different change types. You can allow access to only specific change types.

For example, you might only want to allow changes to the metadata of the seller product, such as the product title, and not allow adding new product versions. In this example, the change type UpdateInformation allows changing the metadata of a seller product, including the title. For more information about the different change types, see [Working with seller products](#) and [Working with a private marketplace](#) in the *AWS Marketplace Catalog API Reference*.

To limit the action to one or multiple change types, specify the ChangeType in the condition keys. In the following example IAM policy, the condition operator StringEquals specifies that the action is only allowed if the ChangeType matches UpdateInformation. For more information about condition keys, see [Condition operators](#) in the *AWS Identity and Access Management User Guide*.

Note

To specify granular, resource-level permissions with actions that create new change sets, you need to also include a ChangeSet ARN to the list of resources. The ChangeSet ARN must include the wildcard (/*) to match any new change set ID that's created as shown.

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

Allowing actions with specific `aws:ResourceTag` condition key

Note

Resource-level permissions and condition context keys for the `StartChangeSet` action are supported only when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

You can allow actions on a group of entities without having to keep updating the policy and specifying a possibly growing list of entity ARNs. You can do this with resource tagging. Adding tags to resources allows you to control access to those resources based on their tags. For example,

you might want to allow describing a group of seller products without specifying individual ARNs for each seller product.

For example, the following IAM policy allows the `DescribeEntity` action on any entity resource ("*") that has a tag key of `product-team` and tag value of `team-xyz`.

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

You can also allow describing and canceling change sets that were created with specific tags.

For example, the following IAM policy allows the `DescribeChangeSet` and `CancelChangeSet` actions on any change set resource ("*") that has a tag key of `product-team` and tag value of `team-xyz`.

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

```
    }
  ]
}
```

Also, you can allow starting change sets on entities only when those entities have specific tags. For example, you can allow changes to seller products with specific tags.

For example, the following IAM policy allows the `StartChangeSet` action on any entity resource ("*") that has a tag key of `product-team` and tag value of `team-xyz`. In addition, the `TagResource` action is required so that when the change set is created, it's tagged with the same tag key and value.

Note

If your policy to allow the `StartChangeSet` action includes a condition to match against specific tags, the same policy must also include the `TagResource` action. This is because the policy condition must match both the tag on the entity and the tag on the newly created change set resulting from the change request. Thus, it requires the user or role to also have the permission to tag the newly created change set. For an example of starting a change set and tagging the change set, see [the section called "Example: Adding tags to an entity and change set during creation"](#).

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

Creating a custom IAM role

Customers who want to use a Resale Authorization ChangeType or a CPPO ChangeType need to create a custom AWS Identity and Access Management (IAM) role. This will support the creation of the Resale Authorization product lifecycle.

To create a custom IAM role

1. Sign in to the IAM console (<https://console.aws.amazon.com/iam/>).
2. Under **Access management**, choose **Policies**.
3. Choose **Create policy**.
4. For **Step 1: Specify permissions**,
 - a. In the **Policy editor**, select the **JSON** button, and then add the following policy:

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. Choose **Next**.
5. For **Step 2: Review and create**,
 - a. For **Policy details**, enter **FullResaleAuthorizationAccess** under **Policy name** and enter an optional **Description**.
 - b. Review the **Permissions defined in this policy**.
 - c. For **Add tags**, add tags (optional).
 - d. Choose **Create policy**.

You have created the FullResaleAuthorizationAccess policy.

6. Under **Access management**, choose **Roles**.
7. Choose **Create role**.
8. For **Step 1: Select trusted entity**,
 - a. For **Trusted entity type**, choose **Custom trust policy**.

- b. Copy and paste the following custom trust policy into the JSON editor.

JSON

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

- c. Choose **Next**.
9. For **Step 2: Add permissions**,
 - a. Enter **FullResaleAuthorizationAccess** in the search bar.
 - b. Select the **FullResaleAuthorizationAccess** permission policy and then choose **Next**.
 10. For **Step 3: Name, review, and create**,
 1. For **Role details**, enter **FullResaleAuthorizationAccess** as the **Role name** and enter an optional **Description**.
 2. Under **Step 1: Select trusted entities**, ensure that the policy name you choose is attached to the role.
 3. Under **Step 2: Add permissions**, review the **Policy name**.
 4. Under **Step 3: Add tags**, add tags (optional).
 5. Choose **Create role**.

You have created the FullResaleAuthorizationAccess role.

Managing tags on resources

You can add, list, and remove tags from existing entities or change sets.

Add tags to resources

To add tags to an entity or change set, use the TagResource API action.

Request

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

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

Request parameters include:

- **Catalog (String) – (Required)** Must be AWSMarketplace.
- **ResourceArn (String) – (Required)** ARN of the change set or entity. A change set describes changes you make with Catalog API. An entity can be a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).
- **Tags (Array of objects) – (Required)** A list of objects specifying each tag key and value. Number of objects allowed: 1–50.
 - **Key (String) – (Required)** Name of the tag.
 - **Regex pattern** – `^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`
 - **Character length** – 1–128
 - **Value (String) – (Required)** Value of the tag.
 - **Regex pattern** – `^([\p{L}\p{Z}\p{N}_.:/+\\-@]*)$`
 - **Character length** – 0–256

Response

```
{}
```

Remove tags from resources

To remove a tag or list of tags from an entity or change set, use the `UntagResource` API action.

Request

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

```
{
  "Catalog": "AWSMarketplace",
  "ResourceArn": "string",
  "TagKeys": [
    "string"
    ...
  ]
}
```

Request parameters include:

- **Catalog (String) – (Required)** Must be `AWSMarketplace`.
- **ResourceArn (String) – (Required)** ARN of the change set or entity. A change set describes changes you make with Catalog API. An entity can be a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).
- **Tags (Array of objects) – (Required)** A list of key names of tags to be removed. Number of strings allowed: 0–256.

Response

```
{}
```

List all tags on a resource

To list all tags that have been added to and not yet removed from a change set or entity, use the `ListTagsForResource` API action.

Request

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

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

Request parameters include:

- **Catalog (String)** – (Required) Must be AWSMarketplace.
- **ResourceArn (String)** - (Required) ARN of the change set or entity. A change set describes changes you make with Catalog API. An entity can be a [seller product in AWS Marketplace](#) or an [experience in a private marketplace](#).

Response

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

Managing tags when requesting changes to resources

You can add tags when entities or change sets are created.

Example: Adding tags to a change set when creating a change set

The following is an example of a `StartChangeSet` request that updates the product metadata for a seller product. This request adds a tag to the change set that's created with this request by including the tag in the `ChangeSetTags` property.

```
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"
    }
  ]
}
```

For more information about managing seller products, see [Working with seller products](#) in the *AWS Marketplace Catalog API Reference*.

Example: Adding tags to an entity and change set during creation

The following is an example of a `StartChangeSet` request that creates a private marketplace experience entity. The request adds tags to both the entity resource and change set resource created with this request by including the tags in the `EntityTags` and `ChangeSetTags` properties. With these tags, the permission policy of a user or role can be specified to only allow describing or canceling the change set this request creates or only allow creating further change sets on the entity this request creates. For more information, see [Granting permission to create entities and change sets only with tags](#).

```
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"
    }
  ]
}
```

For more information about managing a private marketplace, see [Working with a private marketplace](#) in the *AWS Marketplace Catalog API Reference*.

Granting permission to manage tags on resources

To allow a user or role to add, remove, and list tags on all entities or change sets, they need the following IAM policy.

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

Granting permission to manage tags on resources only when those resources have specific tags

You can allow a user or role to add, remove, and list tags on entities or change sets that have specific tags. The following IAM policy allows those actions on any entity resource ("*") that has a tag key of `product-team` and tag value of `team-xyz`.

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

Granting permission to create entities and change sets only with tags

Note

Resource-level permissions and condition context keys for the `StartChangeSet` action are supported only when used with Catalog API. They are not supported when used with the [AWS Marketplace Management Portal](#).

You can enforce tagging when entities or change sets are created. Add the following policy to allow the `StartChangeSet` and the `TagResource` actions, with a condition specifying the tag key matches `product-team` and the tag value matches `team-xyz`. This policy condition must match both the tag on the newly created entity and the tag on the newly created change set resulting

from the creation request. For an example of tagging an entity on creation, see [the section called “Example: Adding tags to an entity and change set during creation”](#).

For existing entities, this policy also enforces tagging change sets when requesting changes to those entities. This also requires that the existing entity has this existing tag. This is because the policy condition must match both the tag on the existing entity and the newly created change set resulting from the change request. For an example of adding tags to change requests, see [the section called “Example: Adding tags to a change set when creating a change set”](#).

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

Access control for the AWS Marketplace Agreement API

To manage agreements in AWS Marketplace using the Agreement Service, you must ensure that your AWS Identity and Access Management (IAM) policies and roles are set up. Users must have the following policies/permissions to allow them to carry out the actions:

- `CreateAgreementRequest` – Grants permission to users to start a new request to create a new agreement or update an existing agreement.
- `AcceptAgreementRequest` – Grants permission to users to accept a previously created request to materialize into an agreement. Note: While accepting the agreement request, if users provide purchase orders, they will need to have `UpdatePurchaseOrders` permission as well to perform the action.

- `GetAgreementEntitlements` – Grants permission to users to list the entitlements of an agreement.
- `CancelAgreement` – Grants permission to users to cancel an active agreement they participate in.
- `ListAgreementCharges` – Grants permission to users to list charges for an agreement.
- `UpdatePurchaseOrders` – Grants permission to users to update purchase orders for agreement charges.
- `SendAgreementCancellationRequest` – Grants permission to users to send a cancellation request.
- `AcceptAgreementCancellationRequest` – Grants permission to users to accept a cancellation request initiated by the seller for an agreement they participate in as acceptor. Note: Users also need `CancelAgreement` permission because approving the cancellation request leads to agreement cancellation.
- `RejectAgreementCancellationRequest` – Grants permission to users to reject a cancellation request initiated by the seller for an agreement they participate in as acceptor.
- `SendAgreementPaymentRequest` – Grants permission to users to send a payment request.
- `AcceptAgreementPaymentRequest` – Grants permission to users to accept a payment request initiated by the seller for an agreement they participate in as acceptor.
- `RejectAgreementPaymentRequest` – Grants permission to users to reject a payment request initiated by the seller for an agreement they participate in as acceptor.
- `GetAgreementCancellationRequest` – Grants permission to users to retrieve detailed information about a specific cancellation request for an agreement they participate in as acceptor.
- `ListAgreementCancellationRequests` – Grants permission to users to list cancellation requests for agreements they participate in as acceptor.
- `GetAgreementPaymentRequest` – Grants permission to users to retrieve detailed information about a specific payment request for an agreement they participate in as acceptor.
- `ListAgreementPaymentRequests` – Grants permission to users to list payment requests for agreements they participate in as acceptor.
- `DescribeAgreement` – Grants permission to users to obtain detailed metadata about any of their agreements.
- `GetAgreementTerms` – Grants permission to users to obtain details about the terms of an agreement.

- SearchAgreements – Grants permission to users to search through all their agreements.

Note

For more information about these permissions, see [Policies and permissions for AWS Marketplace sellers](#) in the *AWS Marketplace Seller Guide* and [Controlling access to AWS Marketplace subscriptions](#) in *AWS Marketplace Buyer Guide*.

Allowing actions with AWS managed policies

AWS Marketplace defines persona specific managed policies. You can use managed policies for your IAM identities which will by-default narrows down the scope of the policy for specified role. Using managed policy will allow you to follow best practices and receive policy updates as new features became available in Marketplace.

For more information about these permissions, see [Policies and permissions for AWS Marketplace sellers](#) in the *AWS Marketplace Seller Guide* and [Controlling access to AWS Marketplace subscriptions](#) in *AWS Marketplace Buyer Guide*.

You're not limited to the permissions in the AWS managed policies that are described here. You can use IAM to create policies with custom permissions and then add those policies to IAM roles. For more information, see [Managing IAM policies](#) and [Adding IAM identity permissions](#) in the *IAM User Guide*.

Allowing actions with customer managed policies

You can also define a customer managed policy to control access to Agreement service APIs. You can customize the policy with mix of effect, actions and supported conditions.

To access the AWS Marketplace Agreements API for managing the product subscriptions, you can create following policy to allow access.

```
{
  "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"
        }
    }
}
]
}

```

Customizing the policy with service defined condition keys

Using AgreementType condition key

Below statement limits the access to UpdatePurchaseOrders operation only for PurchaseAgreement agreement type.

```

{
    "Effect": "Allow",
    "Action": [
        "aws-marketplace:UpdatePurchaseOrders"
    ],

```



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

Using ProductId condition key

Below statement limits the access to `CreateAgreementRequest` and `AcceptAgreementRequest` to only specified AWS Marketplace product ids. You can include this statement to allow creating only pre-approved product ids, such as limiting to Bedrock foundational models.

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

Using PartyType condition key

Below statement grants the access to read details of any agreement if the user is proposer of the agreement.

```
{
  "Effect": "Allow",
  "Action": [
    "aws-marketplace:DescribeAgreement"
  ],
```

```
"Resource": "*",
"Condition": {
  "StringEquals": {
    "aws-marketplace:PartyType": "Proposer"
  }
}
```

Access control for the AWS Marketplace Metering API

To integrate the AWS Marketplace Metering API with AWS Marketplace, the service account must have a constrained AWS Identity and Access Management (IAM) policy. For more information, see [AWS Marketplace metering and entitlement API permissions](#).

Access control for the AWS Marketplace Entitlement API

To integrate the AWS Marketplace Entitlement API with AWS Marketplace, the service account must have a constrained AWS Identity and Access Management (IAM). For more information, see [AWS Marketplace metering and entitlement API permissions](#).

Access control for the AWS Marketplace Deployment API

To manage deployments in AWS Marketplace, you must ensure that you have the necessary AWS Identity and Access Management (IAM) roles and permissions.

Before calling the `PutDeploymentParameter` action, buyers must create the **AWSServiceRoleForMarketplaceDeployment** service-linked role. This provides AWS Marketplace with the permissions required to create, manage, and tag the necessary deployment parameter related resources in the buyer's account. Buyers create this role using prompts as they progress through the configuration process for any Quick Launch experience. For more information, see [Using roles to configure and launch products](#) in *AWS Marketplace Buyer Guide*.

To call `PutDeploymentParameter`, sellers must have IAM permissions for the following actions:

JSON

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

```
        "Action": [
            "aws-marketplace:PutDeploymentParameter",
            "aws-marketplace:TagResource"
        ],
        "Effect": "Allow",
        "Resource": "*"
    }
}
```

The `aws-marketplace:PutDeploymentParameter` action permits the user to call the `PutDeploymentParameter` API. The API also accepts an optional `tags` attribute. If the `tags` attribute is included in the request, the caller must also have permissions for `aws-marketplace:TagResource` on the relevant resource. For more information about creating users, see [Creating a user in your AWS account](#) in the *IAM User Guide*. For more information about creating and assigning policies, see [Changing permissions for an IAM user](#).

Access control for the AWS Marketplace Compliance API

You can use the AWS Marketplace Compliance API to manage seller compliance in AWS Marketplace. However, first make sure your user or role can access the API functionality that you want to call.

Use AWS Identity and Access Management (IAM) to create users and roles and assign policies that grant limited permissions to end users. The policies define the actions that the user or role can take on your resources through the AWS Marketplace Compliance API.

Note

To sell products on AWS Marketplace, your AWS account must be set up as a seller account. For more details about becoming an AWS Marketplace seller, see [Getting started as a seller](#) in the *AWS Marketplace Seller Guide*.

Topics

- [Allowing actions with AWS managed policies](#)
- [Allowing actions on all resources](#)
- [Allowing actions on specific resources](#)

- [Allowing actions with specific aws:ResourceTag condition key](#)
- [Managing tags on resources](#)
- [Granting permission to manage tags on resources](#)
- [Granting permission to manage tags on resources only when those resources have specific tags](#)
- [Requiring tags when starting invoice submission tasks](#)
- [Verification evidence operations](#)

Allowing actions with AWS managed policies

You can use policies that are managed by AWS to grant permissions to your user or role.

To work with invoice submissions on AWS Marketplace, you can use the `AWSMarketplaceSellerFullAccess` IAM managed policy, which includes full access to the AWS Marketplace Compliance API actions in addition to its other permissions. For more information, see [Policies and permissions for AWS Marketplace sellers](#) and [AWS managed policies for AWS Marketplace sellers](#) in the *AWS Marketplace Seller Guide*.

Alternatively, you can create your own IAM policies to have more granular control than is available in AWS managed policies. Use the following topics to create your own IAM policies.

Allowing actions on all resources

Resources are objects that the actions can act upon. The Compliance API has the following resource types:

- **InvoiceSubmissionTask** – An invoice submission task tracks the processing of a seller-submitted invoice in AWS Marketplace.
- **IssuedTaxInvoice** – A tax invoice that AWS Marketplace issued on behalf of a seller.
- **VerificationEvidence** – Contains verification data for a specific verification category and subject being verified.

To allow a user or role full access to invoice submission task operations, you can add the following IAM policy. With this policy, the user or role can use all invoice submission task actions on all resources ("*").

```
{
  "Version": "2012-10-17",
```

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

To allow a user or role full access to issued tax invoice operations, you can add the following IAM policy.

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

To allow a user or role full access to verification operations, you can add the following IAM policy.

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

For information about all actions available for the Compliance API, see [Actions, resources, and condition keys for AWS Marketplace Compliance](#) in the *Service Authorization Reference*.

Allowing actions on specific resources

You can use resource-level permissions to allow actions on a specific resource instead of all resources. You do this by specifying the Amazon Resource Name (ARN) of the resource in the Resource of the IAM policy.

The following example allows the `GetInvoiceSubmissionTask` action on a specific invoice submission task.

```

{
  "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"
      ]
    }
  ]
}
```

The following example allows the `GetIssuedTaxInvoice` action on a specific issued tax invoice.

```

{
  "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"
  ]
}
```

The following example allows the `GetVerificationEvidence` action on a specific verification evidence resource.

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

Allowing actions with specific `aws:ResourceTag` condition key

You can allow actions on resources based on their tags without having to specify individual ARNs. Adding tags to resources allows you to control access to those resources based on their tags.

For example, the following IAM policy allows the `GetInvoiceSubmissionTask` action on any invoice submission task resource ("`*`") that has a tag key of `product-team` and tag value of `team-xyz`.

```
{
```

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

Similarly, the following IAM policy allows the `GetIssuedTaxInvoice` action on any issued tax invoice resource ("`*`") that has a tag key of `Department` and tag value of `Tax`.

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

The following IAM policy allows the `GetVerificationEvidence` action on any verification evidence resource ("`*`") that has a tag key of `Department` and tag value of `Compliance`.

```
{
```



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

Managing tags on resources

You can add, list, and remove tags from existing Compliance API resources such as invoice submission tasks and issued tax invoices.

Add tags to resources

To add tags to a resource, use the TagResource API action.

Request

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

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

Request parameters include:

- **ResourceArn (String)** – (Required) ARN of the resource.
- **Tags (Array of objects)** – (Required) A list of objects specifying each tag key and value. Number of objects allowed: 1–50.
 - **Key (String)** – (Required) Name of the tag. Regex pattern: `^([\p{L}\p{Z}\p{N}_.:/+~\-\@]*)$`. Character length: 1–128.
 - **Value (String)** – (Required) Value of the tag. Regex pattern: `^([\p{L}\p{Z}\p{N}_.:/+~\-\@]*)$`. Character length: 0–256.

Remove tags from resources

To remove a tag or list of tags from a resource, use the `UntagResource` API action.

Request

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

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

Request parameters include:

- **ResourceArn (String)** – (Required) ARN of the resource.
- **TagKeys (Array of strings)** – (Required) A list of key names of tags to be removed.

List all tags on a resource

To list all tags on a resource, use the `ListTagsForResource` API action.

Request

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

```
{
```

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

Response

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

Granting permission to manage tags on resources

To allow a user or role to add, remove, and list tags on all Compliance API resources, they need the following IAM policy.

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

Granting permission to manage tags on resources only when those resources have specific tags

You can allow a user or role to add, remove, and list tags on Compliance API resources that have specific tags. The following IAM policy allows those actions on any resource ("*") that has a tag key of product-team and tag value of 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"
        }
      }
    }
  ]
}
```

Requiring tags when starting invoice submission tasks

You can enforce tagging when invoice submission tasks are created by using the `aws:RequestTag` and `aws:TagKeys` condition keys with the `StartInvoiceSubmissionTask` action.

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

```
    }  
  }  
]  
}
```

Verification evidence operations

The AWS Marketplace Compliance API provides operations for managing verification evidence and verification processes. You can use IAM policies to control access to these operations.

IAM actions for verification operations

The following IAM actions are available for verification operations. All actions use the `aws-marketplace` namespace.

Evidence management actions

- `aws-marketplace:CreateVerificationEvidence` – Create new verification evidence.
- `aws-marketplace:UpdateVerificationEvidence` – Update existing verification evidence.
- `aws-marketplace:GetVerificationEvidence` – Get details of verification evidence.
- `aws-marketplace:ListVerificationEvidence` – List verification evidence resources.

Verification process actions

- `aws-marketplace:StartVerification` – Submit evidence and enable data sharing.
- `aws-marketplace:GetVerification` – Get detailed verification status.
- `aws-marketplace:ListVerifications` – List all verification statuses.

Resource type

The **VerificationEvidence** resource contains verification data for a specific verification category and subject being verified. The ARN format for this resource is:

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

The following is an example ARN for business verification:

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

Using the VerificationType condition key

The `aws-marketplace:VerificationType` condition key filters verification process operations by type. This condition key applies to the following actions:

- `StartVerification`
- `GetVerification`
- `ListVerifications`

Valid values for this condition key are:

- `BusinessVerification`

Note

Evidence management operations (`CreateVerificationEvidence`, `UpdateVerificationEvidence`, `GetVerificationEvidence`, `ListVerificationEvidence`) do not use this condition key because the verification type is already encoded in the resource ARN.

Allowing actions with AWS managed policies

The `AWSMarketplaceSellerFullAccess` IAM managed policy includes all seven verification permissions in addition to its other permissions. For more information, see [AWS managed policies for AWS Marketplace sellers](#) in the *AWS Marketplace Seller Guide*.

Allowing full access to all verification operations

To allow a user or role full access to all verification operations, you can add the following IAM policy.

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

Allowing read-only access to business verification evidence

To allow a user or role read-only access to business verification evidence, you can use ARN-based filtering to restrict access to a specific verification type.

```

{
  "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/*"
      ]
    }
  ]
}
```

Restricting verification process operations by type

To restrict verification process operations to a specific verification type, use the `aws-marketplace:VerificationType` condition key. The following example allows verification process operations only for business verification.

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

Allowing actions on specific verification evidence resources

To allow actions on specific verification evidence resources, specify the ARN of the resource in the Resource element of the IAM policy. The following example allows all evidence management actions on a specific verification evidence resource.

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


Access control for the AWS Marketplace Discovery API

The Discovery API uses AWS Identity and Access Management (IAM) for authentication and authorization. Any AWS customer can call the Discovery API by configuring the appropriate IAM permissions.

Users must have the following permissions to call Discovery API operations:

- `GetListing` – Grants permission to retrieve information about a listing.
- `GetProduct` – Grants permission to retrieve information about a product.
- `GetOffer` – Grants permission to retrieve information about an offer.
- `GetOfferTerms` – Grants permission to retrieve terms for an offer.
- `GetOfferSet` – Grants permission to retrieve information about an offer set.
- `ListPurchaseOptions` – Grants permission to list purchase options available to the buyer.
- `ListFulfillmentOptions` – Grants permission to list fulfillment options for a product.
- `SearchListings` – Grants permission to search for product listings.
- `SearchFacets` – Grants permission to retrieve facet values for filtering listings.

IAM permissions

To call Discovery API operations, the IAM principal (user or role) must have the appropriate `aws-marketplace` permissions. The Discovery API supports resource-level permissions, so you can scope access to specific resource types in your IAM policies.

The following IAM policy grants access to all Discovery API operations with resource-level scoping:

```
{
  "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/*"
    ]
}

```

Replace `<catalog-name>` with the appropriate catalog identifier (for example, `AWSMarketplace`).

Resource ARN formats

The following table shows the resource types and their ARN formats used by Discovery API operations:

Resource type	ARN format	Used by
Product	arn:aws:aws-marketplace:::catalog/ <i>catalog-name</i> /product/ <i>product-id</i>	GetProduct, ListFulfillmentOptions
Listing	arn:aws:aws-marketplace:::catalog/ <i>catalog-name</i> /listing/ <i>listing-id</i>	GetListing, SearchListings, SearchFacets
Offer	arn:aws:aws-marketplace:::catalog/ <i>catalog-name</i> /offer/ <i>offer-id</i>	GetOffer, GetOfferTerms

Resource type	ARN format	Used by
Offer Set	arn:aws:aws-marketplace:::catalog/ <i>catalog-name</i> /offerSet / <i>offerSet-id</i>	GetOfferSet
Purchase Option	arn:aws:aws-marketplace:::catalog/ <i>catalog-name</i> /purchaseOption/ <i>purchaseOption-id</i>	ListPurchaseOptions

Scoping operations to specific resources

For Get operations (GetListing, GetProduct, GetOffer, GetOfferTerms, GetOfferSet) and the ListFulfillmentOptions operation, you can scope access down to specific resource IDs instead of using a wildcard. For example, to restrict access to a single listing:

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

This pattern works for any Get operation and ListFulfillmentOptions. Replace the resource type and ID accordingly:

- 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*

The SearchListings, SearchFacets, and ListPurchaseOptions operations do not support scoping to a single resource because they operate across multiple items. Use the following wildcard ARN patterns for these operations:

- For SearchListings and SearchFacets: `arn:aws:aws-marketplace:::catalog/catalog-name/listing/*`
- For ListPurchaseOptions: `arn:aws:aws-marketplace:::catalog/catalog-name/purchaseOption/*`

Service-linked roles

The Discovery API does not use service-linked roles. All access is controlled through standard IAM policies.

Cross-account access

You can grant cross-account access to the Discovery API using IAM roles. Create a role in the target account with the appropriate Discovery API permissions, then assume the role from the source account.

Access control for the AWS Marketplace Reporting API

You use the AWS Marketplace Reporting API to get the **Procurement insights** dashboard. The API supports the [GetBuyerDashboard](#) action.

To use the API, you must first create the `AWSServiceRoleForProcurementInsightsPolicy` service-linked role. The role does the following:

- Enables AWS Marketplace to access and describe the data for all the accounts in a buyer's organization.
- Gets the **Procurement insights** dashboard
- Enables you to register and deregister delegated administrators

Buyers create the role when they use the AWS Marketplace console to enable trusted access to the **Procurement insights** dashboard. For more information about that process, see [Activating the dashboard](#), in the *AWS Marketplace Buyer Guide*.

⚠ Important

- When using the API or the CLI, you must create the service-linked role before you enable trusted access to the dashboard.
- In addition to the service-linked role, you must enable all features for your organization, and you must belong to an administrator account. For more information, see the following topics in the *AWS Organizations User Guide*:
 - [Enabling all features for an organization with AWS Organizations](#)
 - [Tutorial: Creating and configuring an organization](#)
 - [Managing the management account with AWS Organizations](#)

The `AWSServiceRoleForProcurementInsightsPolicy` must have the following IAM permissions in order to call the [GetBuyerDashboard](#) action, and to register and deregister delegated administrators:

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": "*"
    }
]
}

```

For more information about creating policies, see [Policies and permissions in AWS Identity and Access Management](#), in the *IAM User Guide*.

For more information about the AWS Organizations actions that the policy uses, see the [AWS Organizations API reference](#).

Quotas for the AWS Marketplace API

The following sections provide information about service quotas.

Topics

- [Service quotas for AWS Marketplace Catalog API](#)
- [Service quotas for AWS Marketplace Agreement API](#)
- [Service quotas for AWS Marketplace Entitlement API](#)
- [Service quotas for AWS Marketplace Deployment API](#)
- [Service quotas for AWS Marketplace Metering API](#)
- [Service quotas for AWS Marketplace Discovery API](#)

Service quotas for AWS Marketplace Catalog API

The AWS Marketplace Catalog API has the following quotas.

Request quotas

API operations	Request rate (per AWS account)
ListEntities	10 per second
DescribeEntity	20 per second
StartChangeSet	5 per second
ListChangeSets	5 per second
DescribeChangeSet	10 per second
CancelChangeSet	5 per second

API operations	Request rate (per AWS account)
TagResource	5 per second
UntagResource	5 per second
ListTagsForResource	5 per second
PutResourcePolicy	5 per second
GetResourcePolicy	5 per second
DeleteResourcePolicy	5 per second

Product pricing dimension quotas

Description	Limit
Maximum number of unique contract dimensions (also known as Entitled dimensions) that can be included in a private offer or public offer. Contract dimensions in offers count towards the maximum number of unique contract dimensions per product.	200
Maximum number of unique contract dimensions (also known as Entitled dimensions) per product. This is inclusive of all unique contract dimensions in both private offers and public offers. Dimensions in the restricted state count towards this limit.	200
Maximum number of unique usage dimensions (also known as ExternallyMetered dimensions) that can be included in a private offer or public offer. These usage dimensions in offers count towards the maximum number of unique usage dimensions per product.	200

Description	Limit
Maximum number of unique usage dimensions (also known as <code>ExternallyMetered</code> dimensions) per product. This is inclusive of all unique usage dimensions in both private offers and public offers. Dimensions in the restricted state count towards this limit.	200

Account quotas

Quota	Description
Maximum number of open <code>StartChangeSet</code> requests per account	250
Maximum number of Offers created or updated concurrently per account	20
Maximum number of OfferSets created or updated concurrently per account	20

Request history retention quotas

Description	Quota
Retention period for change requests. This applies after the end time of each change request.	90 days

Note

Change requests and entities are different. Entities exist perpetually regardless of their type or state. For example, [seller product](#), [offer](#), [resale authorization](#), and [private marketplace experience](#) *entities* are never deleted even if they are in the draft state.

Service quotas for AWS Marketplace Agreement API

Your AWS account has the following quotas related to the AWS Marketplace Agreement Service.

Request quotas

API operation	Request rate (per AWS account)
AcceptAgreementCancellationRequest	2 per second
AcceptAgreementPaymentRequest	5 per second
AcceptAgreementRequest	3 per second
BatchCreateBillingAdjustmentRequest	2 per second
CancelAgreement	3 per second
CancelAgreementCancellationRequest	2 per second
CancelAgreementPaymentRequest	5 per second
CreateAgreementRequest	4 per second
DescribeAgreement	5 per second
GetAgreementCancellationRequest	5 per second
GetAgreementEntitlements	10 per second
GetAgreementPaymentRequest	5 per second
GetAgreementTerms	10 per second
GetBillingAdjustmentRequest	5 per second
ListAgreementCancellationRequests	5 per second

API operation	Request rate (per AWS account)
ListAgreementCharges	15 per second
ListAgreementInvoiceLineItems	10 per second
ListAgreementPaymentRequests	5 per second
ListBillingAdjustmentRequests	5 per second
RejectAgreementCancellationRequest	2 per second
RejectAgreementPaymentRequest	5 per second
SearchAgreements	5 per second
SendAgreementCancellationRequest	2 per second
SendAgreementPaymentRequest	5 per second
UpdatePurchaseOrders	3 per second

Agreement quotas

Description	Quota
The maximum agreement value (in USD) that you can specify for targeted proposal.	\$1B (or equivalent in requested currency)
The maximum agreement value (in USD) that you can specify for general proposal.	\$1M (or equivalent in requested currency)
The maximum number of free trials allowed per resource per account.	1
The maximum number of active agreements that you can create per resource per account.	100

Service quotas for AWS Marketplace Entitlement API

Your AWS account has the following quotas related to the AWS Marketplace Entitlement Service.

Request quotas

API action	Request rate (per AWS account)	Description
GetEntitlements	20 per second	The maximum number of GetEntitlements requests that you can make, per second, in this account in the current region.

Service quotas for AWS Marketplace Deployment API

Your AWS account has the following quotas related to the AWS Marketplace Deployment Service.

Request quotas

Quota	Default	Description
Deployment parameter	10	The maximum number of deployment parameters per buyer and product combination
Deployment parameter update frequency	100	The maximum number of times you can update a deployment parameter per 24 hours
PutDeploymentParameter throttle	1	The maximum number of PutDeploymentParameter requests that you can make per second

Quota	Default	Description
ListTagsForResource throttle	20	The maximum number of ListTagsForResource requests that you can make per second
TagResource throttle	20	The maximum number of TagResource requests that you can make per second
UntagResource throttle	20	The maximum number of UntagResource requests that you can make per second
Deployment parameter SecretString name length	15,000	The maximum number of characters for the deployment parameter SecretString string
Deployment parameter name length	400	The maximum number of characters in a deployment parameter name
TagList	50	The maximum number of tags per deployment parameter request
ClientToken	64	The maximum number of characters for the ClientToken string

Service quotas for AWS Marketplace Metering API

Your AWS account has the following quotas related to the AWS Marketplace Metering service.

Request quotas

API action	Request rate (per AWS account)	Description
BatchMeterUsage	10 per second	The maximum number of BatchMeterUsage requests that you can make, per second, in this account in the current region.
MeterUsage	10 per second	The maximum number of MeterUsage requests that you can make, per second, in this account in the current region.
RegisterUsage	5 per second	The maximum number of RegisterUsage requests that you can make, per second, in this account in the current region.

Service quotas for AWS Marketplace Discovery API

Your AWS account has the following quotas related to the AWS Marketplace Discovery API.

Request quotas

API operation	Request rate (per AWS account)
GetListing	5 per second
GetProduct	5 per second
GetOffer	5 per second
GetOfferTerms	5 per second

API operation	Request rate (per AWS account)
GetOfferSet	5 per second
ListPurchaseOptions	5 per second
ListFulfillmentOptions	5 per second
SearchListings	10 per second
SearchFacets	10 per second

Logging for the AWS Marketplace API

The following sections provide information about logging.

Topics

- [Logging AWS Marketplace Catalog API calls with CloudTrail](#)
- [Logging AWS Marketplace Agreement API calls with CloudTrail](#)
- [Logging AWS Marketplace Metering API calls with CloudTrail](#)
- [Logging AWS Marketplace Deployment API calls with CloudTrail](#)
- [Logging AWS Marketplace Discovery API calls with AWS CloudTrail](#)

Logging AWS Marketplace Catalog API calls with CloudTrail

The AWS Marketplace Catalog API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all calls to the Catalog API as events, including calls from the AWS Marketplace Management Portal.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon Simple Storage Service (Amazon S3) bucket. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request, the IP address from which the request was made, who made the request, when it was made, and additional details.

AWS Marketplace Catalog API information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the AWS Marketplace Catalog API, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#) in the *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#)
- [Receiving CloudTrail Log Files from Multiple Accounts](#)

All AWS Marketplace Catalog API actions are logged by CloudTrail and are documented in this API Reference. For example, calls to the `StartChangeSet`, `DescribeChangeSet`, and `ListChangeSets` API actions generate entries in the CloudTrail log files. Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see [CloudTrail userIdentity Element](#) in the *AWS CloudTrail User Guide*.

Understanding AWS Marketplace catalog log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single

request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

Note

These examples have been formatted for improved readability. In a CloudTrail log file, all entries and events are concatenated into a single line. In addition, this example has been limited to a single AWS Marketplace Catalog API entry. In a real CloudTrail log file, you see entries and events from multiple AWS services.

The following example shows a AWS Marketplace Catalog API log entry that demonstrates the `ListEntities` action:

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

Logging AWS Marketplace Agreement API calls with CloudTrail

The Agreements API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Marketplace. CloudTrail captures API calls for the Agreements API as events. The calls captured include calls from the AWS Marketplace website, console, and other interfaces leveraging the Agreements API, as well as direct code calls to Agreements API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for the Agreements API. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to the Agreements API, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the [AWS CloudTrail User Guide](#).

Agreements API information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the Agreements API, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing events with CloudTrail Event history](#) in the *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, including events for the Agreements API, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- [Overview for creating a trail](#)
- [CloudTrail supported services and integrations](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)
- [Receiving CloudTrail log files from multiple regions](#) and [Receiving CloudTrail log files from multiple accounts](#)

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see [CloudTrail userIdentity element](#) in the *AWS CloudTrail User Guide*.

Understanding Agreements API actions

The Agreements API is used to purchase software as a service (SaaS), server (including container), and professional services products on AWS Marketplace. It's also used to manage resulting agreements or subscriptions on AWS Marketplace.

Note

Purchases of AWS Data Exchange products won't be logged by CloudTrail.

The `AcceptAgreementRequest` action is used when an AWS Identity and Access Management (IAM) user or role of an AWS account purchases an applicable product on AWS Marketplace. Similarly, the `CancelAgreement` action is used when an IAM user or role cancels their agreement or subscription. By monitoring CloudTrail logs in the Agreements API, buyers can monitor the most important purchase-related actions happening in their AWS account on AWS Marketplace.

The `DescribeAgreement` action is used when the customer specifically views meta data for a specific agreement. The `GetAgreementTerms` action is used when the terms of a particular agreement are viewed. The `SearchAgreements` action is used when an IAM user or role lists or filters out a subset of their agreements from the full list of all their agreements.

Note

The `AcceptAgreementRequest` and `CancelAgreement` actions are available to buyers but not sellers. However, the `DescribeAgreement`, `GetAgreementTerms`, and `SearchAgreements` actions can be used by both buyers and sellers.

Buyers can also identify the Agreement ID of the agreement from the CloudTrail log. For more information about the agreement, choose the **Manage subscriptions** tab in the AWS Marketplace console, where the Agreement ID is provided in the **Details** view. The Agreement ID can be found in `responseElements` for the `AcceptAgreementRequest` API action and in `requestParameters` for the `CancelAgreement` API action.

Understanding Agreements API log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't display in a specific order.

The following example shows a CloudTrail log entry that demonstrates the `AcceptAgreementRequest` action.

```
{
  "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-1lnrq6riwpg2tczhv378zkn1c"
    },
    "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789010",
    "eventCategory": "Management"
  }
}
```

The following example shows a CloudTrail log entry that demonstrates the `CancelAgreement` action.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHijklmnop12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHijklmnop1234"
  },
  "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"
}
```

The following example shows a CloudTrail log entry that demonstrates the `DescribeAgreement` action.

```
{
  "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",
}
```

The following example shows a CloudTrail log entry that demonstrates the `GetAgreementTerms` action.

```
{
  "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",
}

```

The following example shows a CloudTrail log entry that demonstrates the SearchAgreements action.

```

{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "Unknown",
        "principalId": "ABCDEFGHijklmnop12345",
        "arn": "arn:aws:iam::123456789010:user/CloudtrailTestUser",
        "accountId": "123456789010",
        "accessKeyId": "ABCDEFGHijklmnop123",
    },
    "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",
}

```

Logging AWS Marketplace Metering API calls with CloudTrail

The Metering API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Marketplace. CloudTrail captures API calls for the Metering API as events. The calls captured include calls from the AWS Marketplace website,

console, and other interfaces leveraging the Metering API, as well as direct code calls to Metering API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for the Metering API. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to the Metering API, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the [AWS CloudTrail User Guide](#).

Metering API information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the Metering API, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing events with CloudTrail Event history](#) in the *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, including events for the Metering API, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- [Overview for creating a trail](#)
- [CloudTrail supported services and integrations](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)
- [Receiving CloudTrail log files from multiple regions](#) and [Receiving CloudTrail log files from multiple accounts](#)

The BatchMeterUsage action for SaaS-based products, RegisterUsage action for container-based products, and MeterUsage action for Amazon Machine Image (AMI)-based and container based products are logged by CloudTrail.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see [CloudTrail userIdentity element](#) in the *AWS CloudTrail User Guide*.

Understanding Metering API log file entries

Example: BatchMeterUsage for SaaS-based products

The following example shows a CloudTrail log entry that demonstrates the BatchMeterUsage action from the AWS Marketplace Metering API. When the seller [sends metering records to report their customers' usage](#) for a software as a service (SaaS) product listed in AWS Marketplace, this CloudTrail log entry is logged in the seller's AWS account.

```
{
  "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"
}

```

Example: RegisterUsage for containers-based products deployed on Amazon EKS

The following example shows a CloudTrail log entry that demonstrates the RegisterUsage action from the AWS Marketplace Metering API. When an hourly priced container product from AWS Marketplace is deployed in the buyer's AWS account, the software in the container calls RegisterUsage within the buyer's AWS account to initiate the hourly metering for that Amazon Elastic Container Service (Amazon ECS) task or Amazon Elastic Kubernetes Service (Amazon EKS) pod. This CloudTrail log entry is logged in the buyer's AWS account.

```

{
  "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"
  }
}

```

Example: MeterUsage for container-based products deployed on Amazon EKS

The following example shows a CloudTrail log entry that demonstrates the MeterUsage action from the AWS Marketplace Metering API for containers on Amazon EKS. When a container product with [custom metering](#) from AWS Marketplace is deployed in the buyer's AWS account, the software in the container calls MeterUsage within the buyer's AWS account to report each hour. This CloudTrail log entry is logged in the buyer's AWS account.

```
{
  "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"
  }
}

```

Example: MeterUsage for AMI-based products

The following example shows a CloudTrail log entry that demonstrates the `MeterUsage` action from the AWS Marketplace Metering API for Amazon Machine Images (AMIs). When an AMI product with custom metering from AWS Marketplace is deployed in the buyer's AWS account, the software from the AMI calls `MeterUsage` within the buyer's AWS account to report usage each hour. This CloudTrail log entry is logged in the buyer's AWS account.

```

{
  "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"
}

```

Logging AWS Marketplace Deployment API calls with CloudTrail

The AWS Marketplace Deployment Service is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon Simple Storage Service (Amazon S3) bucket. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request, the IP address from which the request was made, who made the request, when it was made, and additional details.

AWS Marketplace Deployment Service information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the AWS Marketplace Deployment Service, that activity is recorded in a CloudTrail event along with

other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#) in the *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#)
- [Receiving CloudTrail Log Files from Multiple Accounts](#)

All Deployment Service actions are logged by CloudTrail and are documented in this API reference. For example, calls to the `PutDeploymentParameterAPI` action generates entries in the CloudTrail log files. Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see [CloudTrail userIdentity Element](#) in the *AWS CloudTrail User Guide*.

Understanding AWS Marketplace log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

Note

These examples have been formatted for improved readability. In a CloudTrail log file, all entries and events are concatenated into a single line. In addition, this example has been limited to a single Deployment Service entry. In a real CloudTrail log file, you see entries and events from multiple AWS services.

The following example shows a Deployment Service log entry that demonstrates the `PutDeploymentParameter` action:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "Unknown",
    "principalId": "ABCDEFGHijklmnop12345",
    "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
    "accountId": "123456789010",
    "accessKeyId": "ABCDEFGHijklmnop123"
  },
  "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"
}
```

Logging AWS Marketplace Discovery API calls with AWS CloudTrail

The AWS Marketplace Discovery API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all Discovery API calls as events, including calls from the AWS SDK and the AWS CLI.

Discovery API information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the Discovery API, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**.

All Discovery API actions are logged by CloudTrail. The following operations generate entries in CloudTrail log files:

- `GetListing`
- `GetProduct`
- `GetOffer`
- `GetOfferTerms`
- `GetOfferSet`
- `ListPurchaseOptions`
- `ListFulfillmentOptions`
- `SearchFacets`
- `SearchListings`

The event source for Discovery API events is `discovery-marketplace.amazonaws.com`.

Example Discovery API log entry

The following example shows a CloudTrail log entry that demonstrates the `GetListing` action.

```
{
  "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"
}
```

Notifications for the AWS Marketplace API

The following sections provide information about notifications.

Topics

- [Amazon EventBridge notifications for AWS Marketplace Catalog API events](#)

Amazon EventBridge notifications for AWS Marketplace Catalog API events

AWS Marketplace is integrated with Amazon EventBridge, formerly called Amazon CloudWatch Events. EventBridge is an event bus service that you can use to connect your applications with data from a variety of sources.

For information on how sellers, channel partners, and private marketplace administrators can receive ChangeSet status events using EventBridge, see [Events for change sets](#) in the *AWS Marketplace Seller Guide*.

About the AWS Marketplace MCP server

AWS Marketplace Model Context Protocol (MCP) provides AI-powered tools for AWS Marketplace discovery, evaluation, research, and proposal generation. MCP is a standardized protocol that enables AI assistants to interact with external systems through the following:

- **Tools:** Functions that AI models can invoke to perform actions.
- **Resources:** Data and context that can be shared with AI models.

All tools are stateless — each call is independent with no server-side conversation state.

Example use cases

The following are common ways to use the AWS Marketplace MCP server:

- Find products that address business needs using natural language queries.
- Evaluate multiple vendors quickly with AI-generated comparison reports.
- Create comprehensive purchase proposals tailored to business requirements.
- Get side-by-side analysis of competing solutions.

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

Protocol: MCP over Streamable HTTP (JSON-RPC 2.0). All responses are single JSON payloads.

Authentication: None required.

Topics

- [Available tools](#)
- [Integration guide](#)
- [Tool details](#)
- [Workflows](#)
- [End-to-end example](#)
- [Error handling](#)

Available tools

All 6 tools are stateless and require no conversation context or session management. All tool definitions, including full input/output schemas, are available programmatically through the standard MCP `tools/list` endpoint. The schemas returned by `tools/list` are the canonical, always-current source of truth for parameter names, types, and constraints.

The following table summarizes each tool and whether it is read-only.

Tool name	Description	Read-only
<code>get_aws_marketplace_report_guidelines</code>	Returns workflow instructions and report format templates for recommendations, comparisons, or evaluations. Call first for any report workflow.	Yes
<code>search_aws_marketplace_solutions</code>	Searches the AWS Marketplace catalog by queries. Supports batch queries (up to 10) and cursor-based pagination.	Yes
<code>get_aws_marketplace_solution</code>	Gets detailed solution metadata by solution IDs (reviews, pricing, sentiments, rankings, CTAs).	Yes
<code>get_aws_marketplace_related_solutions</code>	Gets related solutions for given solution IDs, ranked by relation score.	Yes
<code>research_aws_marketplace_solution</code>	Performs deep web research on solutions (features, reviews, limitations, competitors, pricing, integrations, case studies). Hidden from clients with native web search.	Yes
<code>submit_aws_marketplace_feedback</code>	Submits user feedback (positive/negative) about report quality.	No

Integration guide

You can integrate with the AWS Marketplace MCP server in two ways: through a no-code Claude Connector or through a programmatic MCP client.

Claude Connector (recommended)

The fastest way to use AWS Marketplace MCP tools is through the **AWS Marketplace connector for Claude**. This requires no coding or MCP client setup — Claude handles tool discovery and invocation automatically.

Setup:

1. Open the [AWS Marketplace connector](#) in the Claude Connectors Directory.
2. Choose **Connect** to enable it.
3. In any Claude conversation, choose the **+** button (lower left) or type **/** to open the menu.
4. Hover over **Connectors** and toggle on **AWS Marketplace**.
5. Ask Claude to find, compare, or evaluate AWS Marketplace products — it automatically uses the MCP tools.

For more details on how connectors work, see [Use connectors to extend Claude's capabilities](#).

Example prompts:

- "Find the best SIEM solutions on AWS Marketplace for a mid-size company"
- "Compare Datadog vs New Relic on AWS Marketplace"
- "Build an evaluation report for Terraform Cloud on AWS Marketplace"

MCP client integration

For programmatic integrations or custom agent frameworks, connect directly to the MCP endpoint.

Prerequisites

You need the following to get started:

1. An MCP-compatible agent framework or client.
2. An HTTP client capable of making JSON-RPC 2.0 requests.

Step 1: Set up MCP client

Use a standard MCP client that supports HTTP transport.

Endpoint:

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

The following methods are supported:

- POST with method: "initialize" — Initialize session and obtain Mcp-Session-Id.
- POST with method: "tools/list" — Discover available tools and their schemas.
- POST with method: "tools/call" — Invoke a tool.

Authentication: No authentication is required. Omit auth headers in HTTP requests.

Step 2: Obtain and register tools (tool discovery)

All tool definitions are available programmatically through the standard MCP `tools/list` method. This is the recommended way to obtain current input/output schemas, as tool schemas may be updated independently of this documentation. Call `tools/list` once during initialization and register the returned tools in your agent's tool manifest or system prompt.

The following example shows a `tools/list` request.

Request:

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

Response (abbreviated):

```
{
  "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": { ... }
  }
]
}
}

```

Each tool entry includes the tool name, a detailed description with usage guidance and examples, and a complete `inputSchema` with parameter types, constraints, and validation rules.

Step 3: Configure telemetry

AWS Marketplace MCP requires a stable identifier on every tool call for request correlation and telemetry. This is not an authentication credential — it is used only for diagnostics and analytics.

Pass the identifier in the custom HTTP header:

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

- **Browser clients:** Derive from an existing session cookie or a frontend-generated UUID. Use the same value across all tool calls.
- **Programmatic clients:** Use POST with method: "initialize" to obtain an Mcp-Session-Id.

HTTP transport

AWS Marketplace MCP uses HTTP transport, where the server operates as an independent HTTP service that can handle multiple client connections.

The following are key transport characteristics:

- **HTTP POST for requests:** Every client message is sent as a new HTTP POST request.
- **JSON responses:** All tools return JSON responses (Content-Type: application/json).
- **Session management:** Session IDs through the Mcp-Session-Id header for request correlation and telemetry.
- **Protocol version:** Specified through the MCP-Protocol-Version: 2025-06-18 header.

Required HTTP headers:

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

The following example shows a tool call request and response.

Example request:

```
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"]
    }
  }
}
```

Example response:

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

Tool details

This section describes each tool's input parameters, output format, and usage guidance.

get_aws_marketplace_report_guidelines

Returns workflow instructions and output format template for a given report type. **Call this first** for any recommendation, comparison, or evaluation workflow.

This tool returns different prompts depending on client capabilities — clients with native web search get prompts with embedded search queries instead of `research_aws_marketplace_solution` tool references.

Input:

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

The following table describes the input parameters.

Field	Type	Required	Description
workflow_type	enum	Yes	"recommendations" for ranked product discovery, "comparison" for side-by-side analysis, "evaluation" for single-product assessment.

Output:

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

The following table describes the output fields.

Field	Type	Description
workflow_type	string	The requested workflow type.
guidelines	string	Combined workflow instructions and markdown format template.

When to use each workflow:

- **recommendations** — "find monitoring tools", "I need a CI/CD solution", "SOC2-compliant security scanning under \$5k/month".
- **comparison** — "compare Datadog vs New Relic", "Splunk vs Elastic for log management".
- **evaluation** — "build evaluation report for Terraform Cloud", "justify purchasing New Relic to my CTO".

search_aws_marketplace_solutions

Performs a direct AWS Marketplace Catalog API search. Returns raw, unranked, unfiltered results without AI processing or relevance scoring. Supports batch queries (up to 10) and cursor-based pagination for single queries.

Input:

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

The following table describes the input parameters.

Field	Type	Required	Description
queries	string[]	Yes	Array of search queries (1-10). Also accepts legacy query (single string, auto-converted to array).
max_results	number	No	Maximum results per query (1-100, default: 10).
cursor	string	No	Pagination cursor (only works for single query, ignored for batch).

Output:

```
{
  "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": "eyJvZmZzZXQiOjEwLCJwYWdlU216ZSI6MTB9"
```

The following table describes the output fields.

Field	Type	Description
results	MinimalSolutionMetadata[]	Array of matching solutions.
next_cursor or	string null	Pagination cursor for next page (single query only). null when no more results.

Batch behavior: For multiple queries, results are combined and deduplicated by `solution_id`. The pagination cursor is not returned for batch requests.

Pagination: If `next_cursor` is returned (single-query requests only), pass it back as the `cursor` parameter to retrieve additional results. When `next_cursor` is `null`, all matching solutions have been retrieved.

get_aws_marketplace_solution

Performs a direct AWS Marketplace Catalog API call. Retrieves comprehensive solution metadata by solution ID without AI processing. Supports batch requests (up to 10 IDs).

Input:

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

The following table describes the input parameters.

Field	Type	Required	Description
solution_ids	string[]	Yes	Array of solution IDs (1-10). IDs start with prodview- . Also accepts legacy solution_id (single string, auto-converted).

Output:

```
{
  "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://..."
}
}
]
```

The following table describes the output fields.

Field	Type	Description
solutions	SolutionOutput[]	Array of solution details (always array, even for single ID).

The following table describes the fields in each SolutionOutput object.

Field	Type	Description
solution_id	string	AWS Marketplace solution ID.
solution_name	string	Product name.
solution_description	string	Long or short description.
solution_url	string	Marketplace URL with ref tags.

Field	Type	Description
vendor_name	string	Vendor display name.
vendor_url	string	Vendor profile URL with ref tags.
highlights	string[]	Product highlights.
reviews_summary	object	{ reviews_count, average_rating }
review_sentiments	ReviewSentiment[]	Sentiment by category (functionality, ease_of_use, customer_service, cost_effectiveness) with polarity scores and sample reviews.
free_trial_available	boolean?	Whether a free trial is available.
pricing_options	PricingOption[]	{ name, description } pairs.
fulfillment_options_types	string[]?	Fulfillment types such as SaaS, Container, or AML.
categories	string[]?	Categories such as "Monitoring & Observability".
security_certificates	string[]?	Certifications such as "SOC 2 Type II" or "ISO 27001".
rankings	Ranking[]?	{ category, rank } — rank within category (1-100).
has_standard_contract	boolean?	Whether the product supports AWS Standard Contract.
call_to_action	CallToAction	{ procurement_url, request_for_demo_url, request_for_private_offer_url, procure_free_trial_url }

get_aws_marketplace_related_solutions

Performs a direct AWS Marketplace Catalog API call. Returns related solutions for given solution IDs as identified by AWS Marketplace, sorted by relation score (descending), without AI analysis or relevance scoring. Supports batch requests (up to 10 IDs).

Input:

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

The following table describes the input parameters.

Field	Type	Required	Description
solution_ids	string[]	Yes	Array of solution IDs (1-10). Also accepts legacy solution_id (single string, auto-converted).

Output:

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

The following table describes the output fields.

Field	Type	Description
results	SingleRelatedSolutionsResult[]	One entry per input solution ID.
results[].solution_id	string	The input solution ID.
results[].related_solutions	MinimalSolutionMetadata[]	Related solutions sorted by relevance.

research_aws_marketplace_solution

Performs deep web research on AWS Marketplace solutions. Runs web searches for each requested section and returns content with inline citation links. Supports batch requests (up to 5 IDs).

Visibility: This tool is hidden from clients with native web search capability (for example, Anthropic/ClaudeAI). Those clients get workflow prompts with embedded search queries instead.

Input:

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

The following table describes the input parameters.

Field	Type	Required	Description
solution_ids	string[]	Yes	Array of solution IDs in prodview-* format (1-5).
sections	ResearchSection[]	No	Sections to research. Default: all 7 sections.

The following table describes each available research section.

Section	What it returns
features	Official documentation, capabilities, specifications
reviews	G2, Gartner, TrustRadius customer feedback
limitations	Known issues, drawbacks, criticisms
competitors	Competitive positioning, alternatives
pricing	Cost information, pricing models, tiers
integrations	AWS ecosystem, APIs, third-party tools
case_studies	Customer success stories, testimonials, ROI

Output:

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

The following table describes the output fields.

Field	Type	Description
<code>results</code>	<code>SingleResearchResult[]</code>	One entry per input solution ID.
<code>results[].solution_id</code>	<code>string</code>	The researched solution ID.
<code>results[].solution_name</code>	<code>string</code>	Product name resolved from solution ID.
<code>results[].research</code>	<code>object</code>	Object with keys for each requested section.
<code>results[].research[section].content</code>	<code>string</code>	Research content with inline citation links.
<code>results[].research[section].citations</code>	<code>Citation[]</code>	{ index, title, url } for each citation.

Concurrency: Solutions are researched in parallel. Sections within each solution are sequential to avoid overwhelming downstream services.

submit_aws_marketplace_feedback

Submits user feedback about an AWS Marketplace report. Call this tool after completing any recommendation, comparison, or evaluation workflow.

Input:

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

The following table describes the input parameters.

Field	Type	Required	Description
sentiment	enum	Yes	"positive" or "negative" .
categories	string[]	No	For negative feedback: "irrelevant" , "incomplete" , "inaccurate" , "other".
feedback_text	string	No	Free-form text (max 1000 characters).

Output:

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

Workflows

The AWS Marketplace MCP server supports three primary workflows for product research and direct catalog lookups.

Report generation (recommendations, comparisons, evaluations)

This is the primary workflow for AI-assisted product research. It uses `get_aws_marketplace_report_guidelines` as the orchestration entry point.

Flow:

1. Call `get_aws_marketplace_report_guidelines` with the appropriate `workflow_type`.
2. The tool returns step-by-step workflow instructions and an output format template.
3. The AI assistant follows the returned instructions, which typically involve calling catalog API and research tools to gather data.
4. The assistant generates a formatted report following the output template.
5. After delivering the report, the assistant collects user feedback through `submit_aws_marketplace_feedback`.

Recommendations workflow:

1. `get_aws_marketplace_report_guidelines({ workflow_type: "recommendations" })`
2. `search_aws_marketplace_solutions({ queries: ["Product A", "Product B"], max_results: 3 })`
3. Write report following the format template.
4. `submit_aws_marketplace_feedback({ sentiment: "positive" })`

Comparison workflow:

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. Build comparison matrix and write report following the format template.
5. `submit_aws_marketplace_feedback({ sentiment: "positive" })`

Evaluation workflow:

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. Write evaluation report following the format template.
5. `submit_aws_marketplace_feedback({ sentiment: "positive" })`

Direct catalog lookup

For programmatic integrations or quick lookups that don't require AI analysis, you can call the catalog API tools directly without `get_aws_marketplace_report_guidelines`.

Search then detail lookup:

1. Call `search_aws_marketplace_solutions` with one or more keyword queries.
2. Use the returned `solution_id` values to call `get_aws_marketplace_solution` for full details.
3. Optionally call `get_aws_marketplace_related_solutions` to discover related products.

Feedback collection

After completing any workflow, the assistant should collect user feedback.

1. Ask the user if the response was helpful.
2. If positive, call `submit_aws_marketplace_feedback` with `sentiment`: "positive".
3. If negative, ask which categories apply (irrelevant, incomplete, inaccurate, other), optionally collect free-text feedback, and submit.

End-to-end example

The following example demonstrates a complete recommendations workflow from initial query to feedback submission.

Step 1 — Get workflow instructions:

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

Step 2 — Search for candidate solutions:

```
{
  "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
  }
}
```

Step 3 — Retrieve detailed metadata for top candidates:

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

Step 4 — Submit user feedback:

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

Error handling

Errors fall into three categories: protocol errors, tool execution errors, and validation errors.

Protocol errors

Protocol errors are returned when the request violates the JSON-RPC or MCP specification (for example, malformed JSON, missing required fields, or unknown methods).

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

Tool execution errors

Tool execution errors are returned when a tool invocation fails at runtime (for example, rate limiting, invalid solution IDs, or upstream service errors). Tool errors are reported within the result object, not as MCP protocol-level errors, so the AI assistant can see and handle the error.

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

Validation errors

Validation errors are returned when input fails schema validation (for example, invalid solution ID format, exceeding max array length, or missing required fields). These are surfaced as tool execution errors with `isError: true`.

Using the AWS Marketplace API as a seller

The following sections provide information on how to use the API as a seller.

Topics

- [Work with seller products](#)
- [Work with offers using the AWS Marketplace APIs](#)
- [Work with multi-product solutions](#)
- [Build interfaces with catalog data](#)

Work with seller products

You can use the AWS Marketplace Catalog API to automate tasks for working with seller products. This includes the ability to create, update, view, list, and sort products. This enables you to automate product management. For example, you can provide self-service publishing capabilities on the AWS Marketplace Management Portal.

A *product* is a unit or resource that you intend to sell in AWS Marketplace, often referred to as a base product. Buyers can't consume a base product until you add product information, deployment attributes, and billing information.

A *product* describes the product information, software deployment attributes, and billing mechanism of the listing that you intend to sell. The *product* must be paired with an *offer* to become a transactable unit that you can sell and buyers can use in AWS Marketplace.

You can also use the AWS Marketplace Catalog API to:

- [Work with private offers using the AWS Marketplace APIs](#)
- [Work with resale authorizations using the AWS Marketplace APIs](#)
- [Work with channel partner private offers using the AWS Marketplace APIs](#)

Each product type has a different product entity. An *entity* can be a product or an offer on AWS Marketplace. The following product types and entities are supported:

Product type	Entity
Amazon Machine Image (AMI) products	AmiProduct@1.0
Container products	ContainerProduct@1.0
Software as a service (SaaS) products	SaaSProduct@1.0
Machine learning (ML) products	MachineLearningProduct@1.0

Note

Single-AMI with CloudFormation product types, AWS Data Exchange data products, and professional services products are not supported.

The following topics assume you have access to the API and have completed any seller prerequisites, as described in [Access control for the AWS Marketplace Catalog API](#).

See the following resources:

- To understand the basics of using the AWS Marketplace Catalog API, see [Using the AWS Marketplace Catalog API](#).
- For end-to-end labs with working code examples, see [Manage products with API](#) in the *AWS Marketplace seller workshop*.
- For code examples of API requests, see [Python](#) and [Java](#) examples in *AWS Samples* on GitHub.

The following topics describe how to use the Catalog API to perform actions on your single-AMI products, container-based products, or SaaS products.

Topics

- [Create a product](#)
- [Update product details](#)
- [Add pricing dimensions](#)
- [Update pricing dimensions](#)

- [Restrict pricing dimensions](#)
- [Update targeting configuration](#)
- [Update product visibility](#)
- [Publish a product](#)
- [Find your product ID](#)
- [Change set status and errors](#)
- [Work with AMI-based products](#)
- [Work with EC2 Image Builder component products](#)
- [Work with container-based products using the AWS Marketplace APIs](#)
- [Work with SaaS products using the AWS Marketplace APIs](#)
- [Work with machine learning products using AWS Marketplace APIs](#)

Create a product

Note

This change type is only needed when you intend to create a brand new product entity in the AWS Marketplace catalog. It is not needed when updating existing products.

You can use the Catalog API to create an AMI, container, machine learning, or SaaS product document with identifiers (product code and product ID) in AWS Marketplace.

You create a product in Draft state by calling the StartChangeSet API operation with the CreateProduct change type.

If your request is processed successfully, then AWS Marketplace Catalog API generates a product in Draft state for you. This is an incomplete product and isn't visible to buyers in AWS Marketplace.

You then use Update change types to complete the create product process: [UpdateInformation](#), [UpdateDimensions](#), [UpdateTargeting](#), and [UpdateVisibility](#).

After the product is completed, you can use the [ReleaseProduct](#) change type to complete the product creation process, and then release the offer. This process validates the entire product and moves the product to the Limited state.

Note

For more information about creating a product using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- [Create your single-AMI product](#)

You cannot update the AMI for the version. If you need to update the AMI, create a new version instead.

- [Creating a container product](#)
- [Creating a SaaS product](#)
- [Creating a machine learning product](#)

If you use the AWS Marketplace Management Portal to create a product, the product will be in the Staging state.

To create a product in Draft state, call the StartChangeSet API operation with the CreateProduct change type, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

```
}
```

Provide information for the fields to add the `CreateProduct` change type. This change type can take in `ProductTitle` attribute, subject to the same restrictions as that sent into `UpdateInformation` change type.

- **Entity (object) (required)** – The named type of object being created.
 - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`, `ContainerProduct@1.0`, `MachineLearningProduct@1.0` or `SaaSProduct@1.0`. For more information, see [Identifier](#).
- **DetailsDocument (object) (required)** – It may be empty.
- **ProductTitle (optional)** – The title for your product, max length is 72 characters. Note that you can also later set or update the product title via the `UpdateInformation` change type.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the Status is SUCCEEDED), a new `ProductId` is generated.

Synchronous Validations

The following schema validations are specific to `CreateProduct` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
ProductTitle (string)	Max length: 72	400

Asynchronous Errors

The following errors are specific to `CreateProduct` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more details about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in ProductTitle field. Provide ProductTitle with no inappropriate content.

Update product details

If you already have a product in AWS Marketplace, you can use the Catalog API to update the product details for an AMI, container, ML, or SaaS product.

Note

For more information about updating the product details using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update product information](#)
- Container-based product: [Creating or updating product information for your container product](#)
- SaaS-based product: [Update product information](#)
- Machine learning product: [Updating product information](#)

To update product details, call the StartChangeSet API operation with the UpdateInformation change type and the details that you want to change, as shown in the following example.

Request Syntax

```
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"
    ]
  }
}
}
}

```

Provide information for the fields to add the UpdateInformation change type:

- Entity (object) (required) – The named type of entity being created.
 - Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
 - Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0, ContainerProduct@1.0, MachineLearningProduct@1.0, or SaaSProduct@1.0.
- DetailsDocument (object) (required) – The details of the request including the information that you want to update for your product. Each field is optional, but you must include at least one change to update.
 - ProductTitle (string) – The name of the product to be displayed to buyers.
 - ShortDescription (string) – The description of key aspects of the product to be displayed to buyers. This is usually 2–3 sentences.
 - LongDescription (string) – The longer description of your product to be displayed to buyers. This is usually 1–3 paragraphs.
 - Sku (string or null) – The free-form string that you define as a reference for your own use. Use null to unset this field.
 - LogoUrl (string) – The URL to an image in a publicly accessible Amazon Simple Storage Service (Amazon S3) bucket. For more information, see [Company and product logo requirements](#).

- **VideoUrls** (array of strings) – The list of URLs to publicly available, externally hosted videos to be provided as a reference to buyers in your product information.

 Note

Currently, AWS Marketplace supports one URL in the array.

- **Highlights** (array of strings) – The list of short callouts for key product features.
- **AdditionalResources** (array of structures) – The list of references to additional resources to learn about your product. Each reference is made up of a text name and a URL:
 - **Text** (string) – The name or title of the resource.
 - **Url** (string) – The URL to a resource that might be helpful for a buyer to understand your product.
- **SupportDescription** (string) – The details about your support offering for your product.
- **Categories** (array of strings) – The list of AWS Marketplace defined product categories that describe your product. For more information, see [Product categories](#) in the *AWS Marketplace Buyer Guide*.
- **SearchKeywords** (array of strings) – The list of keywords for your product to enhance the search experience. Seller name, product name, and product categories are automatically included in search keywords and don't need to be repeated here.

 Note

When you are initially populating product information (metadata) for a Draft product, you will need to supply all of the following in the **DetailsDocument** object of **UpdateInformation** change type: **ProductTitle**, **ShortDescription**, **LongDescription**, **LogoUrl**, **Highlights**, **AdditionalResources**, **SupportDescription**, **Categories**, and **SearchKeywords**.

The **ProductTitle** can be omitted if it has already been provided during **CreateProduct** change type. However, when you are updating existing fields on the product, you can include only the attributes that need to be changed in the **DetailsDocument** object of the **UpdateInformation** change type.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed.

To check request status, use the AWS Marketplace Management Portal or call the [DescribeChangeSet](#) API.

Synchronous Validations

The following schema validations are specific to `UpdateInformation` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
String (general)	No control characters "\\x00-\\x08\\ \\x0B-\\x1F"	400
ProductTitle (string)	Max length: 72 Required	400
ShortDescription (string)	Max length: 1000 Required	400
LongDescription (string)	Max length: 5000 Required	400
Sku (string)	Max length: 100 Optional	400
LogoUrl (string)	URL pattern:	400

Input field	Validation rule	HTTP code
	<code>^https://(www\.)?[-a-zA-Z0-9@.]</code> <code>{1,256}\.[a-zA-Z0-9()]{2,63}\b([-a-</code> <code>zA-Z0-9@+./]*)</code> Required	
VideoUrls (array of strings)	URL pattern: <code>https://(www\.)?[-a-zA-Z0-9@.]</code> <code>{1,256}\.[a-zA-Z0-9()]{2,63}\b([-a-</code> <code>zA-Z0-9@_+./\])</code> Optional	400
Highlights (array of strings)	Required: Min 1 - Max 3	400
AdditionalResources (array of structures)	Max length: 500 Optional	400
SupportDescription (string)	Max length: 2000 Required	400
Categories (array of strings)	Min 1 - Max 3 Required	400
SearchKeywords (array of strings)	Min 1 - Max 15 Max 50 chars for each item Required	400

Asynchronous Errors

The following errors are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is

processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
MISSING_DATA	No data provided to perform an update. Provide data for at least 1 field of the product.
INVALID_INPUT	Provide LogoUrl.
INVALID_INPUT	Provide ProductTitle.
INVALID_INPUT	Provide ShortDescription.
INVALID_INPUT	Provide LongDescription.
INVALID_INPUT	Provide SupportDescription.
INVALID_INPUT	Provide at least one search keyword.
INVALID_INPUT	Provide at least one highlight.
INVALID_INPUT	Provide between 1 and 3 product categories.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in ProductTitle field. Provide ProductTitle with no inappropriate content.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in ShortDescription field. Provide ShortDescription with no inappropriate content.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in LongDescription field. Provide LongDescription with no inappropriate content.
INVALID_INPUT	Inappropriate content '{InappropriateContent}' found in SupportDescription field.

Error code	Error message
	Provide SupportDescription with no inappropriate content.
INVALID_INPUT	Invalid ProductTitle field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid ShortDescription field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid LongDescription field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid SupportDescription field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid ProductTitle field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Invalid ShortDescription field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Invalid LongDescription field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Invalid SupportDescription field. Remove unsupported characters [UnsupportedCharacters].
INVALID_INPUT	Search keywords must be no more than 250 combined characters.
INVALID_INPUT	The input for this change type could not be read. Submit a properly formatted input.
INVALID_ADDITIONAL_RESOURCES	Invalid URLs in AdditionalResources: [InvalidAdditionalResourcesUrls] Provide valid URLs.

Error code	Error message
INVALID_CATEGORY_NAMES	Provide valid category names supported by AWS Marketplace.
InvalidImageProperties	Validation errors found: The file is not image type. Supported image types: [png jpg gif].
EXPLICIT_CONTENT	Explicit content: '{ExplicitContent}' detected. Provide media with no explicit content.
INVALID_MEDIA	Invalid URL: {MediaUrl} Provide a new URL for media stored in S3.
INVALID_MEDIA	Invalid URL: {MediaUrl} Provide a valid URL that does not exceed 2048 characters.
INVALID_MEDIA	Location provided not accessible: {MediaUrl} Provide an accessible URL for media stored in S3.
INVALID_MEDIA	There was an issue copying the media from S3. Image size exceeds 5 MB. Provide an image that is under 5 MB.
INVALID_MEDIA	Malware detected in media. Please resubmit media without malware.
TOO_MANY_MEDIA	Provide no more than 15 media items.
DUPLICATE_MEDIA	Duplicate media is not allowed for a product. Please provide media with no duplicates.

Add pricing dimensions

You can use the [AWS Marketplace Catalog Service Actions](#) to add billable pricing dimensions that enable you to charge users for AMI, container, or SaaS products.

A *pricing dimension* is a unit of measure that sellers define for charging buyers. Sellers must set up this information to bill buyers for using the product, whether it's a usage-based or contract-based pricing model. The type of dimension depends on the product's pricing model.

 Note

For SaaS products with Free pricing model, you must create at least one dimension with `UsageBasedPricingTerm` or `ConfigurableUpfrontPricingTerm`, and all dimensions must be priced at \$0.00. This requirement is unique to SaaS products and does not apply to AMI, container, or machine learning products.

 Note

New pricing dimensions have the following impacts on SaaS buyers:

- For buyers with agreements created from public offers, you can report consumption on the new dimensions even though they did not exist in the offer when the agreement was created.
- For buyers with agreements created from private offers, you can't report consumption on the new dimensions because they did not exist in the private offer when the agreement was created. Calls to the [BatchMeterUsage](#) API will succeed, but the buyer will not be billed, so you must keep track of which buyer can and cannot be billed for any new dimensions. You can also use the [GetAgreementTerms](#) API to see the dimensions included in each buyer agreement.

To report consumption on a new dimension and bill the buyer, you must [extend a replacement offer](#) that includes the dimension, and the buyer must accept the offer.

For more information about product pricing, see the following topics in the *AWS Marketplace Seller Guide*:

- [AMI product pricing](#)
- [Container products pricing](#)
- [SaaS product pricing](#)
- [Machine learning product pricing](#)

For more information about adding pricing dimensions using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based products: [Update pricing](#).
- Container-based products: [Adding a pricing dimension](#).
- SaaS-based products: [Add pricing dimensions](#).
- ML products: Not supported. Machine learning products have fixed pricing dimensions. However, you can [Update pricing](#).

To add pricing dimensions, call the `StartChangeSet` API with the `AddDimensions` change type, as shown in the following example.

Note

After submitting the first `AddDimensions` change type with dimensions specifying a type of pricing model—usage, contract, or contract with consumption— you must work with the AWS Marketplace Seller Operations team. They help you add a dimension with types that are outside of the original pricing model.

Request Syntax

```
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"
    }
}
]
}

```

Provide the following fields for the AddDimensions change type.

- **DetailsDocument** (array of objects) (required) – Details of the request.
 - **Description** (string) (required) – Full details of the dimension that will be the long description on the buyer's viewing page.
 - **Key** (string) (required) – Enter in the facet that will be used for defining the rates in the offer. Also, enter the dimensions published to the AWS Marketplace Metering Service (MMS) if the dimension can't be metered externally. After the dimension is created, this can't be changed.
 - **Units** (string) (required) – The unit type for the dimension. Possible units are Users, Hosts, GB, MB, TB, Gbps, Mbps, Requests, Units, UserHrs, UnitHrs, Units, HostHrs, TierHrs, and TaskHrs.
 - **Name** (string) (required) – The display name for the dimension on the website and customer's bill.
 - **Types** (array of strings) (required) (also known as **Tags**) – These indicate whether the dimension covers metering, entitlement, or support for external metering. This is not changeable after the dimension is created.
 - **Metered** – Indicates that Commerce Platform usage types should be created to allow metering to occur for this dimension.
 - **ExternallyMetered** – Indicates that AWS Marketplace Metering Service (MMS) dimensions should be created during publishing to allow sellers to meter through the AWS SDK.
 - **Entitled** – Indicates that entitlements can be granted for the dimension during the product or offer publishing.

The following table lists the supported combinations of pricing dimensions and products.

Pricing dimension type	Product types
[Metered]	AMI
[Metered, ExternallyMetered]	SaaS, AMI/Flexible Consumption Pricing (FCP) When ExternallyMetered appears, Metered is masked/inferred.
[Entitled]	SaaS Contracts, ProServ Products The Entitled tag grants rights to use a software/service start and end dates for the usage. Also, to grant rights to have usage discount for AMI annual products. Each entitlement is identified by a Dimension Key in AWS Marketplace Entitlement Service for creating or updating the entitlements.
[Metered, ExternallyMetered, Entitled]	Contracts with consumption pricing, where dimensions can be prepaid or metered, are a combination of both [ExternallyMetered] and [Entitled].

- **Entity (object) (required)** – The named type of entity being created.
- **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
- **Type (string) (required)** – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0` or `SaaSProduct@1.0`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to AddDimensions actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Description	Max length: 1000	400
	Required	
Key	Max length: 100	400
	Pattern: [A-Za-z0-9_.-]+\$ Required	
Dimension Units	Max length: 20	400
	Required	
Name	Max length: 500	400
	Required	
Type (tag)	Required: Min 1 - Max 3	400

Input field	Validation rule	HTTP code
	Inputs: Entitled, Metered, Externall yMetered	
	Required	

Asynchronous Errors

The following errors are specific to AddDimensions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
MISSING_DATA	No data provided to perform an update. Provide data for at least 1 dimension.
INVALID_DIMENSION	Provide no more than 200 dimensions.
INVALID_DIMENSION	Can't add duplicate dimensions.
INVALID_DIMENSION	Dimension can't be added in current state '%s'. States that support dimension updates are %s.
INVALID_DIMENSION	Can't add dimension. The field '%s' has duplicate values '%s' in other dimensions.
INVALID_DIMENSION	Provide non-empty fields (Key, Unit, Name, Types) for each dimension.
IINVALID_TYPE	Remove invalid type '%s'. Valid types are ["Metered", "Entitled", "ExternallyMetered"].
INVALID_UNIT	Remove invalid Unit '%s'. Valid units are ["GB", "Gbps", "HostHrs", "Hosts", "MB",

Error code	Error message
	"Mbps", "Requests", "TaskHrs", "TB", "TierHrs", "UnitHrs", "Units", "UserHrs", "Users"].
INVALID_INPUT	Inappropriate content '%s' found in %s field. Provide %s with no inappropriate content.
INVALID_INPUT	Invalid '%s' field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid '%s' field. Remove unsupported characters %s.
INVALID_DIMENSION	Remove invalid dimension type combination %s. Allowed values are %s.
INVALID_DIMENSION	Remove invalid dimension key '%s' for Metered dimension.
INVALID_DIMENSION	Dimension named '%s' for productCode '%s' did not pass AWS Marketplace Metering Service validation %s.
INVALID_DIMENSION	Dimension named '%s' for productCode '%s' has no metering record present in Metering Service. The product has either never been launched for testing or is misconfigured and does not make the appropriate calls to the AWS Marketplace Metering Service.

Update pricing dimensions

You can use the Catalog API to update existing pricing dimensions of an AMI, container, or SaaS product in AWS Marketplace.

Each dimension is uniquely identified by the dimension key and dimension types to perform the update. Updating a dimension doesn't affect any active offer or customers that the original dimension had created.

Note

For more information about updating pricing dimensions using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update pricing](#)
- Container-based product: [Updating dimension information](#)
- SaaS-based product: [Update pricing dimensions](#)
- Machine learning product: [Update product pricing](#)

To update pricing dimensions, call the `StartChangeSet` API operation with the `UpdateDimensions` change type, as shown in the following example.

Request Syntax

```
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"
```

```

    }
  }
]
}

```

Use the following fields with the `UpdateDimensions` change type:

- **DetailsDocument** (array of objects) (required) – Details of the request.
- **Key** (string) (required) – Provide key of existing dimension from the product to change description and name on. For `UpdateDimension`, this field is only for identifying the dimension to be changed.
- **Types** (array of strings) (required) (also known as **Tags**) – These indicate whether the dimension covers metering, entitlement, or support for external metering. This is not changeable after the dimension is created.
 - **Metered** – Indicates that Commerce Platform usage types should be created to allow metering to occur for this dimension.
 - **ExternallyMetered** – Indicates that AWS Marketplace Metering Service (MMS) dimensions should be created during publishing to allow sellers to meter through the AWS SDK.
 - **Entitled** – Indicates that entitlements can be granted for the dimension during product/offer publishing.

Valid Pricing Dimension Types Combinations

Pricing Dimension Type	Product
[Metered]	AMI
[ExternallyMetered]	SaaS, AMI/Flexible Consumption Pricing (FCP) When <code>ExternallyMetered</code> appears, Metered is masked/inferred.
[Entitled]	SaaS Contracts, ProServe Products The <code>Entitled</code> tag grants rights to use a software/service set start and end dates for the usage. Also, to grant rights to have usage discount for AMI annual

Pricing Dimension Type	Product
	products. Each entitlement is identified by a Dimension Key in AWS Marketplace Entitlement Service for creating or updating the entitlements.
[ExternallyMetered, Entitled]	Contracts with Consumption Pricing, where dimensions can be prepaid or metered are a combination of both [ExternallyMetered] and [Entitled].

- **Description** (string) (optional) – Full description of the dimension that will be the long description on the buyer's viewing page.
- **Name** (string) (optional) – Display name for the dimension on the website and customer's bill.
- **Entity** (object) (required) – The named type of entity being created.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0` or `SaaSProduct@1.0`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to UpdateDimensions actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Description	Max length: 1000 Required	400
Key	Max length: 100 Pattern: [A-Za-z0-9_-.]+\$ Required	400
Name	Max length: 5 Required	400
Types (tag)	Required: Min 1 - Max 3 Inputs: Entitled, Metered, ExternallyMetered Required	422

Asynchronous Errors

The following errors are specific to UpdateDimensions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_INPUT	Invalid '%s' field. Remove spaces before trademark symbol.

Error code	Error message
INVALID_INPUT	Invalid '%s' field. Remove unsupported characters %s.
INVALID_DIMENSION	Provide non-empty fields (Key, Types, Name and/or Description) for each dimension.
INVALID_DIMENSION	Cannot update dimension. The field Name has duplicate values '%s' in other dimensions.
INVALID_DIMENSION	Cannot update same dimension with key '%s' and types '%s' multiple times in the same request.
INVALID_DIMENSION	Cannot restrict dimension. The dimension key '%s' with types '%s' does not exist.
INVALID_DIMENSION	Cannot update dimension. The dimension key '%s' is Metered.
INVALID_DIMENSION	Dimension cannot be updated for an already restricted dimension.

Restrict pricing dimensions

You can use the Catalog API to restrict existing pricing dimensions of an AMI or SaaS product in AWS Marketplace.

Each dimension is uniquely identified by the dimension key and dimension types to perform the update. Restricting a dimension doesn't affect any active offer or customers that the original dimension had created.

To restrict pricing dimensions, call the `StartChangeSet` API with the `RestrictDimensions` change type.

Note

Pricing dimension restrictions are only available while the product is in draft state. Changes are not permitted once the product moves to limited or public state.

The following example shows how to restrict the Entitled dimension for a SaaS product.

Request Syntax

```
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"
      }
    }
  ]
}
```

Use the following fields with the RestrictDimensions change type:

- DetailsDocument (array of objects) (required) – Details of the request.
 - Key (string) (required) – Provide key of existing dimension from the product to change description and name on. For RestrictDimensions, this field is only for identifying the dimension to be changed.

- **Types** (array of strings) (required) (also known as **Tags**) – These indicate whether the dimension covers metering, entitlement, or support for external metering. This is not changeable after the dimension is created.
 - ["ExternallyMetered", "Entitled"] – You can only combine these types for SaaS Contract with Consumption pricing where dimensions can be prepaid or metered.
 - ["Metered"] – For hourly pricing dimensions of AMI products. Indicates that Commerce Platform usage types should be created to allow metering to take place for this dimension.
 - ["ExternallyMetered"] – For flexible consumption pricing dimensions (also known as custom metering) of AMI, container, and SaaS products. Indicates that AWS Marketplace Metering Service (MMS) dimensions should be created during publishing to allow sellers to meter through the AWS SDK.
 - ["Entitled"] – For contract pricing dimensions of SaaS Contracts and professional services products. This tag grants rights to use a software or service, sets start and end dates for the usage, and grants usage-discount rights for AMI annual products. Each entitlement is identified by a Dimension Key in AWS Marketplace Entitlement Service for creating and updating the entitlements. The key indicates that entitlements can be granted for the dimension during product and offer publishing.
- **Entity** (object) (required) – The named type of entity being created.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on your product's delivery method (product type): `AmiProduct@1.0` or `SaaSProduct@1.0`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or by calling the [DescribeChangeSet](#) API.

Synchronous Validations

The following schema validations are specific to `RestrictDimensions` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Description	Max length: 1000	400
	Required	
Key	Max length: 100	400
	Pattern: [A-Za-z0-9_.-]+\$	
	Required	
Name	Max length: 5	400
	Required	
Types (tag)	Required: Min 1 - Max 3	400
	Inputs: Entitled, Metered, ExternallyMetered	
	Required	

Asynchronous Errors

The following errors are specific to `RestrictDimensions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` while a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_INPUT	Invalid '%s' field. Remove spaces before trademark symbol.
INVALID_INPUT	Invalid '%s' field. Remove unsupported characters %s.
INVALID_DIMENSION	The dimension key '%s' with types '%s' was already restricted
INVALID_DIMENSION	Cannot restrict dimension. The dimension key '%s' with types '%s' does not exist
INVALID_DIMENSION	Cannot restrict duplicate dimensions.
INVALID_DIMENSION	All Entitled dimensions cannot be restricted. There must be at least one active Entitled dimension.
INVALID_DIMENSION	The dimension key '%s' with types '%s' is associated with another dimension of different types '%s'. Both dimensions of the same key must be restricted at the same time to be valid.

Update targeting configuration

You can use the Catalog API to add AWS account IDs that are allowed to view the AMI, container, ML, or SaaS product in AWS Marketplace before it's moved to a `Public` state by calling the `UpdateTargeting` change type.

Managed Catalog Operations (MCO) accounts are automatically added to the allowed accounts list when new products are created. These MCO accounts are visible to sellers in the AWS Marketplace Management Portal (AMMP) when viewing allowed accounts, and in the `Targeting` section of the `DescribeEntity` API response.

Note

For more information about adding AWS account IDs using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update the allowlist \(preview accounts\)](#)
- Container-based product: [Updating the allowlist of AWS account IDs](#)
- SaaS-based product: [Updating the allowlist of AWS account IDs](#)
- Machine learning product: [Updating the allowlist](#)

To add AWS account IDs that are allowed to view the AMI, container, ML, or SaaS product, call the `StartChangeSet` API operation with the `UpdateTargeting` change type, as shown in the following example.

Request Syntax

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

```
    }  
  ]  
}
```

Use the following the fields with the UpdateTargeting change type.

- Entity (object) (required) – The named type of entity being created.
 - Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
 - Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0, MachineLearningProduct@1.0, or SaaSProduct@1.0.
- DetailsDocument (object) (required) – The details required to run the ChangeSet.
 - PositiveTargeting (object) (optional) – Positive targeting defines the criteria which any buyer's profile should fulfill in order to be allowed to access the offer. This field is optional, but at least one targeting option should be provided when this field is present.
 - BuyerAccounts (array of strings) (optional) – List as an option to allow targeting based on AWS accounts (also known as, Private Offer). If the intention is to not target the offer to an AWS account, this field should be omitted.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

To check request status, use the AWS Marketplace Management Portal or call the [DescribeChangeSet](#) API.

When the request is complete (if the Status is SUCCEEDED), a new ProductId is generated.

Synchronous Validations

The following schema validations are specific to `UpdateTargeting` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Type of targeting	Valid current visibility states	BuyerAccounts (input)	Check
Positive	Public, Limited, or Draft	Array of 12-digit AWS account ID strings. Min size: 0. Max size: 5000.	The input must be different from the current document targeted accounts. Input must be in valid AWS accounts.

Asynchronous Errors

The following errors are specific to `DescribeChangeSet` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_PRODUCT_VISIBILITY	Use an existing Public, Limited or Draft product.
INVALID_AWS_ACCOUNT_IDS	Provide valid AWS account IDs. AWS accounts not found: [x, y, z].
ValidationException	Professional services products do not have allowlists. Unlike other product types, professional services products in the limited state can be extended to any buyer without requiring an allowlist.

Update product visibility

You can use the Catalog API to update the visibility (also known as lifecycle state) of an AMI, container, ML, or SaaS product in AWS Marketplace.

Note

For more information updating product visibility using the AWS Marketplace Management Portal, see the following topics in the *AWS Marketplace Seller Guide*:

- AMI-based product: [Update product visibility](#)
- Container-based product: [Updating product visibility](#)
- SaaS-based product: [Update product visibility](#)
- Machine learning product: [Updating product visibility](#)

Allowed target lifecycle states are Limited, Public, or Restricted.

Limited

The product is complete and has successfully completed the `ReleaseProduct ChangeType`. Sellers can view details of the product in this state. The product is not public. However, sellers can target specific buyers to allow to preview the product.

Public

The product is visible in AWS Marketplace. Buyers can view and subscribe to the product.

Restricted

The product is no longer visible to the public and doesn't accept new subscribers. Existing subscribers can continue using this product until their subscription expires.

Note

The `UpdateVisibility` change type requires a manual review from the AWS Marketplace Seller Operations team, which results in a longer execution time. Use `UpdateVisibility` separately in its own change set.

To update your product's visibility, call the StartChangeSet API operation with the UpdateVisibility change type, as shown in the following example.

Request Syntax

For when TargetVisibility is Public or 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"
      }
    }
  ]
}
```

For when TargetVisibility is Restricted.

```
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"
  }
}
```

Provide information for the fields to add the UpdateVisibility change type.

- Entity (object) (required) – The named type of entity being created.
 - Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
 - Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0, SaaSProduct@1.0, ContainerProduct@1.0, MachineLearningProduct@1.0. For more information, see [Identifier](#).
- DetailsDocument (object) (required) – The details required to run the ChangeSet.
 - TargetVisibility – The intended new visibility of the product.

Possible values: Public, Limited, and Restricted
 - ReplacementProductId (string) (optional) – Replacement product ID for the product to be Restricted. Used to notify current subscribers about the product restriction.

Only accepts Restricted for TargetVisibility.

Synchronous Validations

The following schema validations are specific to UpdateVisibility actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Type of targeting	Valid current states	ReplacementProduct Id (input)	Validation checks
Public	Limited and Restricted	Not allowed	Valid current state

Type of targeting	Valid current states	ReplacementProduct Id (input)	Validation checks
Limited	Public and Restricted	Not allowed	Valid current state
Restricted	Public and Limited	String (Optional)	ReplacementProduct Id must belong to an existing Limited or Public product.

After triggering this change type, it can take up to 37 days to complete. This includes the time the AWS Marketplace Seller Operations Team needs to review, audit, and approve. When restricting a product, you have 24 hours to change your mind, by calling `CancelChangeSet`, before the AWS Marketplace Seller Operations Team begins auditing. For more information, see [CancelChangeSet](#).

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the Status is SUCCEEDED), a new ProductId is generated.

Asynchronous Errors

The following errors are specific to UpdateVisibility actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_PRODUCT_STATE	Use an existing Public, Limited, or Restricted product.
INVALID_TARGET_VISIBILITY	Provide a valid target visibility state: Public, Limited, or Restricted .
EMPTY_TARGET_VISIBILITY	Provide a valid target visibility state: Public, Limited, or Restricted .
INVALID_REPLACEMENT_PRODUCT_ID	Use an existing Public or Limited product as replacement.
INVALID_REPLACEMENT_PRODUCT_ID	Replacement product ID is only valid when restricting a product.
AUDIT_ERROR	Varies based on MCO manual review.
MISSING_SELLER_PROFILE_INFORMATION	Before you can update your product to Public, you must add a public profile to your seller account.

Publish a product

You can use the Catalog API to publish a Draft AMI, container, ML, or SaaS product into Limited state in AWS Marketplace.

Note

For AmiProduct@1.0 and SaaSProduct@1.0, the ReleaseProduct change type must be accompanied by ReleaseOffer change type on the corresponding draft public Offer@1.0 entity created for this product.

To publish a product, call the StartChangeSet API operation with the ReleaseProduct change type, as shown in the following example.

Request Syntax

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

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

Provide information for the fields to add to the ReleaseProduct change type. This change type does not take any parameter payload.

- Entity (object) (required) – The named type of entity being created.
 - Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
 - Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: AmiProduct@1.0, SaaSProduct@1.0, ContainerProduct@1.0, MachineLearningProduct@1.0. For more information, see [Identifier](#).
- DetailsDocument (object) (required) - Must be an empty object. The change type ReleaseProduct doesn't accept any details.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the Status is SUCCEEDED), a new ProductId is generated.

Asynchronous Errors

The following errors are specific to ReleaseProduct actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
VALIDATION_FAILED	Provide Description information.
VALIDATION_FAILED	Provide Versions information.
VALIDATION_FAILED	Provide Dimensions information.
VALIDATION_FAILED	Provide Description PromotionalResources SupportInformation information.

Find your product ID

You must get the product ID for your product before you can modify it with AWS Marketplace Catalog API. There are two ways to find the product ID for server products:

- Open the AWS Marketplace Management Portal and sign in with your seller account. From the **Products** menu, select **Server products**, then choose the product you are interested in. The product ID is listed in the **Product Summary** section.
- Use the [ListEntities](#) action with the EntityType **AmiProduct** or **ContainerProduct**, **SaaSProduct**, **MachineLearningProduct**, or **DataProduct** to get a list of products, including their product IDs, via the Catalog API. ListEntities requires that you do not include the version of the entity type (for example, AmiProduct@1.0).

Note

The product ID is only available after your product has been published and is visible to at least yourself in AWS Marketplace. When you first create your product, it can take several days to be reviewed and fully created. During this time, it will not have a product ID available.

The following topics explain how to find a product by filtering on entity id, product title, last modified date, or visibility.

Topics

- [Find a product based on product title](#)
- [Find a product based on last modified date](#)
- [Find a product based on product visibility](#)
- [Find a product based on product title, last modified date, and product visibility](#)
- [Get additional details about a product](#)

Find a product based on product title

Request

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

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

```
"MaxResults": 10,
"EntityTypeFilters": {
  "AmiProductFilters": {
    "ProductTitle": {
      "WildcardValue": "XYZ"
    }
  }
}
```

Response

```
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": ""
}
```

Find a product based on last modified date

Request

```
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"
          }
        }
      }
    }
  }
}

```

Response

```

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": ""
}

```

Find a product based on product visibility

Request

```
POST /ListEntities HTTP/1.1
```

Content-Type: application/json

```
{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "MaxResults": 10,
  "EntityTypeFilters": {
    "AmiProductFilters": {
      "Visibility": {
        "ValueList": [
          "Public"
        ]
      }
    }
  }
}
```

Response

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": ""
}
```

Find a product based on product title, last modified date, and product visibility

Request

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

Response

```
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": ""
}
```

Get additional details about a product

You can get additional details about the product using the entity id with the DescribeEntity action.

Request

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

Response

```
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",
}
```

```
}
```

Change set status and errors

Making changes to seller products in the AWS Marketplace Catalog API involves creating change sets that describe the changes you want to make, and then using the `StartChangeSet` action to start the changes. The changes from the request can take minutes to hours or longer to complete, depending on the request. The response to this request looks like the following.

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

The change request is added to a queue and processed, including scanning the files and information to ensure that it meets the AWS Marketplace guidelines for products. Depending on the change requests, this process can take a few minutes to days. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action. For more information about change sets, see [Working with change sets](#).

To check the status of your request, use the `DescribeChangeSet` action.

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

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

The result of this call looks like the following (in this case, for adding a new version to a container product).

```
{
  "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": "{\n\"Version\": {\n\"VersionTitle\": \"1.1\", \"ReleaseNotes\": \"Minor bug fix\", \"DeliveryOptions\": [{\n\"DeliveryOptionTitle\": \"EKSDelivery\", \"Details\": {\n\"EcrDeliveryOptionDetails\" : {\n\"ContainerImages\": [\n\"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame1:1.1\"], \"DeploymentResources\": [{\n\"Name\": \"HelmDeploymentTemplate\", \"Url\": \"111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame2:mychart1.1\"}], \"CompatibleServices\": [\n\"EKS\"], \"Description\": \"Sample Description\", \"UsageInstructions\": \"helm pull 111122223333.dkr.ecr.us-east-1.amazonaws.com/sellername/reponame2:mychart1.1\"}]}], {\n\"DeliveryOptionTitle\": \"HelmChartDeliveryOption\", \"Details\": {\n\"HelmDeliveryOptionDetails\": {\n\"CompatibleServices\": [\n\"EKS\", \"EKS-Anywhere\"], \"ContainerImages\": [\n\"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\": [{\n\"Key\": \"HelmKeyName1\", \"DefaultValue\": \"${AWSMP_LICENSE_SECRET}\"}, {\n\"Key\": \"HelmKeyName2\", \"DefaultValue\": \"${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":
  [
  ]
}
]
}

```

The `Status` field shows the current status of the request, in this case, `SUCCEEDED`.

If there are failures, the result can include two types of errors. For most errors, the error message is included directly. However, errors found while scanning the product for security vulnerabilities instead include a URL to a file that lists all of the errors found, in the `ErrorMessage` field. Errors found while scanning have the `ErrorCode` `SCAN_ERROR`.

```

{
  "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"
        }
    ]
}

```

In this example, there is one error directly reported (DUPLICATE_VERSION_TITLE). The other error has a file with error messages (a single SCAN_ERROR can have multiple found errors in the file that is linked).

Note

The link returned in the ErrorMessage is valid for 60 days.

Work with AMI-based products

You can use the AWS Marketplace Catalog API to automate tasks for working with Amazon Machine Image (AMI)-based products.

For information about creating a AMI-based product using the Catalog API, see [Create a product](#).

The following topics describe how to use the Catalog API to perform actions on your AMI-based products:

Topics

- [Add a new version](#)
- [Update version information](#)
- [Restrict a version](#)
- [Update future AWS Region support](#)
- [Add a supported AWS Region](#)
- [Restrict an AWS Region](#)
- [Add a new instance type](#)
- [Restrict an instance type](#)

Add a new version

You can use the Catalog API to add a new version to an existing AMI-based product in AWS Marketplace. For more information about adding new AMI versions to your product using the AWS Marketplace Management Portal, see [Adding a new version](#) in the *AWS Marketplace Seller Guide*.

When you create a request that includes adding a new AMI to AWS Marketplace, the AMI must be copied into the AWS Marketplace system and then scanned for security issues. You must give AWS Marketplace access to the AMI by creating an AWS Identity and Access Management (IAM) role with permissions to perform actions on your AMI. For more information on the required permissions, see [Giving AWS Marketplace access to your AMI](#) in the *AWS Marketplace Seller Guide*.

To add a new version, call the `StartChangeSet` API operation with the `AddDeliveryOptions` change type for AMI-based products, as shown in the following example. To test your API call without actually creating a new version, set the `Intent` parameter to `VALIDATE`. For more information, see [Intent](#).

Note

For AMI products, a version is made up of one or more delivery options. All delivery options in the same version must have the same `AmiSource` object with the same details. All delivery options must be added to a version when the version is created. It is not possible to add delivery options to an existing version.

Request Syntax

```
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"
}

```

The following list provides information about the input fields you use with the `AddDeliveryOptions` change type. All fields are required unless otherwise noted. For more information about these fields, see [Adding a new version](#) in the AWS Marketplace Seller Guide.

- **DeliveryOptions** (array of objects) – A list of delivery option objects, including the details of each. You must include at least one.
 - For *AMI (standalone)* delivery option, specify an object with the following details. You can only have one delivery option with this type.
 - **Details** (object) – Holds the details of this delivery option. Note that this nested details object does not need to be double-escaped.
 - **AmiDeliveryOptionDetails** (object) – Use to provide the details of each AMI delivery option.
 - **AmiSource** (object) – Details about the AMI to be used for the added version.
 - **AmiId** (string) – ID for the source AMI, located in the AWS Region where the API is being called. This must always be US East (N. Virginia) because that is the only region where the Catalog API is available. Must belong to the caller account.

- **AccessRoleArn** (string) – IAM role Amazon Resource Name (ARN) used by AWS Marketplace to access the provided AMI. For details about creating and using this ARN, see [Giving AWS Marketplace access to your AMI](#) in the *AWS Marketplace Seller Guide*.
- **UserName** (string) – Login user name to access the operating system (OS) in the AMI. Typically ec2-user for Linux AMIs or Administrator for Windows.
- **ScanningPort** (integer) – SSH or RDP port used to access the OS. Used for scanning the provided AMI for security vulnerabilities. Defaults to 22.
- **OperatingSystemName** (string) – Name of the operating system displayed to buyers.
- **OperatingSystemVersion** (string) – Operating system version string displayed to buyers.
- **Afids** (array of strings) (optional) – Amazon FPGA Image (AFI) IDs to enable FPGA support for this version. When provided, this enables your product to run on FPGA-enabled F2 instance types. You can provide up to 15 AFI IDs per version. All AFI IDs you provide must originate from the US East (N. Virginia) region, reside within your AWS Marketplace seller account, and the provided IAM access role should have permissions to share this AFI with AWS Marketplace. For more details about the required permissions, see [Giving AWS Marketplace access to your FPGA images](#) in the *AWS Marketplace Seller Guide*.
- **UsageInstructions** (string) – Instructions for using the AMI, or a link to more information about the AMI.
- **AccessEndpointUrl** (object) (optional) – Used to create a path to access the AMI after it is used.
 - **Port** (string) – The port number used to access the service running on the AMI.
 - **Protocol** (string) – The protocol (http or https) used to access the service running on the AMI.
 - **RelativePath** (string) – The path from the web root to access the service running on the AMI (for example /index.html).
- **RecommendedInstanceType** (string) – The instance type that is recommended to run the service with the AMI and is the default for 1-click installs of your service. For a list of instance types, see [Instance types](#) in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.
- **SecurityGroups** (array of objects) – A list of objects representing ingress rules for the automatically created groups for the version.
- **IpProtocol** (string) – The protocol to use (tcp or udp).

- **FromPort** (integer) – The source port.
- **ToPort** (integer) – The destination port.
- **IpRanges** (array of strings) – IP ranges to allow, in CIDR format (in the form xxx.xxx.xxx.xxx/nn, for example, *192.0.2.0/24*).
- For *AMI with CloudFormation* delivery option, specify an object with the following details. You can have up to three delivery options of this type.
 - **DeliveryOptionTitle** (string) – Title for delivery option.
 - **Details** (object) – Holds the details of an AMI delivery option. Note that this nested details object does not need to be double-escaped.
 - **DeploymentTemplateDeliveryOptionDetails** (object) – Use to provide the details of each CFT delivery option.
 - **ShortDescription** (string) – Brief description of your CloudFormation template delivery option.
 - **LongDescription** (string) – Detailed description of your CloudFormation template delivery option.
 - **UsageInstructions** (string) – Instructions for using the AMI, or a link to more information about the AMI.
 - **RecommendedInstanceType** (string) – The instance type that is recommended to run the service with the AMI and is the default for 1-click installs of your service. For a list of instance types, see [Instance types](#) in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.
 - **ArchitectureDiagram** (string) – URL to the location of your architectural diagram in Amazon S3.
 - **Template** (string) – URL to the location of your CloudFormation Template in Amazon S3.
 - **TemplateSources** (array of objects)
 - **ParameterName** (string) – Name of the parameter in the CloudFormation Template that the AMI in this version should be passed in to. For more information, see [Requirements for AMI details](#).
 - **AmiSource** (object)
 - **AmiId** (string) – ID for the source AMI, located in the AWS Region where the API is being called (currently must always be US East (N. Virginia) because that is the only region where the Catalog API is available). Must belong to the caller account.

- **AccessRoleArn** (string) – IAM role Amazon Resource Name (ARN) used by AWS Region to access the provided AMI. For details about creating and using this ARN, see [Giving AWS Marketplace access to your AMI](#) in the *AWS Marketplace Seller Guide*.
- **UserName** (string) – Login user name to access the operating system (OS) in the AMI. Typically ec2-user for Linux AMIs or Administrator for Windows.
- **OperatingSystemName** (string) – Name of the operating system displayed to buyers.
- **OperatingSystemVersion** (string) – Operating system version string displayed to buyers.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about errors in seller product change sets, see [Change set status and errors](#).

When the request is complete, the version is added, and any existing subscribers will receive an email message telling them about the new version. For more information about the process of adding a new version, see [Adding a new version](#) in the *AWS Marketplace Seller Guide*.

Asynchronous Errors

The following errors are specific to `AddDeliveryOptions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is

processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_AMI_NAME	The prefix (x) in the AMI name is invalid. Provide a new name using your company or brand name as a prefix.
INVALID_PRODUCT	Use an existing limited or public product.
DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
INVALID_VERSION_TITLE	Remove unsupported characters: [x, y, z]
INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.
INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	Remove spaces from the beginning of release notes.

Error code	Error message
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
RECOMMENDED_INSTANCE_TYPE_NOT_AVAILABLE	Provide an existing, available instance type.
INVALID_RECOMMENDED_INSTANCE_TYPE	Provide a valid instance type.
INVALID_SECURITY_GROUP	Security group ports must be between 1 and [max].
INVALID_SECURITY_GROUP	Provide a value for CIDR IP ranges.
INVALID_SECURITY_GROUP	Provide security group start port that is not greater than end port.
INVALID_SECURITY_GROUP_PROTOCOL	Security group protocol must either be 'tcp' or 'udp'.
INVALID_CIDR_IP	Provide standard CIDR IP range in form '0.0.0.0/0'.
INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port less than [x].
INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port between 1 and [max].
INVALID_ACCESS_ENDPOINT_PORT	Provide endpoint port.
INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	Remove spaces in the relative path.
INVALID_ACCESS_ENDPOINT_RELATIVE_PATH	Remove preceding '/' from relative path.
INCOMPATIBLE_OPERATING_SYSTEM	Provide operating system name and version that is compatible with instance types: [x]
INCOMPATIBLE_OPERATING_SYSTEM_NAME	Provide name with fewer than (x) characters.
INCOMPATIBLE_OPERATING_SYSTEM_NAME	Provide operating system name that is supported.

Error code	Error message
INCOMPATIBLE_OPERATING_SYSTEM_VERSION	Provide version with fewer than (x) characters.
INVALID_SCANNING_PORT	Provide scanning port between 1 and [max].
INVALID_AMI_ID	Provide valid AMI ID.
EXISTING_AMI_PRODUCT_CODE	Remove product code attached to image X.
INVALID_AMI_ARCHITECTURE	Provide new AMI with architecture [x].
INVALID_AMI_VIRTUALIZATION_TYPE	Provide new AMI with virtualization type [x].
INVALID_AMI_VIRTUALIZATION_TYPE	Provide expected [z] volume on image [x].
INCOMPATIBLE_AMI	Provide new AMI as architecture [x] on [y] is not supported by following instance types: [z]
INCOMPATIBLE_AMI	Provide new AMI as virtualization type [x] on [y] is not supported by following instance types: [z]
INCOMPATIBLE_AMI	Enable ENA support for image x because following instance types require ENA support: [y]
ASSET_NOT_FOUND	Check if [ami-id] exists in us-east-1 Region of [account-id] AWS account and the AccessARN provided [ARN] has permissions to share this AMI with AWS Marketplace.
ASSET_ACCESS_EXCEPTION	Unable to copy AMI [x] into AWS Marketplace account.
SCAN_ERROR	Fix security vulnerability [y] on Image [x].

Error code	Error message
ASSET_NOT_FOUND	Check if [afi-id] exists in us-east-1 Region of [account-id] AWS account and the AccessARN provided [ARN] has permissions to share this AFI with AWS Marketplace.
ASSET_ACCESS_EXCEPTION	Unable to access AFI. Check if AccessARN provided [ARN] has permissions to share this AFI with AWS Marketplace and access role Account is AFI owner.
MISSING_FPGA_INSTANCE_TYPE	No FPGA based instance type found. Use AddInstanceTypes change type to add FPGA based instance type.

You can also create a AMI-based product using the AWS Marketplace Management Portal. For more information, see [AMI products](#) in the *AWS Marketplace Seller Guide*.

For a walk-through showing how to automate updating your AMI-based product, you can also refer to the video, [Automating updates to your product listings in AWS Marketplace with Catalog API](#) (5:08).

Update version information

You can use the Catalog API to update the details of an existing version of your AMI-based product in AWS Marketplace.

Note

For more information about updating version information using the AWS Marketplace Management Portal, see [Updating version information](#) in the *AWS Marketplace Seller Guide*.

You cannot update the AMI for the version. If you need to update the AMI, create a new version instead.

To add a new version, call the `StartChangeSet` API operation with the `UpdateDeliveryOptions` change type, as shown in the following example.

Request Syntax

```
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."
  }
}

```

The following is information about the input fields you provide for adding the UpdateDeliveryOptions change type. For more information about these fields, see [Updating version information](#) in the AWS Marketplace Seller Guide.

- **Entity** (object) (required) – Your AMI-based product.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) (required) – Details of the request. It includes any information about the version of your AMI-based product that you would like to update. The included fields are all optional, but you must include at least one field to update.
 - **Version** (object) – Details about the software version.
 - **ReleaseNotes** (string) – Notes for buyers to tell them about changes from one version to the next.
 - **DeliveryOptions** (array of objects) – List of delivery option objects. Include only the delivery options you want to update.
 - For *AMI (standalone)* delivery option, specify an object with the following details. Include only the fields you want to update. All fields are optional unless otherwise noted.
 - **Id** (string) (required) – Unique identifier for the delivery option (you can get the unique identifier for the delivery option by calling the `DescribeEntity` action on the product you are updating).
 - **Details** (object) – Holds the details of an AMI delivery option. Note that this nested details object does *not* need to be double-escaped.
 - **AmiDeliveryOptionDetails** (object) – The details of one AMI delivery option.
 - **UsageInstructions** (string) – Instructions for using the AMI, or a link to more information about the AMI.

- **AccessEndpointUrl** (object) – Used to create a path to access the AMI after it is used.
- **Port** (string) – The port number used to access the service running on the AMI.
- **Protocol** (string) – The protocol (`http` or `https`) used to access the service running on the AMI.
- **RelativePath** (string) – The path from the web root to access the service running on the AMI (for example `/index.html`).
- **RecommendedInstanceType** (string) – The instance type that is recommended to run the service with the AMI and is the default for 1-click installs of your service.
- **SecurityGroups** (array of objects) – A list of objects representing ingress rules for the automatically created groups for the version:
 - **FromPort** (integer) – The source port.
 - **IpProtocol** (string) – The protocol to use (`tcp` or `udp`).
 - **IpRanges** (array of strings) – IP ranges to allow, in CIDR format (in the form `xxx.xxx.xxx.xxx/nn`, for example, `192.0.2.0/24`).
 - **ToPort** (integer) – The destination port.
- For *AMI with CloudFormation* delivery option, specify an object with the following details. Include only the fields you want to update. All fields are optional unless otherwise noted.
 - **Id** (string) (required) – Unique identifier for the delivery option (you can get the unique identifier for the delivery option by calling the `DescribeEntity` action on the product you are updating).
 - **Details** (object) – Holds the details of an AMI delivery option. Note that this nested details object does *not* need to be double-escaped.
 - **DeploymentTemplateDeliveryOptionDetails** (object) – Use to provide the details of each CFT delivery option.
 - **DeliveryOptionTitle** (string) – Title for delivery option.
 - **ShortDescription** (string) – Brief description of your CloudFormation template delivery option.
 - **LongDescription** (string) – Detailed description of your CloudFormation template delivery option.
 - **UsageInstructions** (string) – Instructions for using the AMI, or a link to more information about the AMI.
 - **RecommendedInstanceType** (string) – The instance type that is recommended to run the service with the AMI and is the default for 1-click installs of your service. For a list

of instance types, see [Instance types](#) in the *Amazon Elastic Compute Cloud User Guide for Linux Instances*.

- **ArchitectureDiagram** (string) – URL to the location of your architectural diagram in Amazon S3.
- **Template** (string) – URL to the location of your CloudFormation Template in Amazon S3.
- **TemplateSources** (array of objects)
 - **ParameterName** (string) – Name of the parameter in the CloudFormation Template that the AMI in this version should be passed in to. For more information, see [Requirements for AMI details](#).
 - **AmiSource** (object)
 - **AmiId** (string) – ID for the source AMI, located in the AWS Region where the API is being called (currently must always be US East (N. Virginia) because that is the only region where the Catalog API is available). Must belong to the caller account.
 - **AccessRoleArn** (string) – IAM role Amazon Resource Name (ARN) used by AWS Region to access the provided AMI. For details about creating and using this ARN, see [Giving AWS Marketplace access to your AMI](#) in the *AWS Marketplace Seller Guide*.
 - **UserName** (string) – Login user name to access the operating system (OS) in the AMI. Typically `ec2-user` for Linux AMIs or `Administrator` for Windows.
 - **OperatingSystemName** (string) – Name of the operating system displayed to buyers.
 - **OperatingSystemVersion** (string) – Operating system version string displayed to buyers.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

Asynchronous Errors

The following errors are specific to UpdateDeliveryOptions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

You can retrieve UpdatePricingTerms actions by using the DescribeChangeSet operation after the change set begins processing. For more information about error details and troubleshooting, refer to [Change set status and errors](#), earlier in this guide.

Error code	Error message
INVALID_PRODUCT	Use an existing limited or public product.
MISSING_DELIVERY_OPTION_IDS	Provide at least one delivery option ID.
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_DELIVERY_OPTIONS	Provide delivery option IDs that belong to the same version.

Restrict a version

You can use the Catalog API to restrict a version of your AMI-based product. This prevents new buyers from using that version, but keeps it available to existing buyers. Restricting all delivery options from a version restricts the version. Restricting one or more, but not all, delivery options from a version restricts just the delivery options in that version. You must always have at least one

unrestricted version of a product available, so you cannot restrict the last publicly available version of a product.

Note

For more information about restricting AMI versions in AWS Marketplace via the AWS Marketplace Management Portal, see [Restricting a version](#) in the *AWS Marketplace Seller Guide*.

To restrict a version, call the `StartChangeSet` API operation with the `RestrictDeliveryOptions` change type, as shown in the following example.

Note

All subscribers can use the current version regardless of the restriction status. AWS Marketplace guidelines require that you continue to offer support to existing buyers for 90 days after restricting the version. Your AMI will be marked as deprecated after the version is restricted. For more information, see [Deprecate an AMI](#) in the *Amazon Elastic Compute Cloud User Guide for Windows Instances*.

Request Syntax

```
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"  
      ]  
    }  
  }  
]
```

The following is information about the input fields you provide for adding the `RestrictDeliveryOptions` change type:

- **Entity** (object) (required) – Your AMI-based product.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) (required) – Details of the request. It includes IDs for the versions of your AMI-based product that you would like to restrict.
 - **DeliveryOptionIds** (array of objects) – List of `DeliveryOption` IDs for the versions that you want to restrict. You can get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the version you are restricting.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Asynchronous Errors

The following errors are specific to `RestrictDeliveryOptions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_PRODUCT	Use an existing public product.
MISSING_DELIVERY_OPTION_IDS	Provide at least one delivery option ID.
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_DELIVERY_OPTION	Provide delivery option IDs that are in a public state. IDs not in public state: [x]
ALL_DELIVERY_OPTIONS_RESTRICTED	Provide fewer delivery options to restrict as at least one must remain in public state.

Update future AWS Region support

You can use the Catalog API to change future AWS Region support preferences for your AMI-based product in AWS Marketplace.

Note

For more information about changing future Region support using the AWS Marketplace Management Portal, see [Update support for future AWS Regions](#) in the *AWS Marketplace Seller Guide*.

Note

The `UpdateFutureRegionSupport` change type is only available on `AmiProduct@1.0`.

To change future AWS Region support preferences, call the `StartChangeSet` API operation with the `UpdateFutureRegionSupport` change type, as shown in the following example.

Request Syntax

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

Provide information for the fields to add the `UpdateFutureRegionSupport` change type:

- **Entity (object) (required)** – Your AMI-based product.
 - **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument (object) (required)** – The details required to execute the `ChangeSet`.
 - **FutureRegionSupport – Object**

- **SupportedRegions** – Single-element array of strings

Element supported values: one of ["All", "US", "None"]

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Asynchronous Errors

The following errors are specific to `UpdateFutureRegionSupport` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
TOO_MANY_REGIONS	Currently, only 1 value is supported for <code>FutureRegionSupport</code> : All, US, or None
INVALID_REGIONS	Requested Regions [a, b, c] are invalid or unavailable. Only supported values are [x, y, z].
INVALID_INPUT	<code>SupportedRegions</code> can't be empty.

Add a supported AWS Region

You can use the Catalog API to add new supported AWS Regions for your AMI-based product in AWS Marketplace.

Note

For more information about adding new supported Regions using the AWS Marketplace Management Portal, see [Add an AWS Region](#) in the *AWS Marketplace Seller Guide*.

Note

The AddRegions change type is only available on AmiProduct@1.0.

To add new supported Regions, call the StartChangeSet API operation with the AddRegions change type, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

```
}
```

Provide information for the fields to add the AddRegions change type.

- Entity (object) (required) – Your AMI-based product.
 - Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
 - Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.

- **Example**

<caption>DetailsDocument (object) (required) – The details required to execute the ChangeSet.</caption>

- Regions: Array of strings

Element supported values: Valid AWS Region code strings.

For example, `["us-east-1"]`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

- If the change set execution status is `SUCCEEDED`: A new Entity Identifier (or `EntityId`) is generated. You can use the [DescribeEntity](#) API operation on the product entity to check the result.

- If the change set execution status is `CLIENT_ERROR`: The `DescribeChangeSet` response gives the details of the error, as well as corresponding actions to take to fix the error.

Asynchronous Errors

The following errors are specific to `AddRegions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_REGIONS	Requested regions [a, b, c] are invalid or unavailable. Only supported values are [x, y, z].
INVALID_INPUT	Regions can't be empty.

Restrict an AWS Region

You can use the Catalog API to restrict previously supported AWS Regions for your AMI-based product in AWS Marketplace.

Note

For more information about restricting previously supported Regions using the AWS Marketplace Management Portal, see [Restrict an AWS Region](#) in the *AWS Marketplace Seller Guide*.

Note

The `RestrictRegions` change type is only available on `AmiProduct@1.0`.

To restrict previously supported Regions, call the `StartChangeSet` API operation with the `RestrictRegions` change type, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the AddRegions change type.

- **Entity** (object) (required) – Your AMI-based product.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) (required) – The details required to execute the ChangeSet.
 - **Regions** – Array of strings

Element supported values: Valid AWS Region code strings, such as "us-east-1".

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

- If the change set execution status is SUCCEEDED – A new Entity Identifier (or EntityId) is generated. You can use DescribeEntity on the product entity to check the result. For more information, see [DescribeEntity](#).
- If the change set execution status is CLIENT_ERROR: The DescribeChangeSet response gives the details of the error, as well as corresponding actions to take to fix the error.

Asynchronous Errors

The following errors are specific to AddRegions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_REGIONS	Requested regions [a, b, c] are invalid or unavailable. Only supported values are [x, y, z].
INVALID_INPUT	Regions can't be empty.

Add a new instance type

You can use the Catalog API to add new instance types for your AMI-based product in AWS Marketplace.

Note

For more information about adding instance types using the AWS Marketplace Management Portal, see [Add an instance](#) in the *AWS Marketplace Seller Guide*.

AddInstanceTypes will add new instance types to existing products and newly created products when creating a product. The change type will update all versions in product document with a new instance type.

Note

The AddInstanceTypes change type is only available on AmiProduct@1.0.

When adding a restricted instance type, the instance type can be removed from the restricted list and added to the available instance type list. This gives sellers more control to change their product restriction. The instance type list is interchangeable and not a permanent restricted status for a product.

For internally metered products, sellers need to call separate change types AddDimensions and UpdatePricingTerms to update pricing for the instance type.

To add new instance types, call the StartChangeSet API operation with the RestrictRegions change type, as shown in the following example.

Request Syntax

Only AddInstanceTypes change type is shown below. Although internally metered AMI sellers are required to call AddInstanceTypes and UpdatePricingTerms change types for their AMI.

```
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"
        ]
    }
}
]
}

```

Provide information for the fields to add the AddInstanceTypes change type:

- **Entity** (object) (required) – Your AMI-based product.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) (required) – The details required to execute the ChangeSet, in this case InstanceTypes.
 - **InstanceTypes** (array of strings) (required) – List of InstanceTypes to add to the product. These instances will be added to the existing InstanceTypes.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

```

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

```

The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to AddInstanceTypes actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
InstanceTypes	Required	422
InstanceTypes	Must not be empty	422
InstanceTypes	Entries must be between 1 to 24 characters long. Must match <code>^[A-Za-z0-9_-.]+\$</code>	422
InstanceTypes	Entries must be unique	422
InstanceTypes	Must not be more than 1500 entries	422
An unknown property	No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to AddInstanceTypes actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_INSTANCE_TYPES	The following instance types are not valid: [x]

Error code	Error message
INVALID_INSTANCE_VIRTUALIZATION	Virtualization of the product is [x]. It is not supported by the following instance types: [x]
INVALID_AMI_ARCHITECTURE	CPU architecture of the product is '%s'." + "It is not supported by the following instance types: [x]
INCOMPATIBLE_OPERATING_SYSTEM	The instance types are incompatible with the OS defined in the product. Provide instance types that are compatible with the OS defined in the product.
INVALID_PRODUCT_TYPE	Use an existing single AMI product.
INVALID_ENA_SETTING	The product does not support ENA. ENA support is required by the following instance types: [x]
INVALID_DIMENSIONS	No internally metered dimensions found for instance types: [x]
MISSING_DIMENSIONS	No dimensions found for the product. AddDimensions is required before AddInstanceTypes.
UPDATE_PRICING_REQUIRED	UpdatePricingTerms change type is required when internally metered dimensions are available on the product.

Restrict an instance type

You can use the Catalog API to limit or restrict the instance types available for your AMI-based product in AWS Marketplace.

Note

For more information about limiting or restrict the instance types available using the AWS Marketplace Management Portal, see [Restrict an instance](#) in the *AWS Marketplace Seller Guide*.

Existing subscribers won't be impacted by this change and they are able to use the restricted instance types. However, no new buyers will be able to use restricted instance types. To stop current instance types subscriptions (once instance types are restricted), you must contact the AWS Marketplace Seller Operations Team.

`RestrictInstanceTypes` restricts instance types to all the versions in the product document. In the `AddInstanceTypes` change type, you are updating all versions of the product. You won't be able to restrict the recommended instance types. The recommended instance type is at the version level, so it's possible that the seller won't be able to restrict multiple instance types.

For an internally metered product, you need to call separate change types when calling `RestrictDimensions`. This prevents new offers being created for the restricted instance types.

Note

The `RestrictInstanceTypes` change type is only available on `AmiProduct@1.0`.

To limit or restrict the instance types available for your AMI-based product, call the `StartChangeSet` API operation with the `RestrictInstanceTypes` change type, as shown in the following example.

Request Syntax

```
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"
        ]
    }
}
]
}

```

Provide information for the fields to add the RestrictInstanceTypes change type.

- **Entity** (object) (required) – Your AMI-based product.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: `AmiProduct@1.0`.
- **DetailsDocument** (object) (required) – The details required to execute the ChangeSet, in this case InstanceTypes.
 - **InstanceTypes** (array of strings) (required) – List of InstanceTypes to restrict to the product. These instances are added to the current (or if there are no existing instance types, it will add) to restricted InstanceTypes.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

```

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

```


The change request is added to a queue and processed. This included validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `RestrictInstanceTypes` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
InstanceTypes	Required	422
InstanceTypes	Must not be empty	422
InstanceTypes	Entries must be between 1 to 24 characters long. Must match <code>^[A-Za-z0-9_-.]+\$</code>	422
InstanceTypes	Entries must be unique	422
InstanceTypes	Must not be more than 1500 entries	422
An unknown property	No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to `RestrictInstanceTypes` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_INSTANCE_TYPES	The following instance types are not valid: [x]

Error code	Error message
INVALID_PRODUCT_TYPE	Use an existing single AMI product.
DUPLICATE_INSTANCE_TYPE	Provide instance types with no duplicates.
UNAVAILABLE_INSTANCE_TYPE	Provide an available instance type.
RECOMMENDED_INSTANCE_TYPE_RESTRICTED	The following instance types cannot be restricted. Recommended instance type must be changed to a different one before being restricted. Delivery Options Id [X] Instance Type[X]
DIMENSIONS_NOT_RESTRICTED	Restrict dimensions before restricting internally metered instance types: [x]
REGION_NO_INSTANCES	Your restricted instance types would cause product launch failure in region: X. Consider restricting fewer instances.
INCOMPATIBLE_RESTRICTION	Your restricted instance types would cause product with no FPGA based instance types for AFIs. Product needs to support at least one FPGA based instance type.

Work with EC2 Image Builder component products

As an AWS Marketplace seller, you can list AMI-based products delivered to AWS Marketplace buyers using EC2 Image Builder components. To create your component and publish an AWS Marketplace listing, proceed sequentially through the following sections.

Topics

- [Building and testing your Image Builder component](#)
- [Copying the component ARN](#)
- [Creating AWS Marketplace IAM policies](#)
- [Creating the AWS Marketplace IAM role](#)

- [Prepare your Image Builder component listing](#)
- [Publishing your Image Builder component product listing](#)
- [Updating Image Builder component product information](#)
- [Adding a new version to an existing Image Builder component product](#)
- [Updating information about an existing version](#)
- [Restricting an Image Builder component product version](#)
- [Monitoring a change set](#)
- [Securing software downloads](#)

Building and testing your Image Builder component

Build and test your component on Image Builder. For instructions, refer to [Develop custom components for your Image Builder image](#) in the *Image Builder User Guide*. When creating your component using Image Builder, ensure that you do the following:

- The component and all of its underlying dependencies, such as an Amazon Simple Storage Service (Amazon S3) bucket, secrets, or parameters, must be created in the US East (N. Virginia) (us-east-1) AWS Region.
- Include supported architecture and any software dependencies in the component description.
- Test your component in your AWS account by creating an [image pipeline](#) and deploying the AMI created by the pipeline.
- If your component contains instructions to copy binaries, packages, or files from an S3 bucket, use the S3Download action module. In the S3Download module, for source, enter the static location of your file in the S3 bucket. The following example copies a binary from an S3 bucket as part of the component installation.

```
- name: DownloadMyFile
  action: S3Download
  inputs:
    - source: s3://amzn-s3-demo-source-bucket/path/to/package.zip
      destination: C:\myfolder\package.zip
```

- Components can ingest files of up to 2 GB when using the S3Download action.
- If your component uses [parameters](#), ensure that all parameters have default values. For example, if you have a parameter named `region`, ensure that you have a valid default value such as `us-`

east -1. These default values are for AWS Marketplace processing and testing. Testing may fail if you do not include default values.

- If your component uses AWS Secrets Manager, Parameter Store, or a capability of AWS Systems Manager to store parameters, do the following:
 - To retrieve values as a step in your component, embed AWS Command Line Interface commands in your YAML configuration file.
 - Create a corresponding entry in Secrets Manager or Parameter Store in your AWS account. Use the default key and provide a valid value that will assist in building the component during the AWS Marketplace testing process. For example, say you have a parameter called `saas_token` with a default value of `token` that uses Parameter Store. In this case, create a key-value pair in Parameter Store. Use `token` as the key. For the value, enter a valid SaaS token for your application.

Note that values stored in your AWS Marketplace seller account will only be used for AWS Marketplace testing purposes. These values will not be shared with buyers.

- AWS Marketplace automatically generates Amazon Machine Images (AMIs) for your component across all compatible operating system versions you choose during the component creation process. When building your component, choose at least one compatible operating system version. Validate your component's compatibility with all chosen operating system versions by using EC2 Image Builder pipelines to create and test AMIs.

Copying the component ARN

After creating and testing the component on Image Builder, copy and save the component ARN. You will use the ARN when you publish the product listing using the AWS Marketplace Catalog API.

To copy the Image Builder component ARN

1. Sign in to the AWS Management Console and open the Image Builder console at <https://console.aws.amazon.com/imagebuilder/>.
2. In the left navigation bar, under **Saved resources**, choose **Components**.
3. On the **Components** page, for **Filter owner**, choose **Owned by me**.
4. Choose the component name.
5. On the component detail page, in the **Summary** section, copy the ARN.

Creating AWS Marketplace IAM policies

Create the following IAM policies to grant AWS Marketplace access to your Image Builder component and related resources like Amazon S3 buckets and secrets. Use the example policies provided. You attach these policies to an [AWS Marketplace IAM role](#). For help creating policies, see [Creating policies using the JSON editor](#) in the *IAM User Guide*.

- Image Builder get-component policy, to allow AWS Marketplace to access your component on Image Builder. This policy is required. Name the policy mp_ib_ingest.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "imagebuilder:GetComponent",
      "Resource": "*"
    }
  ]
}
```

- Amazon S3 read-access policy, to allow AWS Marketplace to retrieve binaries from an S3 bucket. This policy is only required if your component uses the S3Download action module and stores associated binaries in an S3 bucket. Name the policy mp_ib_s3_read_only.

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/*"
      ]
    }
  ]
}

```

- Secrets Manager read-access policy, to allow AWS Marketplace to retrieve secrets stored in Secrets Manager. This policy is only required if your component uses Secrets Manager to store secrets. Name the policy `mp_ib_sm_read_only`. To restrict the policy to just your secret, replace the `*` in the Resource section with your secret.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}

```

- Parameter Store read-access policy, to allow AWS Marketplace to retrieve secrets stored in Parameter Store. This policy is only required if your component uses Parameter Store to store secrets. Name the policy `mp_ib_ssm_parameter_read_only`. To restrict the policy to just your secret, replace the `*` in the Resource section with your secret.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Action": [
        "ssm:getParameter"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Creating the AWS Marketplace IAM role

Use the following procedure to create an AWS Marketplace IAM role with policies to grant AWS Marketplace access to your component and its dependencies.

To create the AWS Marketplace IAM role

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the left navigation bar, choose **Roles**.
3. Choose **Create role**.
4. Select **Custom trust policy**.
5. Enter the following statement:

JSON

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

```
        "Sid": "Statement1",
        "Effect": "Allow",
        "Principal": {
            "Service": "assets.marketplace.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
    }
}
```

6. Choose **Next**.
7. Add the Image Builder get-component policy you created previously. The get-component policy is required. Add the relevant policies for S3, Secrets Manager, and Parameter Store if your component uses these AWS services.
8. Choose **Next**.
9. Enter a role name, such as MPEC2IBIngestion.
10. Choose **Create role**.

Copy AWS Marketplace IAM role ARN

After creating the AWS Marketplace IAM role, copy and save the role ARN. You will use the ARN when publishing the listing using the AWS Marketplace Catalog API.

To copy the AWS Marketplace IAM role ARN

1. In the IAM console, in the left navigation bar, choose **Roles**.
2. Choose the AWS Marketplace IAM role you created previously, such as MPEC2IBIngestion.
3. On the role detail page, in the **Summary** section, copy the ARN.

Prepare your Image Builder component listing

Before publishing your AWS Marketplace listing, ensure that you have the following information ready:

- **Product metadata** – Metadata includes the product logo, product title, End User License agreement, supported instance types, and AWS Region.
- **Pricing information** – You can offer your product for free, at an hourly rate, or at an hourly rate with an initial free trial period. Bring your own license (BYOL) is not supported.

- **Component details** – Details include the component Amazon Resource Number (ARN), usage details, and AWS Identity and Access Management (IAM) role that AWS Marketplace will assume to process your component.

Publishing your Image Builder component product listing

This topic contains instructions to publish your EC2 Image Builder component listing on AWS Marketplace using the AWS Marketplace Catalog API.

Prerequisites

Ensure that you have the following before publishing your Image Builder component product listing:

- Registration as a seller in AWS Marketplace. For more information, refer to [Register as an AWS Marketplace seller](#).
- An IAM user with `AWSMarketplaceSellerFullAccess` permission.
- A publicly accessible Amazon Simple Storage Service (Amazon S3) bucket to host your company logo and EULA, if used in your component. You will enter the URL for the S3 bucket in your `ChangeSet` JSON file.
- AWS Command Line Interface (AWS CLI). For more information, refer to [What is the AWS Command Line Interface?](#) in the *AWS Command Line Interface User Guide*.

Creating an Image Builder component product on AWS Marketplace

To create an EC2 Image Builder component product on AWS Marketplace using the Catalog API, refer to [Create a product](#).

Updating Image Builder component product information

You can update information about a Image Builder component product on the AWS Marketplace Management Portal.

To update Image Builder component product information

1. Open the [AWS Marketplace Management Portal](#) and sign in to your seller account.
2. On the **Products** menu, choose **Server**.

3. On the **Server products** page, select the product.
4. On the product detail page, on the **Request changes** menu, choose the item that corresponds to the information you want to update.
5. After submitting any changes, the request will appear on the **Requests** tab in "Under review" status, and will change to "Succeeded" once completed.

Adding a new version to an existing Image Builder component product

You can add a new version to an Image Builder component product on AWS Marketplace using the AWS Marketplace Catalog API.

To add a new version

1. From the AWS Marketplace Management Portal, obtain the Product ID.
 - a. Open the [AWS Marketplace Management Portal](#) and sign in to your seller account.
 - b. On the **Products** menu, choose **Server**.
 - c. On the **Server products** page, select the product.
 - d. In the **Product summary**, copy the **Product ID**.
2. Using the following code example, create a changeset file in JSON format. In the example, replace *your-product-ID* with the product ID you obtained in step 1. Replace *new-version-name* with your version title. Replace *new-delivery-option-title* with your delivery option title.

```
[
  {
    "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. Save the changeset file with the name `addIBversion.json`.
4. In your terminal or AWS CloudShell, run the following command:

```
aws marketplace-catalog start-change-set --catalog AWSMarketplace --region us-
east-1 --change-set file://addIBversion.json
```

The `start-change-set` command will return a `ChangeSetId` value. To monitor a change set, see [Monitoring a change set](#).

Asynchronous Errors

The following errors are specific to `AddDeliveryOptions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Note

The following error codes are specific to the Image Builder delivery method. For existing error messages on fields such as `Usage Instructions`, `Recommended Instance type`, and `AccessRoleArn`, see [Add a new version](#).

Error code	Error message
ASSET_NOT_FOUND	Check if [component arn] exists in [account-id] AWS account and the AccessARN provided [ARN] has permissions to share this Image Builder component with AWS Marketplace.
ASSET_NOT_FOUND	The specified S3 link [x] within the component[x] does not exist. Provide a valid S3 link within the component.
DUPLICATE_COMPONENT_NAME	The component name [x] already exists within the same version. Provide a different component name.
DUPLICATE_COMPONENT_NAME	The component name [x] you provided is already in use by a different product. Provide a different component name.
DUPLICATE_COMPONENT_ARN	The component Arn [x] must be different from Component Arn used in other delivery options of the same version of this product.
DUPLICATE_COMPONENT_ARN	The component ARN [x] must be different from component ARN used in other versions of this product.
SCAN_ERROR	Fix security vulnerability [y] on Component [x].
ASSET_ACCESS_EXCEPTION	We don't have the necessary permissions to read the contents from the specified S3 bucket[x]. Ensure the S3 bucket has the appropriate read permissions.
ASSET_ACCESS_EXCEPTION	We don't have the necessary permissions to read the contents of the specified component ARN[x]. Ensure the component ARN has the appropriate read permissions.
ASSET_ACCESS_EXCEPTION	We don't have the necessary permissions to read the contents of the specified package URI [x]. Ensure the package URI has the appropriate read permissions.
ASSET_ACCESS_EXCEPTION	Failed to read from the HTTP source [x]. Verify that the provided HTTP source is correct.

Error code	Error message
INVALID_IMAGE_BUILDER_COMPONENT_PACKAGE	Component assets exceed the size limit for ingestion. Reduce the size of S3/web downloads or eliminate unnecessary downloads to proceed.
INVALID_IMAGE_BUILDER_COMPONENT_PACKAGE	Component assets are taking too long to ingest. Ensure that your network connection is stable and has adequate bandwidth .
TOO_MANY_IMAGE_BUILDER_COMPONENTS	Max of 5 unique component names are supported per product, restrict the additional delivery option.
INCOMPLETE_SELLER_PUBLIC_PROFILE	Your seller public profile is not complete. Complete your public profile before adding versions to the product.
INVALID_DESCRIPTION	The description is missing. Provide a description with fewer than 1024 characters in component [x].
INVALID_COMPONENT_NAME	Provide a component name with fewer than [x] characters.
INVALID_SUPPORTED_OS_VERSION	The OS version is missing. Provide a valid supported OS version in component [x].
INVALID_PATH_FORMAT	Step [STEP_NAME] in phase [PHASE_NAME] is not allowed to use S3 or web URLs for the InstallMSI/UninstallMSI actions. Ensure that the path specified is a valid local path accessible from the system executing the action.
INCOMPATIBLE_OS_TYPE	The specified component OS type[x] is not compatible with the platform of the base image. Provide an OS type that is compatible with the base image.
ASSET_ACCESS_EXCEPTION	We don't have the necessary permissions to read the contents from the specified SSM parameter [x]. Ensure that the IAM access role provided in the API input has the required read permissions.

Error code	Error message
ASSET_ACCESS_EXCEPTION	We don't have the necessary permissions to read the contents from the specified SecretsManager secret [x]. Ensure that the IAM access role provided in the API input has the required read permissions.
INVALID_IB_COMPONENT_BUILD_VERSION	Component ARN [x] has a build version of [y]. AWS Marketplace only supports build version 1. Create a new EC2 Image Builder version with Build versions 1, and try again on AWS Marketplace
INVALID_IB_COMPONENT_PARAMETER	Unable to parse the SSM parameter in input [x] for component [y].
INVALID_IB_COMPONENT_PARAMETER	Unable to parse the SecretsManager secret in input [x] for component [y].
MISSING_IB_COMPONENT_PARAMETER	Parameter not found in component [x] for input [y]
INVALID_IB_COMPONENT_PARAMETER	Provide a default value for parameter [x] in component [y].
FAILED_LAUNCH_TEST	The launch test for component [x] has failed. Error message: [z]
SSHAAuthFailedForUserAndKeypair	Unable to log into the instance with OS default username [X].
INVALID_IB_COMPONENT	A component build phase is required. Add a valid build phase to component [x]
DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
INVALID_VERSION_TITLE	Remove unsupported characters: [x, y, z]
INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.

Error code	Error message
INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.
INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
DUPLICATE_DELIVERY_OPTION_TITLES	Provide unique delivery option title.
INVALID_DELIVERY_OPTION_TITLES	Delivery option title already exists, retry with a different title.

Updating information about an existing version

To update information about an existing version

1. From the AWS Marketplace Management Portal, obtain the Product ID.
 - a. Open the [AWS Marketplace Management Portal](#) and sign in to your seller account.
 - b. On the **Products** menu, choose **Server**.
 - c. On the **Server products** page, select the product.
 - d. In the **Product summary**, copy the **Product ID**.

2. In your terminal, run the following command. In the command, replace *your-product-ID* with the ID you obtained in step 1.

```
aws marketplace-catalog describe-entity --catalog AWSMarketplace --region us-east-1
--entity-id 'your-product-ID'
```

3. In the output returned, go to the DetailsDocument, Versions section. Copy the DeliveryOptions, Id value for the version you want to update.
4. Using the following code example, create a changeset file in JSON format. In the example, replace *your-product-ID* with the product ID you obtained in step 1. Replace *your-release-notes* with your release notes. Replace *your-delivery-option-ID* with the ID of the delivery option you obtained in step 3. Replace *your-usage-instructions* with your usage instructions.

```
[
  {
    "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. Save the changeset file with the name `updateVersionInfo.json`.
6. In your terminal or AWS CloudShell, run the following command:


```
aws marketplace-catalog start-change-set --catalog AWSMarketplace --region us-east-1 --change-set file://updateVersionInfo.json
```

The `start-change-set` command will return a `ChangeSetId` value. To monitor a change set, see [Monitoring a change set](#).

Restricting an Image Builder component product version

Restricting a version makes it unavailable to buyers. You can restrict a version of your Image Builder component product on AWS Marketplace using the AWS Marketplace Catalog API. You must keep at least one version of your product unrestricted on AWS Marketplace. You can't restrict access to the only public version.

To update information about an existing version

1. From the AWS Marketplace Management Portal, obtain the Product ID.
 - a. Open the [AWS Marketplace Management Portal](#) and sign in to your seller account.
 - b. On the **Products** menu, choose **Server**.
 - c. On the **Server products** page, select the product.
 - d. In the **Product summary**, copy the **Product ID**.
2. In your terminal, run the following command. In the command, replace *your-product-ID* with the ID you obtained in step 1.

```
aws marketplace-catalog describe-entity --catalog AWSMarketplace --region us-east-1 --entity-id 'your-product-ID'
```

3. In the output returned, go to the `DetailsDocument`, `Versions` section. Copy the `DeliveryOptions`, `Id` value for the version you want to update.
4. Using the following code example, create a changeset file in JSON format. In the example, replace *your-product-ID* with the product ID you obtained in step 1. Replace *your-delivery-option-ID* with the ID of the delivery option you obtained in step 3.

```
[
  {
    "ChangeType": "RestrictDeliveryOptions",
    "Entity": {
```

```
        "Identifier": "your-product-ID",
        "Type": "AmiProduct@1.0"
    },
    "DetailsDocument": {
        "DeliveryOptionIds": [
            "your-delivery-option-ID"
        ]
    }
}
```

5. Save the changeset file with the name `restrictec2ibversion.json`.
6. In your terminal or AWS CloudShell, run the following command:

```
aws marketplace-catalog start-change-set --catalog AWSMarketplace --region us-east-1 --change-set file://restrictec2ibversion.json
```

The `start-change-set` command will return a `ChangeSetId` value. To monitor a change set, see [Monitoring a change set](#).

Monitoring a change set

The `start-change-set` command will return a `ChangeSetId` value. You can monitor change set progress in the following ways:

- Run the following command in your terminal. In the command, replace *changeset-ID* with the `ChangeSetId` value returned by the `start-change-set` command.

```
aws marketplace-catalog describe-change-set --catalog AWSMarketplace --change-set-id changesetID
```

- View the request status on the **Requests** tab in the [AWS Marketplace Management Portal](#).

Securing software downloads

To safeguard ISV software intellectual property and ensure stable, consistent software delivery to AWS Marketplace buyers, AWS Marketplace automatically parses `S3Download` and `WebDownload` action modules in your component. The referenced files are then securely stored in a private Amazon S3 bucket managed by AWS Marketplace. To opt out of this ingestion process and

manage software downloads independently, run bash scripts that use the `wget` or `curl` download commands.

Work with container-based products using the AWS Marketplace APIs

You can use the AWS Marketplace Catalog API to automate tasks for working with container-based products.

For information about creating a container-based product using the Catalog API, see [Create a product](#).

The following topics describe how to use the Catalog API to perform actions on your container-based products:

Topics

- [Add a new version](#)
- [Update the visibility for an Amazon EKS add-on](#)
- [Create repositories and resources](#)
- [Update version information](#)
- [Restrict a version](#)

Add a new version

If you already have a container-based product in AWS Marketplace, you can use the AWS Marketplace Catalog API to add a new version. This requires that you have already created repositories in AWS Marketplace for each container image or artifact that is part of your product, and that you can copy them from your local Docker and Helm files.

Note

For details about creating a container-based product using the AWS Marketplace Management Portal, see [Getting started with container products](#) in the *AWS Marketplace Seller Guide*.

For details about adding a new version, including creating repositories and building Docker and Helm files into those repositories, by using the AWS Marketplace Management Portal, see [Add a new version of your product](#) in the *AWS Marketplace Seller Guide*.

If you have not already created new repositories, you can create them using the Catalog API, see [Create repositories and resources](#).

To add a new version, call the `StartChangeSet` API operation with the `AddDeliveryOptions` change type, as shown in the following example.

Note

A version of a container-based product is made up of one or more delivery options. For example, you might have two delivery options, one that works with a noSQL database, and another that works with MySQL, so that your users can choose how they want to work with your product. You create the version of your product and add multiple delivery options in a single request with `AddDeliveryOptions`.

Container Image Delivery Request Syntax

```
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 Runtime delivery request syntax

```

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 Chart Delivery Request Syntax

```

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 Add-On Delivery Request Syntax


```

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"
}
]
}

```

Provide information for the fields to add the AddDeliveryOptions change type:

- Entity (object) (required) – Your container-based product.
- Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
- Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: ContainerProduct@1.0.
- DetailsDocument (object) (required) – Details of the request. It includes all the information about the version that you are adding. This field is a string field.
- Version (object) – Details about the version that you are adding to your product.
 - VersionTitle (string) – The title of the version that you are creating. Typically this is a description of the version, like **Version 1.1** or simply **1.1**. Your buyers will be able to choose the version to deploy from a list of version titles.
 - ReleaseNotes (string) – The detailed notes about this version. Must be less than 30,000 characters.
- DeliveryOptions (array of objects) – An array of delivery options, where each is a method of delivery for your product version. For example, if you have one delivery option for Amazon Elastic Container Service (Amazon ECS) and another for Amazon Elastic Kubernetes Service (Amazon EKS), you must have two delivery options.

- `DeliveryOptionTitle` (string) – A short description that helps your buyer to choose between your delivery options.
- `Details` (object) – The resources used for this delivery option. This is a details field within the details field. You do not need to doubly escape characters in this field.
- `AddOnName` – A unique add-on name that buyers will see in the Amazon EKS catalog. This name will add a prefix later using `SellerAlias`. For example, where `isv-alias_` is the ISV provided add-on name.
- `AddOnType` – The type of add-on chosen from a list of supported values from Amazon EKS: `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` – A semantic version so that buyer can choose a specific version of `AddOn` they need to install or upgrade.
- `CompatibleKubernetesVersions` – The Amazon EKS Kubernetes versions that this software is compatible with.
- `CompatibleServices` (array of strings) – An array of services that the release is compatible with. Valid options: `ECS`, `EKS`, `ECS-Anywhere`, `EKS-Anywhere`, and `Bedrock-AgentCore`.
- `ContainerImages` (array of strings) – An array of container image URLs used by this version. The path will be the repository that you have uploaded the image to, with the tag for the image used by this version. The list must include all needed images, even images that have not changed from previous versions. See the next section for information about creating repositories using the Catalog API.
- `Description` (string) – A longer description of the delivery option to give details to your buyer. You can also include a link to more instructions provided elsewhere.
- `EcrDeliveryOptionDetails` – `AgenticType` – The type of runtime agent. Valid options: `AGENT`, `MCP_SERVER`, or `A2A_SERVER`.
- `EcrDeliveryOptionDetails` – `EnvironmentVariables` – List of environment variables that are required by the `AgentCore` Runtime container and that will be pre-populated for buyers upon deployment. For each variable, provide an object with the *name* as expected by your container, a *description*, and an optional *defaultValue*. For variables such as credentials or API keys that are unique, do not provide a default value. You can use the description to specify details about the variable as well as possible values.

All of the provided variables with their default values will be pre-populated when buyers launch your product.

- `EcrDeliveryOptionDetails` – `DeploymentResources` (array of objects)
– An array of other resources needed for the version, such as Helm charts. Each resource includes a `Name` to describe it, and a `URL` that points at the resource.
- `EnvironmentOverrideParameters` – List of system parameters to be used by the add-on. Some of the ISV provided AddOn (HelmChart) might require configurations with information derived from the Amazon EKS execution environment state (/system information). For example, `EksClusterRegion`, `EKSClusterName`, and others. You can avoid additional actions from Buyer by dynamically substituting these values at Amazon EKS AddOn launch. Amazon EKS System already supports automatic substitutions of system param for add-ons. AWS Marketplace ISV experience can be extended to collect this params which would require substitution.

The generic system information to be substituted can be indicated by providing a AWS Marketplace specified constant following convention similar to Helm substitution. The supported values are `${AWS_REGION}` and `${AWS_EKS_CLUSTER_NAME}`.

```
"EnvironmentOverrideParameters" : [ {  
  "Key" : "my-field.region"  
  "Value" : "${AWS_REGION}"  
},  
{  
  "Key" : "my-second-field"  
  "Value" : "${AWS_EKS_CLUSTER_NAME}"  
},
```

- `HelmDeliveryOptionDetails` – `HelmChartUri` (string) – The URL to the Helm chart hosted in Amazon ECR that the buyer will install to launch the software.
- `HelmDeliveryOptionDetails` – `MarketplaceServiceAccountName` (string) – *Optional* – The name of the Kubernetes service account. The service account will be used to connect to AWS Identity and Access Management (IAM) for permissions to call AWS services.
- `HelmDeliveryOptionDetails` – `ReleaseName` (string) – *Optional* – The name for the Helm release provided to the `helm install` command that buyers use to launch

the software. If not included, Helm will provide an automatically generated release name for you.

- `HelmDeliveryOptionDetails` - `Namespace` (string) – *Optional* – The Kubernetes namespace where the Helm chart will be installed.
- `HelmDeliveryOptionDetails` - `OverrideParameters` (array of objects) – Parameters that will be used in the Helm commands that launch the application. Buyers can override the default values.

Note

For Amazon EKS Anywhere products, provide at least 1 override parameter for the license secret. Provide `DefaultValue` of `"${AWSMP_LICENSE_SECRET}"`. For paid products, provide at least 1 override parameter for service account configuration. Provide `DefaultValue` of `"${AWSMP_SERVICE_ACCOUNT}"`.

- `Key` (string) – The key for the parameter in dot notation (override.example.key).
- `DefaultValue` (string) – The default value for this override parameter.
- `Namespace` – The ISV provided namespace for add-on installation.
- `SupportedArchitectures` – The list of supported architectures, like amd64 and arm64.
- `UsageInstructions` (string) – Provide instructions about the usage for this delivery option. Can be up to 4,000 characters.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed, including scanning the container images and other information to ensure that it meets the [AWS Marketplace guidelines for container](#)

[products](#). This process can take a few minutes to hours, depending on the number and size of your containers.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

Asynchronous Errors

The following errors are specific to AddDeliveryOptions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use an existing limited or public product.
INCOMPATIBLE_SERVICES	The service list contains incompatible services. [incompatible_services] Provide a valid list of compatible services.
NO_SERVICE_SPECIFIED	Provide at least 1 compatible service.
DUPLICATE_COMPATIBLE_AWS_SERVICES	The service list contains duplicate entries. Remove them. Each entry must be unique.
INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.
INVALID_VERSION_TITLE	Remove the following unsupported characters: [x, y, z]
INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.

Error code	Error message
DUPLICATE_VERSION_TITLE	The version title [duplicate_version_title] is a duplicate. Remove or change the title.
INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions.
MISSING_CONTAINER_IMAGES	Provide at least 1 container image.
NO_LICENSE_SECRET_KEYS	For Amazon EKS Anywhere products, provide 1 override parameter for license secret. Needs <code>DefaultValue</code> of <code>"\${AWSMP_LICENSE_SECRET}"</code> , see example in section.
TOO_MANY_CONTAINER_IMAGES	Provide fewer than 50 container images.
DUPLICATE_CONTAINER_IMAGES	The container image list contains duplicate images: [duplicate_images]. Provide a list with unique images.
INVALID_CONTAINER_IMAGES	Provide a valid URI for the container image.

Error code	Error message
INVALID_CONTAINER_IMAGE_URI	The image [invalid_image_uri] doesn't have access to this product. Upload the image to its corresponding product repository. For information about uploading, see Getting started with container products .
INVALID_CONTAINER_IMAGE_TAG	Avoid using 'latest' tag.
DUPLICATE_DELIVERY_OPTION_TITLES	Duplicate delivery option titles: [duplicate_titles]. Remove the duplicates
INVALID_DELIVERY_OPTION_TITLES	The delivery option titles [existing_titles] already exist. Provide a different title.
INVALID_FULFILLMENT_OPTION_TITLE	Provide delivery option title with fewer than (x) characters.
NO_SERVICE_ACCOUNT_CONFIGURATION	For paid products, provide 1 override parameter for service account configuration. Needs DefaultValue of "\${AWSMP_SERVICE_ACCOUNT}" , see example in section.
INVALID_DETAILS	Provided Details is not valid.
EMPTY_RESOURCE_NAME	Provide resource name.
EMPTY_RESOURCE_URL	Provide resource URL.
INVALID_RESOURCE_NAME	Provide resource name with fewer than 256 characters.
INVALID_RESOURCE_URL	Provide resource URL with fewer than 256 characters.
INVALID_SHORT_DESCRIPTION	Provide a short description with fewer than 1,000 characters.

Error code	Error message
INVALID_SHORT_DESCRIPTION	Provide short description.
SCAN_ERROR	Fix security vulnerability ""[y]"" on Image ""[x]"".
IMAGE_NOT_FOUND	The public image URI [invalid_image_uri] is invalid. Provide a valid URI.
INVALID_ARN	Provide a valid ARN for image access.
IMAGE_INACCESSIBLE	Provide a valid ARN for image access.
DUPLICATE_ADDON_NAME	The AddOn name you provided is already in use by a different product. Provide a different name.
DUPLICATE_ADDON_VERSION	The add-on version title [duplicate_version_title] is already in use. Provide a different title.
INVALID_ADDON_TYPE	The add-on types [invalid_types] are invalid. Provide a type from the supported list: [eks_addon_do_supported_types].
INVALID_KUBERNETES_VERSION	The Kubernetes versions [invalid_versions] are invalid. Provide versions from the supported list: [eks_addon_do_supported_kubernetes_versions].
DUPLICATE_KUBERNETES_VERSIONS	Duplicate Kubernetes versions: [duplicate_versions]. Provide a list with unique versions.
INVALID_ARCHITECTURE	The architectures [invalid_architectures] are invalid. Provide architectures from the Amazon EKS supported architectures : [eks_addon_do_supported_architectures].

Error code	Error message
DUPLICATE_SUPPORTED_ARCHITECTURES	Duplicate architectures: [duplicate_architectures]. Provide a list of unique, supported architectures.
INVALID_VISIBILITY_STATE	The states [invalid_states] are invalid for the {EKS_DO} delivery option. Provide a valid visibility state from the following allowed values: Limited.
INVALID_ENVIRONMENT_OVERRIDE_PARAMETER_VALUE	The override parameter values [invalid_values] are invalid. Provide a valid value from the following list: [eks_addon_do_environment_override_parameter_values].
DUPLICATE_ENVIRONMENT_OVERRIDE_PARAMETER_KEY	The environment override parameters contain duplicate keys: [duplicate_keys]. Remove them.
TOO_MANY_EKS_ADDON_DELIVERY_OPTIONS	Provide only one Amazon EKS add-on delivery option for the version.
INCOMPATIBLE_ADDON_NAME	The add-on name [provided_name] does not match the existing name. Reuse the existing name from the public version, or previous versions, of this add-on. You can only use one add-on name for each product.
INCOMPATIBLE_ADDON_TYPE	The add-on type [provided_type] does not match the existing type. Reuse the existing type from the public version, or previous versions, of this add-on. You can only use one add-on type for each product.

Error code	Error message
INCOMPATIBLE_ADDON_NAMESPACE	The add-on namespace [provided_namespace] does not match the existing namespace . Reuse the existing namespace from the public version, or previous versions, of this add-on. You can only use one add-on namespace for each product.
INVALID_HELM_CHART_URI	The Helm chart URI [invalid_uri] is invalid. Provide a URI in the SemVer 2 format.
INCOMPATIBLE_HELM_OBJECTS(INVALID_HELM_OBJECTS)	Provide a Helm chart without using the following unsupported Helm Objects: <unsupported-objects>.
INVALID_DEPENDENT_HELM_CHARTS	Provide a Helm chart that contains the following dependent charts directly in the parent chart directory and not externally sourced: <invalid-subcharts>.
INVALID_HELM_SENSITIVE_CONFIG	Provide an advanced configuration schema without sensitive information or secrets. Keywords: <sensitive-parameters-identified>
INVALID_HELM_UNDECLARED_IMAGES	Provide the following Helm chart images within the delivery option of the request: <list-of-images>.
INVALID_HELM_CHART_IMAGES	Provide a Helm chart containing images within repositories created via the AddRepositories change type. External images: <images-identified>.
INVALID_HELM_LINT	Provide a Helm chart that successfully passes Helm lint.
INVALID_HELM_TEMPLATE	Provide a Helm chart that successfully passes Helm template.

Error code	Error message
INVALID_HELM_CHART	Provide a Helm chart that adheres to AWS Marketplace guidance identified in Helm Charts bulleted list in the <i>AWS Marketplace Seller Guide</i> .
INVALID_ADDON_NAME	Provide an AddOn name that follows the following regex pattern: xx
INVALID_ADDON_NAMESPACE	The namespace values [invalid_namespaces] are invalid. The namespace must follow the {EKS_ADD_ON_NAMESPACE_REGEX} regular expression. For example, namespace, namespace-test.
INVALID_ADDON_NAME_PATTERN	Provide an add-on name that starts with a letter or digit, and then a combination of letters, digits, and hyphens. For example, test-addon, eksaddon
INVALID_ADDON_VERSION_PATTERN	Provide an add-on version using the following pattern: "<major>.<minor>.<patch>" (for example, 1.2.3, 0.1.2, 0.1.1)
EMPTY_DELIVERY_OPTION_IDS	Provide a list of delivery option IDs.
INVALID_DELIVERY_OPTIONS_INPUT	The list contains one or more invalid delivery options. Provide a valid list, and ensure that each option has a single delivery method.
OVERRIDE_PARAMETER_KEYS_CONTAINS_SPECIAL_CHARS	The override parameter keys [invalid_keys] contain invalid characters. You keys must use only letters, numbers, double quotes (") and plus signs (+).
INVALID_CONTAINER_IMAGE_REPOSITORY	The repositories [invalid_repositories] are invalid. Provide repositories created through the AddRepositories change type.

Error code	Error message
INVALID_CONTAINER_IMAGE_TAG_FORMAT	The container image tag [invalid_image_tag] is invalid. Provide a tag conforming to the {CONTAINER_IMAGE_TAG_REGEX} regular expression.
DUPLICATE_OVERRIDE_PARAMETER_KEYS	The override parameters contain duplicate keys [duplicate_keys]. Remove the duplicates.
UNSUPPORTED_CONTAINER_IMAGE_URI	The container image [unsupported_image] is unsupported. Provide an image that follows the Image Manifest V 2, Schema 1 .
DUPLICATE_REPOSITORY_NAMES	Duplicate repository names: [duplicate_repo_names]. Provide unique names.
INVALID_NAMESPACE	The namespace values [invalid_namespaces] are invalid. Provide values that conform to the {HELM_RELEASE_PARAM_REGEX} regular expression.
INVALID_RELEASE_NAME	The releaseName values [invalid_release_names] are invalid. Provide values that conform to the {HELM_RELEASE_PARAM_REGEX} regular expression.
OVERRIDE_PARAMETER_KEYS_CONTAINS_RESERVED_PARAMETER_KEYS	The override parameter key for delivery option titles [invalid_keys] is reserved. Reserved keys: [reserved_param_keys]. Provide a different key.

Update the visibility for an Amazon EKS add-on

You can use the Catalog API to update visibility for an Amazon EKS add-on delivery option of your product version in AWS Marketplace. Container and Helm delivery options for your container product are automatically created with 'Public' visibility status.

Note

The ability to update visibility of your product version is supported only for the Amazon EKS add-on delivery option from the listed versions. If your product isn't 'Public' already, submit a request to publish the product with 'Public' visibility status by using the AWS Marketplace Management Portal.

By default, when you create a version with the Amazon EKS add-on delivery option, it's published in 'Limited' status. A 'Limited' status means the product isn't publicly available across all the Regions for your buyers to use and deploy in an Amazon EKS cluster. You can update the visibility of the delivery option from 'Limited' to 'Public' by calling the `StartChangeSet` API operation with the `UpdateDeliveryOptionsVisibility` change type. Specify the `DeliveryOptions Id` from your product version that corresponds to the Amazon EKS add-on delivery option.

Request Syntax

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "UpdateDeliveryOptionsVisibility",
      "Entity":
      {
        "Identifier": "prod-example12345",
        "Type": "ContainerProduct@1.0"
      },
      "DetailsDocument":
      {
        "DeliveryOptions":
        [
          {
            "Id": "do-1234567891234567891234",
            "TargetVisibility": "Public"
          }
        ]
      }
    }
  ]
}
```

```
}
```

To add the `UpdateDeliveryOptionsVisibility` change type, provide information for the following fields :

- **Entity (object) (required)** – Your container-based product.
 - **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product uses: `ContainerProduct@1.0`.
- **DetailsDocument (object) (required)** – Details of the request, including the information about the repositories that you want to create. The following fields are all required.
 - **DeliveryOptions (list of objects)** – List of `DeliveryOption` objects, including the details of each:
 - **Id (string)** – Unique identifier for the `DeliveryOption`. (To get the unique identifier for the `DeliveryOption`, call the `DescribeEntity` action on the product that you're updating.
 - **TargetVisibility** – The intended new visibility of the product.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set.

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

The change request is added to a queue and processed, including scanning the container images and other information to ensure that it meets the [AWS Marketplace guidelines for container products](#). This process can take a few minutes to hours, depending on the number and size of your containers.

You can check the status of the request through the AWS Marketplace Management Portal, or through the AWS Marketplace Catalog API by using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

Asynchronous Errors

The following table shows errors that are specific to `AddDeliveryOptions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
EMPTY_DELIVERY_OPTION_IDS	Provide a list of delivery option IDs.
INVALID_VISIBILITY_STATE	The <code>TargetVisibility</code> option you provided is not supported. Please try again with an allowed option. The allowed option(s) are: <code>Public</code>
INVALID_DELIVERY_OPTION_IDS	You provided invalid delivery option details. Provide delivery option IDs that can be found in the product. IDs not found: [x]
DUPLICATE_DELIVERY_OPTION_IDS	Duplicate delivery option IDs: [duplicate_ids]. Provide unique delivery option IDs.
AUDIT_ERROR	You haven't completed independent software vendor (ISV) testing for all compatible Amazon EKS cluster versions for your Amazon EKS add-on version(s). You must complete testing before we can release the delivery option(s).
INVALID_DELIVERY_OPTION_TYPE	The delivery option type you provided is not valid. Ensure that your delivery option is of type: <code>EksAddOn</code> and try again.
INCOMPATIBLE_HELM_OBJECTS	Provide a Helm chart without unsupported Helm Objects: Unsupported Helm objects are

Error code	Error message
	as follows: all Release objects (except .Name and .Namespace), Helm hooks, and lookup functions.
INCOMPATIBLE_ADDON_NAME	The add-on name [provided_name] does not match the public version name. Update the public name before releasing.
INCOMPATIBLE_ADDON_TYPE	The add-on types don't match. Reuse the existing add-on type from the public add-on version or previous add-on versions of this product. Only one add-on is supported for each product.
INCOMPATIBLE_ADDON_NAMESPACE	The provided add-on namespace [provided_namespace] does not match the public version namespace. Update the add-on namespace before releasing.

Create repositories and resources

To create a new version of a container-based product, you must have the resources for the version available in AWS Marketplace repositories. You create the repositories and then push (upload) the Docker (and Helm) resources into the repositories. To learn how to create the repositories through the AWS Marketplace Management Portal, see [Add a new version of your product](#) in the *AWS Marketplace Seller Guide*.

To create new repositories, call `StartChangeSet` with the `AddRepositories` change type, as shown in the following example.

Request Syntax

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

Provide information for the fields to add the AddRepositories change type:

For more information about creating repositories, see [Adding a new version](#) in the *AWS Marketplace Seller Guide*.

- Entity (object) (required) – Your container-based product.
 - **Identifier** (string) (required) – Your product ID. For more information, see [Identifier](#).
 - **Type** (string) (required) – The Type is based on the delivery method (product type) that your product will be using: ContainerProduct@1.0.
- DetailsDocument (object) (required) – Details of the request. It includes the information about the repositories that you want to create. The included fields are all required.
 - **Repositories** (array of structures) – A list of repository objects. Each repository object includes a name and type.
 - **RepositoryName** (string) – The name of the repository to create.

- `RepositoryType` (string) – The type of the repository to create. The only allowed value is ECR.

Note

You can have up to 70 repositories per product, although you can add multiple resources, and versions of resources, to a single repository by giving them different tags when you push them.

After you create one or more repositories, you add your resources to the repositories. For information about how to push resources to repositories, see [Pushing an image](#) in the *Amazon Elastic Container Registry User Guide*. For information about the specific push commands needed for one of your repositories, see [Adding a new version](#) in the *AWS Marketplace Seller Guide*.

Asynchronous Errors

The following errors are specific to `AddRepositories` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_ECR_REPOSITORY_NAME	Provide repository name in the format: 'nginx-web-app'
DUPLICATE_ECR_REPOSITORY_NAME	The repository [duplicate_repo_names] already exists. Choose a different name.
MISSING_REPOSITORY_INFORMATION	Provide at least 1 repository name.
INVALID_ECR_REPOSITORY_NAME	Maximum character length 256 reached. Character length count is inclusive of the seller namespace.

Update version information

You can use the Catalog API to update the details of an existing version of your container-based product in AWS Marketplace.

Note

When a product is publicly available, you cannot update the version title, container images, delivery option title, or deployment resources for the version. If you need to update these aspects of a product, create a new version instead.

To update an existing version of your container-based product, call the `StartChangeSet` API operation with the `UpdateDeliveryOptions` change type, as shown in the following example. This updates the detail information for the delivery options that you specify, as well as the associated version. You must include at least one delivery option.

Container Image Delivery Request Syntax

```
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 Chart Delivery Request Syntax

```
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-ccccccccccccc",
            "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 Add-On Delivery Request Syntax

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

```

Provide information for the fields to add the `UpdateDeliveryOptions` change type:

For more information about these fields, see [Adding a new version](#) in the *AWS Marketplace Seller Guide*.

- **Entity (object) (required)** – Your container-based product.
 - **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product will be using: `ContainerProduct@1.0`.
- **DetailsDocument (object) (required)** – Details of the request. It includes any information about the version of your container-based product that you would like to update. The included fields are all optional, but you must include at least one field to update.
- **Version (object)** – Details about the software version.
 - **VersionTitle (string)** – The title of the version that you are creating. Typically this is a description of the version, such as **Version 1.1** or simply **1.1**. Your buyers will be able to choose the version to deploy from a list of all version titles.

This property can't be updated if the product is already published publicly.

- **ReleaseNotes (string)** – Notes for buyers to tell them about changes from one version to the next.
- **DeliveryOptions (list of objects)** – List of `DeliveryOption` objects, including the details of each:
 - **Id (string)** – Unique identifier for the `DeliveryOption` (you can get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the product you are updating).
 - **Details (object)** – Holds the details of a delivery option. Note that this nested details object does *not* need to be double-escaped.
 - **EcrDeliveryOptionDetails (object)** – The details of the container image delivery option.
 - **DeliveryOptionTitle (string)** – A short description that allows your buyer to choose between your delivery options.

This property can't be updated if the product is already published publicly.

- **ContainerImages (array of strings)** – An array of container image URLs used by this version. The path will be the repository that you have uploaded the image to, with the

tag for the image used by this version. If this field is included, the list must include all needed images, even images that are not changing.

This property can't be updated if the product is already published publicly.

- **DeploymentResources** (array of objects) – An array of other deployment resources needed for the version, such as links to Helm charts or other documentation. Each resource includes a name to describe it and a URL that points at the resource. On the launch page for your version, this displays as a list of links.

This property can't be updated if the product is already published publicly.

- **Name** (string) – The text of the hyperlink that is shown to the buyer.
- **Url** (string) – The URL of the hyperlink shown to the buyer.
- **CompatibleServices** (array of strings) – An array of services that the release is compatible with. Valid options: ECS, EKS, ECS-Anywhere, EKS-Anywhere, and Bedrock-AgentCore.
- **AgenticType** The type of runtime agent. Valid options: AGENT, MCP_SERVER, or A2A_SERVER.
- **Description** (string) – A longer description of the delivery option to give details to your buyer. You can also include a link to more instructions hosted elsewhere.
- **UsageInstructions** (string) – Provide instructions on how to deploy and use your product. You can also add a link to usage instructions hosted elsewhere. Can be up to 4,000 characters.
- **EnvironmentVariables** – List of environment variables that are required by the AgentCore Runtime container and that will be pre-populated for buyers upon deployment. For each variable, provide an object with the *name* as expected by your container, a *description*, and an optional *defaultValue*. For variables such as credentials or API keys that are unique, do not provide a default value. You can use the description to specify details about the variable as well as possible values. All of the provided variables with their default values will be pre-populated when buyers launch your product.
- **Id** (string) – Unique identifier for the `DeliveryOption` (you can get the unique identifier for the `DeliveryOption` by calling the `DescribeEntity` action on the product you are updating).
- **Details** (object) – Holds the details of a delivery option. Note that this nested details object does *not* need to be double-escaped.

- `HelmDeliveryOptionDetails` (object) – The details of the Helm chart delivery option.
- `DeliveryOptionTitle` (string) – A short description that allows your buyer to choose between your delivery options.

This property can't be updated if the product is already published publicly.

- `ContainerImages` (array of strings) – An array of container image URLs used by this version. The path will be the repository that you have uploaded the image to, with the tag for the image used by this version. The list must include all needed images, even images that have not changed from previous versions. See the next section for information about creating repositories using the Catalog API.
- `HelmChartUri` (string) – The URL to the Helm chart hosted in Amazon ECR that the buyer will install to launch the software.
- `CompatibleServices` (array of strings) – An array of services that the release is compatible with. Valid options are ECS and EKS.
- `Description` (string) – A longer description of the delivery option to give details to your buyer. You can also include a link to more instructions provided elsewhere.
- `UsageInstructions` (string) – Provide instructions about the usage for this delivery option. Can be up to 4,000 characters.
- `MarketplaceServiceAccountName` (string) – The name of the Kubernetes service account. The service account will be used to connect to AWS Identity and Access Management for permissions to call AWS services.
- `ReleaseName` (string) – The name for the Helm release provided to the `helm install` command that buyers use to launch the software.
- `Namespace` (string) – The Kubernetes namespace where the Helm chart will be installed.
- `OverrideParameters` (array of objects) – Parameters that will be used in the Helm commands that launch the application. Buyers can override the default values.
 - `Key` (string) – The key for the parameter in dot notation (`override.example.key`).
 - `DefaultValue` (string) – The default value for this override parameter.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed, including scanning the container images and other information to ensure that it meets the [AWS Marketplace guidelines for container products](#). This process can take a few minutes to hours, depending on the number and size of your containers.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

Asynchronous Errors

The following errors are specific to UpdateDeliveryOptions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use an existing limited or public product.
INCOMPATIBLE_SERVICES	The service list contains incompatible services. [incompatible_services] Provide a valid list of compatible services.
NO_SERVICE_SPECIFIED	Provide at least 1 compatible service.
DUPLICATE_COMPATIBLE_AWS_SERVICES	The service list contains duplicate entries. Remove them. Each entry must be unique.
INVALID_VERSION_TITLE	Remove spaces before the trademark symbol.

Error code	Error message
INVALID_VERSION_TITLE	Remove the following unsupported characters: [x, y, z]
INVALID_VERSION_TITLE	Remove spaces from the beginning of the version title.
INVALID_VERSION_TITLE	Provide version title with fewer than [x] characters.
DUPLICATE_VERSION_TITLE	The version title [duplicate_version_title] is a duplicate. Remove or change the title.
INVALID_RELEASE_NOTES	Remove spaces before the trademark symbol.
INVALID_RELEASE_NOTES	Remove unsupported characters: [x, y, z]
INVALID_RELEASE_NOTES	Remove spaces from the beginning of release notes.
INVALID_RELEASE_NOTES	Provide release notes with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Remove spaces before the trademark symbol.
INVALID_USAGE_INSTRUCTIONS	Remove unsupported characters: [x, y, z]
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions with fewer than (x) characters.
INVALID_USAGE_INSTRUCTIONS	Provide usage instructions.
MISSING_CONTAINER_IMAGES	Provide at least 1 container image.
TOO_MANY_CONTAINER_IMAGES	Provide fewer than 50 container images.
DUPLICATE_CONTAINER_IMAGES	The container image list contains duplicate images: [duplicate_images]. Provide a list with unique images.

Error code	Error message
INVALID_CONTAINER_IMAGES	Provide a valid URI for the container image.
INVALID_CONTAINER_IMAGE_URI	The image [invalid_image_uri] doesn't have access to this product. Upload the image to its corresponding product repository. For information about uploading, see Getting started with container products .
INVALID_CONTAINER_IMAGE_TAG	Avoid using 'latest' tag.
MISSING_DELIVERY_OPTION_IDS	Provide delivery option from existing list of Ids.
EMPTY_DELIVERY_OPTION_IDS	Provide non-empty list of delivery option IDs.
DUPLICATE_DELIVERY_OPTION_IDS	Duplicate delivery option IDs: [duplicate_ids]. Provide unique delivery option IDs.
DUPLICATE_DELIVERY_OPTION_TITLES	Duplicate delivery option titles: [duplicate_titles]. Remove the duplicates
INVALID_DELIVERY_OPTION_TITLES	The delivery option titles [existing_titles] already exist. Provide a different title.
INVALID_FULFILLMENT_OPTION_TITLE	Provide delivery option title with fewer than (x) characters.
EMPTY_RESOURCE_NAME	Provide resource name.
EMPTY_RESOURCE_URL	Provide resource URL.
INVALID_RESOURCE_NAME	Provide resource name with fewer than 256 characters.
INVALID_RESOURCE_URL	Provide resource URL with fewer than 256 characters.

Error code	Error message
INVALID_SHORT_DESCRIPTION	Provide a short description with fewer than 1,000 characters.
INVALID_SHORT_DESCRIPTION	Provide short description.
NO_LICENSE_SECRET_KEYS	For Amazon EKS Anywhere products, provide 1 override parameter for license secret. Needs <code>DefaultValue</code> of <code>"\${AWSMP_LICENSE_SECRET}"</code> , see example in section.
NO_SERVICE_ACCOUNT_CONFIGURATION	For paid products, provide 1 override parameter for service account configuration. Needs <code>DefaultValue</code> of <code>"\${AWSMP_SERVICE_ACCOUNT}"</code> , see example in section.
SCAN_ERROR	Fix security vulnerability <code>""[y]""</code> on Image <code>""[x]""</code> .
FIELD_NOT_ALLOWED_TO_CHANGE	Field [x] cannot be changed.
INVALID_DELIVERY_OPTIONS_STATUS	The delivery option IDs [invalid_ids] are invalid. Provide delivery options in the limited or public state.
NO_CHANGE_FOUND	Provide at least 1 change.
MULTIPLE_VERSION_UPDATE	Provide delivery option IDs from the same version.
OVERRIDE_PARAMETER_KEYS_CONTAINS_SPECIAL_CHARS	The override parameter keys [invalid_keys] contain invalid characters. You keys must use only letters, numbers, double quotes (" ") and plus signs (+).

Error code	Error message
INVALID_CONTAINER_IMAGE_REPOSITORY	The repositories [invalid_repositories] are invalid. Provide repositories created through the AddRepositories change type.
INVALID_CONTAINER_IMAGE_TAG_FORMAT	The container image tag [invalid_image_tag] is invalid. Provide a tag conforming to the {CONTAINER_IMAGE_TAG_REGEX} regular expression.
DUPLICATE_OVERRIDE_PARAMETER_KEYS	The override parameters contain duplicate keys [duplicate_keys]. Remove the duplicates.
UNSUPPORTED_CONTAINER_IMAGE_URI	The container image [unsupported_image] is unsupported. Provide an image that follows the Image Manifest V 2, Schema 1 .
INVALID_NAMESPACE	The namespace values [invalid_namespaces] are invalid. Provide values that conform to the {HELM_RELEASE_PARAM_REGEX} regular expression.
INVALID_RELEASE_NAME	The releaseName values [invalid_release_names] are invalid. Provide values that conform to the {HELM_RELEASE_PARAM_REGEX} regular expression.
OVERRIDE_PARAMETER_KEYS_CONTAINS_RESERVED_PARAMETER_KEYS	The override parameter key for delivery option titles [invalid_keys] is reserved. Reserved keys: [reserved_param_keys]. Provide a different key.
INCOMPATIBLE_ADDON_NAME	The add-on name [provided_name] does not match the existing name. Reuse the existing name from the public version, or previous versions, of this add-on. You can only use one add-on name for each product.

Error code	Error message
INCOMPATIBLE_ADDON_NAMESPACE	The add-on namespace [provided_namespace] does not match the existing namespace . Reuse the existing namespace from the public version, or previous versions, of this add-on. You can only use one add-on namespace for each product.

Restrict a version

You can use the Catalog API to restrict a version of your container-based product in AWS Marketplace. This prevents new buyers from being able to use that version. There must be at least one publicly available version in a product. You cannot restrict the only remaining publicly available version for a product.

To restrict a version, call the `StartChangeSet` API operation with the `RestrictDeliveryOptions` change type, as shown in the following example.

Note

Restricting one or more, but not all, delivery options from a version will remove those options from being available to your buyers. Restricting all delivery options for a version will remove that version from the AWS Marketplace catalog.

Restricting an Amazon EKS add-on is currently not supported through the Catalog API. Restricted versions are still available for existing customers.

Request Syntax

```
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"
        ]
    }
  ]
}

```

Provide information for the fields to add the RestrictDeliveryOptions change type:

- Entity (object) (required) – Your container-based product.
 - Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
 - Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: ContainerProduct@1.0.
- DetailsDocument (object) (required) – Details of the request. It includes IDs for the delivery options of your container-based product that you would like to restrict.
 - DeliveryOptionIds (array of strings) – List of DeliveryOption IDs for the versions that you want to restrict. You can get the unique identifier for the DeliveryOption by calling the DescribeEntity action on the product you are restricting.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

```

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

```

The change request is added to a queue and processed. This process can take a few minutes to hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

For more information about change sets, see [Working with change sets](#). For more information about errors in seller product change sets, see [Change set status and errors](#).

Asynchronous Errors

The following errors are specific to RestrictDeliveryOptions actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use a public product.
MISSING_DELIVERY_OPTION_IDS	Provide delivery option from existing list of IDs.
INVALID_DELIVERY_OPTIONS_STATUS	The delivery option IDs [invalid_ids] are invalid. Provide delivery options in the public state.
EMPTY_DELIVERY_OPTION_IDS	Provide non-empty list of delivery option IDs.
INVALID_MINIMUM_PUBLIC_DELIVERY_OPTIONS	Cannot restrict all delivery option IDs.
DUPLICATE_DELIVERY_OPTION_IDS	Duplicate delivery option IDs: [duplicate_ids]. Provide unique delivery option IDs.

Work with SaaS products using the AWS Marketplace APIs

You can use the AWS Marketplace Catalog API to automate tasks for working with SaaS-based products. For information about creating SaaS-based products, see [Create a product](#). The following topics describe how to perform Catalog API actions:

Topics

- [Configure Free pricing model for SaaS products](#)
- [SaaS Quick Launch options](#)
- [Add delivery options](#)
- [SaaSUrlDeliveryOption](#)

Configure Free pricing model for SaaS products

SaaS products can be offered with a Free pricing model, which allows buyers to use your product at no cost. When using the Free pricing model for SaaS products, you must configure pricing dimensions with zero-dollar pricing in either `UsageBasedPricingTerm` or `ConfigurableUpfrontPricingTerm`.

Note

The Free pricing model requirement for SaaS products is unique. For SaaS products with Free pricing, you must create at least one dimension with either `UsageBasedPricingTerm` or `ConfigurableUpfrontPricingTerm`, and all dimensions must be priced at \$0.00. This requirement does not apply to AMI, container, or machine learning products.

To configure Free pricing for a SaaS product:

1. Create pricing dimensions for your product using the `AddDimensions` change type. For more information, see [Add pricing dimensions](#) in the seller products documentation.
2. Create an offer with the `PricingModel` set to `Free` and include either `UsageBasedPricingTerm` or `ConfigurableUpfrontPricingTerm` with all dimension prices set to \$0.00. For more information about creating offers, see [Work with offers using the AWS Marketplace APIs](#).

When configuring the offer, ensure that:

- All rate card prices in `UsageBasedPricingTerm` are set to "0.00" (string format)
- All charge amounts in `ConfigurableUpfrontPricingTerm` are set to "0.00" (string format)
- At least one dimension with \$0.00 pricing is included

SaaS Quick Launch options

For information about Quick Launch options, see the following resources:

- (Buyers) For Quick Launch options for SaaS products, see [Configuring and launching SaaS products using Quick Launch](#) in the *AWS Marketplace Buyer Guide*.
- (Sellers) For Quick Launch options for SaaS products, see [Configure Quick Launch](#) in the *AWS Marketplace Seller Guide*.
- For a Quick Launch workshop for SaaS products, see [Lab: Enable SaaS Quick Launch](#) in the *AWS Marketplace seller workshop*.

Add delivery options

You can use the Catalog API to add delivery options for a SaaS product in AWS Marketplace. API delivery options enable sellers to offer API-based services that integrate with AWS services such as Amazon Bedrock for AI agent workflows.

To add API delivery options, call the `StartChangeSet` API operation with the `AddDeliveryOptions` change type to add delivery details, as shown in the following example.

The following topics explain how to add use the `ApiDeliveryOptionDetails` and `SaaSUrlDeliveryOptionDetails` options.

Topics

- [ApiDeliveryOptionDetails](#)
- [Update delivery options](#)
- [Update delivery option visibility](#)

ApiDeliveryOptionDetails

The example in this section supports the `ApiDeliveryOptionDetails` delivery option type, which allows you to specify API endpoints, authentication methods, and integration protocols for your SaaS product.

Request Syntax

```
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\\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"
                  }
                ]
              }
            ]
          }
        }
      }
    ]
  }
}
```

```
}  
}  
]  
}  
]  
}  
]  
}  
]  
}
```

Provide information for the fields to add the AddDeliveryOptions change type:

- **Entity (object) (required)** – Your SaaS-based product.
 - **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product will be using: `SaaSProduct@1.0`.
- **DetailsDocument (object) (required)** – Details of the request.
 - **DeliveryOptions (array)** – Details of the API delivery options being added.
 - **Details (object)** – Contains the `ApiDeliveryOptionDetails` of a delivery option to be added.
 - **ApiDeliveryOptionDetails (object)** – Contains the API delivery option details for a SaaS product.
 - **ApiType (string) (required)** – Type of API offering. Valid values: `MCP_SERVER`, `KNOWLEDGE_BASE`, `AGENT`, `GUARDRAIL`, `OTHER`.
 - **QuickLaunchEnabled (boolean) (required)** – Determines if buyers can use Quick Launch to configure and launch the software.
 - **FulfillmentUrl (string) (required)** – The URL to the seller's software registration landing page.
 - **UsageInstructions (string) (required)** – Instructions for using this API delivery option. Supports markdown formatting. Maximum 30,000 characters.
 - **CompatibleServices (array) (optional)** – Supported AWS services for this delivery option. Currently supports `Bedrock-AgentCore`.
 - **Endpoints (array) (required)** – The API endpoints available for this offering. Must contain exactly one endpoint.

- **Name** (string) (optional) – The name of the API endpoint. Must match pattern `^[A-Za-z][a-zA-Z0-9-]+$`. Maximum 100 characters.
- **EndpointUrl** (string) (required) – The URL of the API endpoint. Must be a valid HTTPS URL.
- **Description** (string) (optional) – A description of the API endpoint and its functionality. Maximum 4,000 characters.
- **AuthorizationTypes** (array) (required) – The types of authorization required to access the API endpoint. Valid values: `API_KEY`, `OAUTH2`. Must contain 1-2 unique values.
- **Schemas** (array) (optional) – The schema specifications for the API endpoint. Maximum 1 schema.
 - **Type** (string) (required) – Schema type. Valid value: `OPEN_API`.
 - **SchemaUrl** (string) (required) – The S3 URL of the schema that has been ingested into a Marketplace owned S3 bucket.
- **IntegrationProtocols** (array) (optional) – Protocol types supported by the endpoint. Maximum 2 protocols.
 - **Type** (string) (required) – Protocol identifier. Valid values: `MCP`, `A2A`.
 - **UsageInstructions** (string) (required) – Additional instructions for utilizing the protocol with the endpoint. Maximum 30,000 characters.

Response Syntax

A change set is created for your request. The response to this request gives you the ID for the change set and looks like the following.

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

Synchronous Validations

Error condition	Message	HTTP code
Invalid API Type	You provided an invalid API type. Valid values are: MCP_SERVER, KNOWLEDGE_BASE, AGENT, GUARDRAIL, OTHER.	422
Invalid Fulfillment Url	Provide a valid fulfillment URL beginning with "https://".	422
Missing Required Fields	Required parameter is missing. You must provide ApiType, QuickLaunchEnabled , FulfillmentUrl, UsageInstructions, and Endpoints.	422
Invalid Endpoint URL	Provide a valid endpoint URL beginning with "https://".	422
Missing Authorization Types	You must provide at least one authorization type. Valid values are: API_KEY, OAUTH2.	422
Invalid Authorization Types	You provided invalid authorization types. Valid values are: API_KEY, OAUTH2.	422
Too Many Endpoints	You cannot provide more than 1 endpoint for API delivery options.	422
Invalid Compatible Services	You provided invalid compatible services. Valid values are: Bedrock-AgentCore.	422
Invalid Schema Type	You provided an invalid schema type. Valid value is: OPEN_API.	422
Invalid Schema URL	Provide a valid schema URL that points to a Marketplace owned S3 bucket.	422
Invalid Integration Protocol	You provided an invalid integration protocol type. Valid values are: MCP, A2A.	422

Error condition	Message	HTTP code
Too Many Integration Protocols	You cannot provide more than 2 integration protocols.	422
Invalid Usage Instructions	Usage instructions exceed the maximum length of 30,000 characters.	422

Asynchronous Errors

Error code	Error message
DUPLICATE_DELIVERY_OPTIONS	You provided one or more delivery option types that already exist for this product. Provide a unique delivery option type or use <code>UpdateDeliveryOptions</code> if you intended to change an existing delivery option.
INVALID_FULFILLMENT_URL	The URL you provided returned HTTP status code [x]. Provide a fulfillment URL that renders with a 200.
INVALID_ENDPOINT_URL	The URL you provided returned HTTP status code [x]. Provide an endpoint URL that renders with a 200.
INVALID_SCHEMA_URL	The schema URL you provided is invalid or inaccessible. Provide a valid schema URL that points to an Amazon S3 bucket owned by AWS Marketplace.

Update delivery options

You can use the Catalog API to update the delivery options for a SaaS product in AWS Marketplace.

To update the delivery options, call the `StartChangeSet` API operation with the `UpdateDeliveryOptions` change type, as shown in the following example.

Note

This supports the `ApiDeliveryOptionDetails` delivery option type. You must provide the delivery option ID to identify which option to update.

Request Syntax

```
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",
```


- `ApiType` (string) (required) – Type of API offering. Valid values: `MCP_SERVER`, `KNOWLEDGE_BASE`, `AGENT`, `GUARDRAIL`, `OTHER`.
- `QuickLaunchEnabled` (boolean) (required) – Determines if buyers can use Quick Launch to configure and launch the software.
- `FulfillmentUrl` (string) (required) – The URL to be updated for the SaaS product.
- `UsageInstructions` (string) (required) – Instructions for using this API delivery option. Supports markdown formatting. Maximum 30,000 characters.
- `CompatibleServices` (array) (optional) – Supported AWS services for this delivery option. Currently supports `Bedrock-AgentCore`.
- `Endpoints` (array) (required) – The API endpoints available for this offering. Must contain exactly one endpoint.
 - `Name` (string) (optional) – The name of the API endpoint.
 - `EndpointUrl` (string) (required) – The URL of the API endpoint to be updated.
 - `Description` (string) (optional) – A description of the API endpoint and its functionality.
 - `AuthorizationTypes` (array) (required) – The types of authorization required to access the API endpoint. Valid values: `API_KEY`, `OAuth2`.
 - `Schemas` (array) (optional) – The schema specifications for the API endpoint.
 - `Type` (string) (required) – Schema type. Valid value: `OPEN_API`.
 - `SchemaUrl` (string) (required) – The S3 URL of the updated schema.
 - `IntegrationProtocols` (array) (optional) – Protocol types supported by the endpoint.
 - `Type` (string) (required) – Protocol identifier. Valid values: `MCP`, `A2A`.
 - `UsageInstructions` (string) (required) – Additional instructions for utilizing the protocol with the endpoint.

Response Syntax

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

The change request is added to a queue and processed. This includes validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action.

Synchronous Validations

Error condition	Message	HTTP code
Empty Delivery Option Ids	Provided Details is not valid. String " at /DeliveryOptions/0/Id does not match required schema regex, '^do-[a-zA-Z0-9]+\$'	422
Missing Delivery Option Ids	Provided Details is not valid. JSON at /DeliveryOptions/0 is missing required properties: ['Id'].	422
Duplicate Delivery Option Ids	Provide unique delivery option IDs.	422
Invalid Fulfillment Url	Provide a valid fulfillment URL beginning with "https://".	422
Invalid Delivery Option IDs	Provide delivery option IDs that can be found in the product. IDs not found: [x]	422
Multiple URL Delivery Options	You provided more than one URL delivery option. Provide one URL delivery option.	422
Missing delivery option ids	The delivery option ID is missing. Provide one or more valid delivery option IDs that you wish to update, or use <code>AddDeliveryOptions</code> if you intended to add a new delivery option.	422

Error condition	Message	HTTP code
Invalid Launch URL	Provide a valid launch URL beginning with "https://".	422
Missing Launch Url	Required parameter LaunchUrl is missing. You must provide a LaunchUrl .	422
Missing deployment templates	The deployment template is missing. Provide at least one deployment template.	422
Too many deployment templates	You cannot provide more than 20 deployment templates.	422
Invalid template URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]	422
Invalid deployment template stack name	The deployment template stack name is invalid. Provide a valid stack name using only alphanumeric characters and hyphens. It must start with an alphabetical character and can't be longer than 128 characters.	422
Duplicate deployment template title	You provided duplicate deployment template titles. Provide unique deployment template titles.	422
Duplicate deployment template URL	You provided duplicate deployment template urls. Provide unique deployment template urls.	422

Error condition	Message	HTTP code
Invalid deployment template type	The deployment template type is invalid. Provide a valid deployment template type. Supported values are ["CloudFormation@1.0"].	422
Invalid deployment template IAM policy	The deployment template IAM policy is invalid. Provide a valid IAM policy.	422
Invalid usage instructions	• Images aren't supported by the usage instructions. Remove the image [x]. • You provided a link to an invalid URL in the usage instructions: [x]. Provide a valid URL. • You provided a link with an unsupported URI scheme in the usage instructions. Use a supported scheme: ["http", "https", "tel", "mailto"].	422

Asynchronous Errors

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
AUDIT_ERROR	AWS MP Catalog Audits List - CQ team
INVALID_FULFILLMENT_URL	The URL you provided returned HTTP status code [x]. Provide a fulfillment URL that renders with a 200.

Error code	Error message
INVALID_LAUNCH_URL	The URL you provided returned HTTP status code [x]. Provide a launch URL that renders with a 200.
INVALID_TEMPLATE_URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]

Update delivery option visibility

You can use the Catalog API to configure permissions so that only some users can change the visibility for a SaaS product in AWS Marketplace.

To configure permissions so that only some users can change the visibility for a SaaS product, call the StartChangeSet API operation with the UpdateDeliveryOptionsVisibility change type, as shown in the following example.

Note

This is only supported for one delivery option: SaaSUrlDeliveryOptionDetails.

Request Syntax

```
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"
                        ]
                    }
                }
            }
        ]
    }
}

```

Provide information for the fields to add the UpdateDeliveryOptionsVisibility change type:

- Entity (object) (required) – Your SaaS-based product.
 - Identifier (string) (required) – Your product ID. For more information, see [Identifier](#).
 - Type (string) (required) – The Type is based on the delivery method (product type) that your product will be using: SaaSProduct@1.0.
- DetailsDocument (object) (required) – Details of the request.
 - DeliveryOptions (array) – List of DeliveryOptions to be updated.
 - TargetVisibility (string) – The delivery option id to be updated.
 - TargetVisibility (string) – The intended new visibility of the delivery option.

Possible values: Limited, Public, and Unavailable.

 Note

There is always exactly one Public delivery option, and a maximum of one Limited delivery option.

- **Targeting (object) *optional*** – Targeting of the delivery option, used in conjunction with the Limited visibility status to be able to test the new delivery option before changing the visibility to Public.
- **PositiveTargeting (object)** – Specifying inclusive targeting.
 - **BuyerAccounts (array of strings)** – The list of buyer AWS account ids who will be able to use the new delivery option.

Min size: 0. Max size: 100.

Response Syntax

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the DescribeChangeSet action.

Synchronous Validations

Error condition	Message	HTTP code
Missing delivery option ids	The delivery option ID is missing. Provide one or more valid delivery option IDs that you wish to update, or use AddDeliveryOptions if	422

Error condition	Message	HTTP code
	you intended to add a new delivery option.	
Invalid visibility	You provided invalid option for <code>TargetVisibility</code> . Allowed options are: <code>Limited</code> , <code>Public</code> , <code>Unavailable</code> .	422
Invalid targeting	You provided invalid option for <code>PositiveTargeting</code> . You must provide a valid parameter for <code>BuyerAccounts</code> .	422
Missing Visibility and Targeting	You provided invalid delivery option visibility details. You must provide a valid parameter for at least one of <code>TargetVisibility</code> or <code>Targeting</code> .	422
Too many AWS account ids	You cannot provide more than 100 targeted buyer accounts.	422

Asynchronous Errors

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	You provided invalid delivery option details. Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_VISIBILITY	You provided more than one delivery option for the public state. Provide only one public delivery option.
INVALID_VISIBILITY	You didn't provide a public delivery option. Provide one public delivery option.

Error code	Error message
AUDIT_ERROR	Varies based on MCO manual review.

SaaSUrlDeliveryOption

The following example shows how to use the SaaSUrlDeliveryOptionDetails to update the FulfillmentUrl.

Request Syntax

```
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/*\"
                    ]}]}"
                  }
                ]
              }
            }
          ]
        }
      }
    }
  ]
}
```

Provide information for the fields to add the AddDeliveryOptions change type:

- **Entity (object) (required)** – Your SaaS-based product.
 - **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product will be using: SaaSProduct@1.0.
- **DetailsDocument (object) (required)** – Details of the request.
 - **DeliveryOptions (array)** – Contains the details of the delivery options being updated.
 - **TargetVisibility (string)** – Unique identifier for the DeliveryOption. (Get the unique identifier for the DeliveryOption by calling the DescribeEntity action on the product you are updating.)
 - **Details (object)** – Contains the SaaSUrlDeliveryOptionDetails of the delivery option to be updated.
 - **SaaSUrlDeliveryOptionDetails (object)** – Contains the FulfillmentUrl of a delivery option for SaaS product.
 - **FulfillmentUrl (string)** – The URL to be updated for the SaaS product.
 - **LaunchUrl (string)** – The URL to your SaaS product's landing page. This is required if QuickLaunchEnabled is set to True.
 - **UsageInstructions (string)** – Instructions for using this delivery option. Include documentation for manual steps for customers who won't use DeploymentTemplates.
 - **DeploymentTemplates (array)** – Deployment templates that customers can use to set up and configure the SaaS product and any related AWS resources.
 - **Title (string)** – The display name of the deployment template.
 - **Description (string)** – A description for what the deployment template contains.

- **IamPolicy** (string) – An IAM policy describing the permissions needed to deploy the template. Buyers can use this IAM policy to quickly deploy the template.
- **CloudFormationDetails** (object) – The details of a CloudFormation template.
 - **TemplateUrl** (string) – The URL for the deployment template.
 - **DefaultStackName** (string) – The default name used in CloudFormation when the customer creates the template.

Response Syntax

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

The change request is added to a queue and processed. This includes validating information to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action.

Synchronous Validations

Error condition	Message	HTTP code
Empty Delivery Option Ids	Provided Details is not valid. String " at /DeliveryOptions/0/Id does not match required schema regex, '^do-[a-zA-Z0-9]+\$'	422
Missing Delivery Option Ids	Provided Details is not valid. JSON at /DeliveryOptions/0 is missing required properties: ['Id'].	422
Duplicate Delivery Option Ids	Provide unique delivery option IDs.	422
Invalid Fulfillment Url	Provide a valid fulfillment URL beginning with "https://".	422

Error condition	Message	HTTP code
Invalid Delivery Option IDs	Provide delivery option IDs that can be found in the product. IDs not found: [x]	422
Multiple URL Delivery Options	You provided more than one URL delivery option. Provide one URL delivery option.	422
Missing delivery option ids	The delivery option ID is missing. Provide one or more valid delivery option IDs that you wish to update, or use <code>AddDeliveryOptions</code> if you intended to add a new delivery option.	422
Invalid Launch URL	Provide a valid launch URL beginning with "https://".	422
Missing Launch Url	Required parameter <code>LaunchUrl</code> is missing. You must provide a <code>LaunchUrl</code> .	422
Missing deployment templates	The deployment template is missing. Provide at least one deployment template.	422
Too many deployment templates	You cannot provide more than 20 deployment templates.	422
Invalid template URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]	422

Error condition	Message	HTTP code
Invalid deployment template stack name	The deployment template stack name is invalid. Provide a valid stack name using only alphanumeric characters and hyphens. It must start with an alphabetical character and can't be longer than 128 characters.	422
Duplicate deployment template title	You provided duplicate deployment template titles. Provide unique deployment template titles.	422
Duplicate deployment template URL	You provided duplicate deployment template urls. Provide unique deployment template urls.	422
Invalid deployment template type	The deployment template type is invalid. Provide a valid deployment template type. Supported values are ["CloudFormation@1.0"].	422
Invalid deployment template IAM policy	The deployment template IAM policy is invalid. Provide a valid IAM policy.	422

Error condition	Message	HTTP code
Invalid usage instructions	- Images aren't supported by the usage instructions. Remove the image [x]. - You provided a link to an invalid URL in the usage instructions: [x]. Provide a valid URL. - You provided a link with an unsupported URI scheme in the usage instructions. Use a supported scheme: ["http", "https", "tel", "mailto"].	422

Asynchronous Errors

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
AUDIT_ERROR	AWS MP Catalog Audits List - CQ team
INVALID_FULFILLMENT_URL	The URL you provided returned HTTP status code [x]. Provide a fulfillment URL that renders with a 200.
INVALID_LAUNCH_URL	The URL you provided returned HTTP status code [x]. Provide a launch URL that renders with a 200.
INVALID_TEMPLATE_URL	Quick Start URL is invalid. Provide deployment template URL that is published through AWS QuickStarts to Amazon S3. Invalid deployment templates URL: [x]

Update delivery option visibility

You can use the Catalog API to configure permissions so that only some users can change the visibility for a SaaS product in AWS Marketplace.

To do so, call the `StartChangeSet` API operation with the `UpdateDeliveryOptionsVisibility` change type, as shown in the following example.

Note

This is only supported for one delivery option: `SaaSUrlDeliveryOptionDetails`.

Request Syntax

```
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"
            ]
        }
    }
}

```

Provide information for the fields to add the `UpdateDeliveryOptionsVisibility` change type:

- **Entity (object) (required)** – Your SaaS-based product.
 - **Identifier (string) (required)** – Your product ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – The Type is based on the delivery method (product type) that your product will be using: `SaaSProduct@1.0`.
- **DetailsDocument (object) (required)** – Details of the request.
 - **DeliveryOptions (array)** – List of `DeliveryOptions` to be updated.
 - **TargetVisibility (string)** – The delivery option id to be updated.
 - **TargetVisibility (string)** – The intended new visibility of the delivery option.

Possible values: `Limited`, `Public`, and `Unavailable`.

Note

There is always exactly one `Public` delivery option, and a maximum of one `Limited` delivery option.

- **Targeting (object) *optional*** – Targeting of the delivery option, used in conjunction with the `Limited` visibility status to be able to test the new delivery option before changing the visibility to `Public`.
 - **PositiveTargeting (object)** – Specifying inclusive targeting.

- **BuyerAccounts** (array of strings) – The list of buyer AWS account ids who will be able to use the new delivery option.

Min size: 0. Max size: 100.

Response Syntax

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action.

Synchronous Validations

Error condition	Message	HTTP code
Missing delivery option ids	The delivery option ID is missing. Provide one or more valid delivery option IDs that you wish to update, or use <code>AddDeliveryOptions</code> if you intended to add a new delivery option.	422
Invalid visibility	You provided invalid option for <code>TargetVisibility</code> . Allowed options are: <code>Limited</code> , <code>Public</code> , <code>Unavailable</code> .	422
Invalid targeting	You provided invalid option for <code>PositiveTargeting</code> . You must provide a valid parameter for <code>BuyerAccounts</code> .	422

Error condition	Message	HTTP code
Missing Visibility and Targeting	You provided invalid delivery option visibility details. You must provide a valid parameter for at least one of <code>TargetVisibility</code> or <code>Targeting</code> .	422
Too many AWS account ids	You cannot provide more than 100 targeted buyer accounts.	422

Asynchronous Errors

Error code	Error message
INVALID_DELIVERY_OPTION_IDS	You provided invalid delivery option details. Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_VISIBILITY	You provided more than one delivery option for the public state. Provide only one public delivery option.
INVALID_VISIBILITY	You didn't provide a public delivery option. Provide one public delivery option.
AUDIT_ERROR	Varies based on MCO manual review.

Work with machine learning products using AWS Marketplace APIs

You can use the AWS Marketplace Catalog API to automate tasks for managing machine learning (ML)-based products. This section covers key operations for ML products using the Catalog API.

For information about creating machine learning products using the Catalog API, refer to [Listing your product in AWS Marketplace](#) in the *AWS Marketplace Seller Guide*.

Topics

- [Prerequisites for publishing ML product listings](#)
- [Add a new version](#)
- [Update version information](#)
- [Update pricing terms](#)
- [Restrict a version](#)

Prerequisites for publishing ML product listings

Ensure that you have the following before getting started with publishing your machine learning product listing:

- Registration as a seller in AWS Marketplace. For more information, refer to [Register as an AWS Marketplace seller](#).
- An IAM user with `AWSMarketplaceSellerFullAccess` permission.
- A publicly accessible Amazon Simple Storage Service (Amazon S3) bucket to host your company logo and EULA, if you provide one. You enter the URL for the S3 bucket in your `ChangeSet` JSON file.
- A valid SageMaker AI ARN of the model package or algorithm resource you want to list.
- A valid IAM role that has a trust relationship with the AWS Marketplace service principal and provides access to your package. For more information about the role, refer to *IAM role for the AWS Marketplace service principal* in the [Prerequisites](#) page for machine learning products.

Add a new version

You can use the Catalog API or the AWS Marketplace Management Portal to add a new version to a machine learning product. For more information about using the portal, refer to [Adding a new version](#) in the *AWS Marketplace Seller Guide*.

Note

For ML products, a version consists of a single delivery option, which represents the product you're making available. In the Catalog API, working with delivery options for ML products effectively modifies versions of your product.

When adding new instance types, include an `UpdatePricingTerms` change type in your change set to specify pricing for new instance types. For new products, `UpdatePricingTerms` must cover all supported instance types.

UpdatePricingDimensions is not required or supported for ML products, because dimensions are automatically generated for all supported instance types. For more information, refer to [the section called “Update pricing dimensions”](#).

Using StartChangeSet to add a version

To add a new version, call the StartChangeSet operation with the AddDeliveryOptions change type:

1. To validate your API call without creating a version, set Intent to VALIDATE.
2. For actual version creation, set Intent to APPLY.

Request syntax

```
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"
}

```

Required fields

- **Entity (object)**—required

Contains information about your ML product.

- **Identifier (string)**—required

Your product ID. For more information, see [Identifier](#).

- **Type (string)**—required

Specifies the delivery method (product type). It must be `MachineLearningProduct@1.0`.

- **DetailsDocument (object)**—required

Contains all details about the new version of your product.

- **Version (object)**—required

Details about the version being added.

- **VersionTitle (string)**—required

The title of the version, such as "Version 1.1" or "1.1". Buyers select versions from these titles.

- **ReleaseNotes (string)**—required

Detailed notes about this version. Must be under 30,000 characters.

- **DeliveryOptions (array)**—required

Array of delivery methods for your product version. Limited to one delivery option per version.

- **Details (object)**—required

- **SageMakerModelPackageDeliveryOptionDetails** or **SageMakerAlgorithmDeliveryOptionDetails (object)**

- **SageMakerModelPackageArn** or **SageMakerAlgorithmArn (string)**—required

Amazon Resource Name (ARN) of your model package or algorithm.

- **AccessRoleArn (string)**—required

IAM role ARN for AWS Marketplace to access the SageMaker resource.

- **SampleNotebookUrl (string)**—required

Link to sample Jupyter notebook with usage code. For more information, refer to a [sample notebook template](#) on GitHub.

- **RepositoryUrl (string)**—required

Git repository URL for notebook and sample data access. For more information, see a [sample Git repository](#) on GitHub.

- **UsageInstructions (string)**—required

Training information for algorithms or usage details for models.

- **InputProperties (object)**—required

- **Description (string)**—required

Description of model/algorithm inputs

- **Limitations (string)**

Input limitations

- **SampleInput (object)**—required

RealtimeInferenceUrl (string) | RealtimeInferenceText (string) | BatchTransformUrl (string) | BatchTransformText (string)

- **Parameters (Array<Object>)**

Name (string)—required | Description (string)—Required | Constraints (string) | Required (boolean)

- SageMakerCustomAttributes (Array<Object>)

Describes any [CustomAttributes](#) supported by your model.

Name (string)—required | Description (string)—Required | Constraints (string) | Required (boolean)

- OutputProperties (object)—required
 - Description (string)—required
 - SampleOutput (Array<Object>)—required

RealtimeInferenceUrl (string) | RealtimeInferenceText (string) | BatchTransformUrl (string) | BatchTransformText (string)

- Parameters (Array)

Name (string)—required | Description (string)—required | AlwaysReturned (boolean)

- RecommendedInstanceTypes (object)—required
 - BatchTransform (string)—Required
 - RealtimeInference (string)—Required
 - Training (string)—Required for SageMaker Algorithms only

Response syntax

When you submit the request, a change set is created and the API returns:

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

Change set processing

The change request enters a processing queue, where it undergoes several steps:

1. Validation: The system checks to ensure all information meets AWS Marketplace guidelines.

- Processing time: Few minutes to several hours
 - For validation errors, see [Change set status and errors](#).
2. Status monitoring: You can check the status of the request two ways.
 - Via AWS Marketplace Management Portal
 - Using the `DescribeChangeSet` operation
 3. Completion: When approved, the new version is added.

Errors

Asynchronous errors

The following errors are specific to `AddDeliveryOptions` actions in the AWS Marketplace Catalog API. These errors appear when you call `DescribeChangeSet` while a change set is being processed. For more information about using `DescribeChangeSet` to check the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	We couldn't locate the provided product. Provide a valid product.
DUPLICATE_VERSION_TITLE	The provided version title is already in use. Create a unique version title.
INVALID_RECOMMENDED_INSTANCE_TYPE	You didn't provide a valid instance type for [x]. Enter a valid instance type and try again. Valid types are: [valids]
INCOMPATIBLE_DELIVERY_OPTIONS	The delivery option you provided doesn't match your previous selection: [previous selection]. Update your delivery option and try again.
INVALID_ASSET_ARN	You didn't provide a valid ARN for SageMaker AlgorithmDeliveryOption. Enter a valid ARN and try again.

Error code	Error message
DUPLICATE_ASSET	You didn't provide a unique ARN for this product. Enter a unique ARN and try again.
ASSET_NOT_FOUND	We couldn't locate the ARN you provided. Verify that the ARN is correct and has the required permissions.
ASSET_VALIDATION_EXCEPTION	Unable to ingest SagemakerModelArn/ SagemakerAlgorithmArn [x] into AWS Sagemaker account
INVALID_ACCESS_ROLE	We couldn't locate the IAM role ARN you provided. Verify that the ARN is correct and try again.
UPDATE_PRICING_REQUIRED	UpdatePricingTerms is required.

Update version information

You can use the Catalog API to update details of an existing version of your machine learning product in AWS Marketplace.

Important

You cannot update the ARN for a version. If you need to modify the ARN, you must create a new version instead.

Using StartChangeSet to add a version:

- To update version information, call the StartChangeSet operation with the UpdateDeliveryOptions change type.

Request syntax

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

```
}  
]  
}
```

Required fields

- **Entity (object)**—required

Contains information about your ML product.

- **Identifier (string)**—required

Your product ID. For more information, see [Identifier](#).

- **Type (string)**—required

Specifies the delivery method (product type). It must be `MachineLearningProduct@1.0`.

- **DetailsDocument (object)**—required

Contains the updated version information.

- **Version (object)**—required

Defines version-specific information.

- **VersionTitle (string)**—required

The version identifier displayed to buyers, such as "Version 1.1" or "1.1". Buyers use this title to select versions for deployment.

- **ReleaseNotes (string)**—required

Detailed notes about this version, limited to 30,000 characters.

- **DeliveryOptions (array)**—required

Specifies delivery methods for your product version. Limited to one delivery option per version.

- **Id (string)**—required

Unique identifier for the `DeliveryOption`. Retrieve this using the `DescribeEntity` action on your product.

- **Details (object)**—required

Contains the delivery option specifications.

- `SageMakerModelPackageSubscriptionDetails` or `SageMakerAlgorithmSubscriptionDetails` (object)—required

Details of the delivery option.

- `SampleNotebookUrl` (string)—required

Sample Jupyter notebook link providing code for buyer usage.

- `RepositoryUrl` (string)—required

Git repository URL for cloning notebook and sample data.

- `UsageInstructions` (string)—required

For algorithms: training information. For models: additional usage information.

- `InputProperties` (object)—required

Details of the model/algorithm's input requirements.

- `Description` (string)—required

Description of required inputs.

- `Limitations` (string)

Any input limitations.

- `SampleInput` (object)—required

Sample input data for different operations.

- `RealtimeInferenceUrl` (string)

Sample input URL for realtime inference.

- `RealtimeInferenceText` (string)

Sample input text for realtime inference.

- `BatchTransformUrl` (string)

Sample input URL for batch transform jobs.

- `BatchTransformText` (string)

Sample input text for batch transform jobs.

- `Parameters` (Array<Object>)

Details for each input parameter.

- **Name (string)**—required

Name of the input parameter.

- **Description (string)**—required

Description of the input parameter.

- **Constraints (string)**

Parameter constraints (MinValue, MaxValue, AllowedValues, MinLength, MaxLength, Pattern, etc.).

- **Required (boolean)**

Indicates if the parameter is required. Default is false.

- **SageMakerCustomAttributes (Array<Object>)**

Details for supported CustomAttributes.

- **Name (string)**—required

Name of the custom attribute.

- **Description (string)**—required

Description of the custom attribute.

- **Constraints (string)**

Attribute constraints (MinValue, MaxValue, AllowedValues, MinLength, MaxLength, Pattern, etc.).

- **Required (boolean)**

Indicates if the attribute is required. Default is false.

- **OutputProperties (object)**—required

Details of the model/algorithm's output.

- **Description (string)**—required

Description of model/algorithm outputs.

- **SampleOutput (Array<Object>)**—required

Sample output data for different operations.

- `RealtimeInferenceUrl` (string)

Sample output URL for realtime inference.

- `RealtimeInferenceText` (string)

Sample output text for realtime inference.

- `BatchTransformUrl` (string)

Sample output URL for batch transform jobs.

- `BatchTransformText` (string)

Sample output text for batch transform jobs.

- `Parameters` (Array)

Details for each output parameter.

- `Name` (string)—required

Name of the output parameter.

- `Description` (string)—required

Description of the output parameter.

- `AlwaysReturned` (boolean)

Indicates if the parameter is always returned. Default is false.

- `RecommendedInstanceTypes` (object)—required

Recommended instance types for optimal performance.

- `BatchTransform` (string)—required

Recommended instance type for batch transform operations.

- `RealtimeInference` (string)—required

Recommended instance type for realtime inference operations.

- `Training` (string)—required

Recommended instance type for algorithm training operations. Required only for SageMaker Algorithms.

Response syntax

A successful request returns:

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

Change set processing

The change request enters a processing queue, where it undergoes several steps:

1. Validation: The system checks if all information meets AWS Marketplace guidelines.
 - Processing time: Few minutes to several hours
 - For validation errors, see [Change set status and errors](#).
2. Status monitoring: You can check the status of the request two ways.
 - Via AWS Marketplace Management Portal
 - Using the DescribeChangeSet operation
3. Completion: When approved, the new version is updated.

Errors

Asynchronous errors

Specific errors for UpdateDeliveryOptions actions can be retrieved using the DescribeChangeSet operation after the change set begins processing. For error details and troubleshooting, see [Change set status and errors](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use an existing limited or public product.

Error code	Error message
INVALID_DELIVERY_OPTION_ID	Provide delivery option IDs that can be found in the product. IDs not found: [x]
INCOMPATIBLE_DELIVERY_OPTION_STATUS	The delivery option cannot be updated because it's in restricted status. Try adding a new version instead.
INCOMPATIBLE_DELIVERY_OPTIONS	Product previously used [X] as delivery option, therefore all the upcoming delivery options should be of type [X]
INVALID_RECOMMENDED_INSTANCE_TYPE	Provide an existing, available instance type for [X] (X can be Batch Transform, Realtime Inference or ALgorithm Training)
DUPLICATE_VERSION_TITLE	The version title must be different from any other version titles of this product.
FIELD_NOT_ALLOWED_TO_CHANGE	Field X is not allowed to be changed.

Update pricing terms

You can use the Catalog API to update the pricing of your machine learning products in AWS Marketplace.

Note

You must price all supported instance types of your machine learning products. When creating your first version, the `UpdatePricingTerms` and `AddDeliveryOptions` change types with prices for all supported instances types in order to publish your product. When adding a new version to an existing product with new instance types that were not supported before, you must include those instance types in the `UpdatePricingTerms` and `AddDeliveryOptions` change types.

To update pricing terms, call the `StartChangeSet` operation with the `UpdatePricingTerms` change type.

Request syntax

```
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"
        }
    ]
}
]
},
]

```

}

Required fields

- **Entity (object) (required)** - Contains information about the offer of your ML product.
 - **Identifier (string) (required)** – The offer id for which you would like to update pricing terms. For more information, see [Identifier](#).
 - **Type (string) (required)** – The Type must be `Offer@1.0` for updating pricing.
- **DetailsDocument (object) (required)** – Details about the pricing terms of your machine learning product.
 - **PricingModel (string) (required)** - The pricing model for your product - you can choose between Usage, Free, and Contract (for private offers). For more information, see [Machine Learning Product Pricing](#).
- **Terms (array of objects) - An array of pricing terms that collectively define the overall pricing of your product.**
 - **Type (string) (required)** - The type of pricing term. Valid options are `UsageBasedPricingTerm` or `FreeTrialPricingTerm` or `FixedUpfrontPricingTerm` (for private offers).
 - **CurrencyCode (string) (required for UsageBasedPricingTerm or FixedUpfrontPricingTerm)** - the currency of the pricing term. Valid values are USD.
 - **RateCards (array of objects) (required for UsageBasedPricingTerm)** - The rate cards that define the pricing for your machine learning product. You must include one RateCard only within this array.
 - **RateCard (array of objects)** - The array of dimensions and rates for your machine learning product. All supported instance types must be priced in order to publish your product.
 - **DimensionKey** - The dimension you are pricing. Valid dimension keys are:
 - Instance type followed by the operation for hourly rates:
 - Instance type will start with `m1` followed by the instance name and size. For example, `m1.m4.xlarge`
 - Operation will be `m.i.b` for batch transform, `m.i.r` for hourly real-time inference, and `a.t` for algorithm training.
 - Examples: `m1.m4.4xlarge.m.i.b` for batch transform, `m1.m4.xlarge.m.i.r` for real-time inference, or `m1.m4.16xlarge.a.t` for algorithm training
 - `m.i.c` for real-time per inference pricing.

- **Price** - The rate of the dimension. The rate is either hourly or per-inference, depending on the `dimensionKey`.
- **Duration** (string) (required for `FreeTrialPricingTerm` or `FixedUpfrontPricingTerm`) - The duration of your free trial or contract. For free trial valid values are between X and Y. For contract, valid values are between X and Y.
- **Grants** (array of objects) - Details about which `dimensionKeys` are eligible for free trial. All supported `dimensionKeys` must be provided.
- **DimensionKey** - The dimension to include as part of the free trial offer.
 - Instance type followed by the operation for hourly rates:
 - Instance type starts with `m1` followed by the instance name and size e.g. `m1.m4.xlarge`
 - Operation is `m.i.b` for batch transform, `m.i.r` for hourly real-time inference, or `a.t` for algorithm training.
 - Examples: `m1.m4.4xlarge.m.i.b` for batch transform, `m1.m4.xlarge.m.i.r` for real-time inference, or `m1.m4.16xlarge.a.t` for algorithm training
 - `m.i.c` for real-time per inference pricing.

Response syntax

A successful request returns:

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

Change set processing

The change request enters a processing queue, where it undergoes several steps:

1. **Validation:** The system checks to ensure all information meets AWS Marketplace guidelines.
 - Processing time: Few minutes to several hours
 - For validation errors, see [Change set status and errors](#).
2. **Status monitoring:** You can check the status of the request two ways.
 - Via AWS Marketplace Management Portal

- Using the DescribeChangeSet operation
3. Completion: When approved, the pricing terms are updated.

Errors

Asynchronous errors

Specific errors for UpdatePricingTerms actions can be retrieved using the DescribeChangeSet operation after the change set begins processing. For error details and troubleshooting, see [Change set status and errors](#).

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use an existing limited or public product.

Restrict a version

You can use the Catalog API to restrict a version of your machine learning product in AWS Marketplace. When restricted, new buyers cannot access that version. Existing subscribers retain access to restricted versions. AWS Marketplace guidelines require continued support for existing buyers for 90 days after restriction.

Important

At least one unrestricted version must remain available. You cannot restrict the last publicly available version of a product.

To restrict a version:

- To restrict a version, call the StartChangeSet operation with the RestrictDeliveryOptions change type.

Request syntax

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

Required fields

Required fields

- Entity (object)—required

Contains information about your ML product.

- Identifier (string)—required

Your product ID. For more information, see [Identifier](#).

- Type (string)—required

Specifies the delivery method (product type). It must be MachineLearningProduct@1.0.

- DetailsDocument (object)—required

Contains the updated version information.

- DeliveryOptionIds (array of objects)—required

List of DeliveryOption IDs for the versions you want to restrict. Retrieve the unique identifier for each DeliveryOption by calling the DescribeEntity action on the version you're restricting.

Response syntax

A successful request returns:

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

1. Validation: The system checks if all information meets AWS Marketplace guidelines.

- Processing time: Few minutes to several hours
- For validation errors, see [Change set status and errors](#).

2. Status monitoring: You can check the status of the request two ways.

- Via AWS Marketplace Management Portal
- Using the DescribeChangeSet operation

3. Completion: When approved, the new version is restricted.

Errors

Asynchronous errors

The following errors may occur during change set processing and can be retrieved using the DescribeChangeSet operation:

Error code	Error message
INCOMPATIBLE_PRODUCT_STATUS	Use an existing public or limited product
MISSING_DELIVERY_OPTION_IDS	Provide delivery option from existing list of IDs.
INVALID_DELIVERY_OPTION_IDS	Provide delivery option IDs that can be found in the product. IDs not found: [x]
INVALID_DELIVERY_OPTION_STATUS	The delivery option IDs [invalid_ids] are invalid. Provide delivery options in the public state.

Error code	Error message
ALL_DELIVERY_OPTIONS_RESTRICTED	Provide fewer delivery options to restrict as at least one must remain in public state.

Work with offers using the AWS Marketplace APIs

An offer defines the terms and rules governing the purchase and consumption of a product. It encompasses a collection of agreement terms between two parties. The accepted offer terms are then translated into an agreement, serving as proof of the transaction.

The following sections provide information on how to manage offers as a seller or channel partner.

Topics

- [Work with private offers using the AWS Marketplace APIs](#)
- [Work with resale authorizations using the AWS Marketplace APIs](#)
- [Work with channel partner private offers using the AWS Marketplace APIs](#)
- [Work with renewals](#)

Work with private offers using the AWS Marketplace APIs

You can use the AWS Marketplace Catalog API to automate tasks for working with offers.

While the *product* describes what is being sold in AWS Marketplace, the *offer* describes the terms and rules of how a product is purchased and consumed. AWS Marketplace products can have multiple offers sold by different sellers. Each AWS Marketplace offer, however, can only be created for one product. An *offer* contains a collection of agreement terms between two parties. The accepted offer terms are translated into an agreement as proof of a transaction.

There are two types of offers:

- **Private offers** are for sellers and buyers to negotiate pricing. Sellers sign an end-user license agreement (EULA) for software purchases in AWS Marketplace. An offer is visible only to a specified buyer. For more information, see [Private offers](#) in the *AWS Marketplace Seller Guide*.
- **Public offers** are for global purchasing programs. Sellers identify customers based on available programs and geographical locations, which makes the offer accessible only to specific customers.

See the following resources:

- For working code examples, see [Manage offers with API](#) in the *AWS Marketplace seller workshop*.
- For API request code examples, see [Python](#) and [Java](#) examples in *AWS Samples* on GitHub.
- For a video on creating private offers, see [Create a Private Offer Using the AWS Marketplace Catalog API](#) on YouTube.
- For a video on updating AMI pricing, see [Update AMI Product Pricing Using the AWS Marketplace Catalog API](#) on YouTube.

The following topics describe how to use the Catalog API to create and update offers:

Topics

- [Create an offer](#)
- [Create a replacement offer](#)
- [Update offer information](#)
- [Update targeting configuration](#)
- [Update refund policy](#)
- [Update legal resources](#)
- [Update pricing](#)
- [Update the discoverability of the offer](#)
- [Define the expiration date of agreements created using the offer](#)
- [Update payment schedule details](#)
- [Modify renewal options](#)
- [Publish an offer](#)
- [Describe existing offer details](#)

Create an offer

You can use the Catalog API to create a new offer in AWS Marketplace. If your request processes successfully, the AWS Marketplace Catalog API creates a `Draft`, which is an incomplete offer that's invisible to buyers. To complete an offer, use the `Update` change type. When the offer is complete, use the [ReleaseOffer](#) change type to create and release it. Releasing an offer validates it and makes it visible to buyers in AWS Marketplace.

To create a new offer, call the `StartChangeSet` API operation with the `CreateOffer` change type, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the `CreateOffer` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **ProductId** (string) (required) – The unique identifier of the product being offered.
 - **Name** (string) (optional) – The name associated with the offer for better readability to you and your customers. It is displayed as a part of the Agreement information as well.
 - **OfferSetId** (string) (optional) – The ID of the offer set to associate this offer with. Only specify this field when creating an offer that will be part of an offer set. If `OfferSetId` is not provided, an individual offer will be created that can be purchased standalone. Note that specifying an `OfferSetId` during offer creation only indicates your intent to associate the offer with that offer set. To complete the association, you must [use the AssociateOffers change type](#) after the offer is created.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

If the Status is SUCCEEDED, then a new OfferId is generated.

The response looks like the following.

```
{
  "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": []
    }
  ]
}
```

Synchronous Validations

The following schema validations are specific to `CreateOffer` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
ProductId	Required	422
	Length must be between 1 and 50 characters	
	Must not contain illegal characters (\, <, >)	
ProductId	Required User must be authorized to create offer for the given product	403
ProductId	Required Must be an existing product in the catalog or being created in the same change set	404
Name	Optional	422
	Length must be between 1 and 150 characters	
	Must not contain illegal characters (\, <, >)	

Asynchronous Errors

The following errors are specific to `CreateOffer` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT	Use an active product in Limited or Public state.
INCOMPATIBLE_PRODUCT	Managing offers for your chosen product type isn't currently supported in the AWS Marketplace Catalog API.
INCOMPATIBLE_PRODUCT	Managing offers for the product isn't currently supported in the AWS Marketplace Catalog API.
INCOMPATIBLE_PRODUCT	OfferSetId isn't supported in offers for the product.
INCOMPATIBLE_PRODUCT	CreateOffer change type can't be invoked to create an offer for the product. Use CreateOfferUsingResaleAuthorization change type.

Create a replacement offer

You can use the Catalog API to create a replacement offer (also known as an agreement-based offer) in AWS Marketplace.

If your request has been processed successfully, AWS Marketplace Catalog API will have an offer in `Draft` state generated for you, which is an incomplete offer and not visible to buyers on AWS Marketplace. You will use `Update` change types to complete the offer. After the offer is completed, you will use [ReleaseOffer](#) change type to complete offer creation process and release the offer, which will validate the entire offer and make your offer visible to buyers on AWS Marketplace. From there, the buyer has the option to accept the replacement offer or to continue to operate under the original agreement.

To create a replacement offer, call the `StartChangeSet` API operation with the `CreateReplacementOffer` change type and provide a pre-existing agreement id, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the `CreateReplacementOffer` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **AgreementId** (string) (required) – The unique identifier for the current agreement to be replaced.
 - **Name** (string) (optional) – The name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `CreateReplacementOffer` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP Code
AgreementId	Required	422
	Length must be between 1 and 64 characters	
AgreementId	Required User must be authorized to create offer for the given agreement	403
Name	Optional	422
	Length must be between 1 and 150 characters	
	Must not contain invalid characters (\, <, >)	

Asynchronous Errors

The following errors are specific to `CreateReplacementOffer` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT	Replacement offers aren't supported for the product.
INCOMPATIBLE_AGREEMENT	CreateReplacementOffer change type can't be invoked to create a replacement offer for the agreement. Use CreateReplacementOfferUsing ResaleAuthorization change type.

Update offer information

You can use the Catalog API to update the offer information in AWS Marketplace.

To update the offer information, call the `StartChangeSet` API operation with the `UpdateInformation` change type, as shown in the following example. All other information will remain unchanged.

Request Syntax

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

```
]
}
```

Provide information for the fields to add the `UpdateInformation` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **Name** (string) (optional) – Name associated with the offer for better readability. It is displayed as part of agreement information.
 - **Description** (string) (optional) – A free-form text that is meant to be used only by you and will never be visible to buyers.
 - **PreExistingAgreement** (object) (optional) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
 - **AcquisitionChannel** (string) (required) – Indicates if the existing agreement was signed outside AWS Marketplace or within AWS Marketplace.

Possible values: `External`, `AwsMarketplace`
 - **PricingModel** (string) (required) – Indicates which pricing model the existing agreement uses.

Possible values: `Contract`, `Usage`, `Byo1`, `Free`

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Properties	At least one of the following properties must be provided	422
Name	Optional Length must be between 1 and 150 characters Must not contain illegal characters (\, <, >)	422
Description	Optional Length must be between 1 and 255 characters	422
PreExistingAgreement	Optional Can be null to remove PreExistingAgreement from offer	422
PreExistingAgreement.PricingModel	Required Can be one of these values: [Byo1, Free, Usage, Contract]	422

Input field	Validation rule	HTTP code
PreExistingAgreement.AcquisitionChannel	Required Can be one of these values: [AwsMarketplace , External]	422

Asynchronous Errors

The following errors are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRE_EXISTING_AGREEMENT	PreExistingAgreement can't be changed after the offer is released.

Update targeting configuration

You can use the Catalog API to update the targeting configuration of your offer in AWS Marketplace.

All existing targeting options that aren't included in the latest request and will be removed from the offer.

Note

An offer can optionally include PositiveTargeting or NegativeTargeting, but not both.

Positive Targeting options:

- **Country codes only** – Creates a public offer available to buyers in the specified countries.
- **Buyer accounts only** – Creates a private offer targeted to specific AWS accounts.
- **Both country codes and buyer accounts** – Creates a private offer where targeted accounts can only accept the offer if they are located in one of the specified countries.

Negative Targeting options:

- **Country codes** – Excludes buyers from the specified countries. This creates a public offer available to all countries except those listed.

To update the targeting configuration of your offer, call the `StartChangeSet` API operation with the `UpdateTargeting` change type, as shown in the following example.

Request Syntax

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



```
}
```

Provide information for the fields to add the `UpdateTargeting` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **PositiveTargeting** (object) (optional) – Positive targeting defines the criteria which any buyer's profile should fulfill in order to be allowed to access the offer. This field is optional, but at least one targeting option should be provided when this field is present.
 - **CountryCodes** (array of strings) (optional) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a country, this field should be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code, using this format: ISO 3166-1 alpha-2.
 - **BuyerAccounts** (array of strings) (optional) – List as an option to allow targeting based on AWS accounts (also known as Private Offer). If the intention is to not target the offer to an AWS account, this field should be omitted.
- **NegativeTargeting** (object) (optional) – Negative targeting defines the criteria which any customer's profile should fulfill to be restricted to access the offer. Although this field is optional, at least one targeting option should be provided when this field is present.
 - **CountryCodes** (array of strings) (required) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a specific country, then this field should be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code using this format: ISO 3166-1 alpha-2.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdateTargeting` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
NegativeTargeting	Optional Can have either one of the following : [CountryCodes]	422
NegativeTargeting.CountryCodes	Optional List size must be between 1 and 244 Country codes must be valid (ISO 3166-1 alpha-2)	422
PositiveTargeting	Optional Can have either one of the following : [CountryCodes , BuyerAccounts]	422
PositiveTargeting.BuyerAccounts	Optional List size must be between 1 and 26 AWS account IDs must be in valid format (12-digit number)	422
PositiveTargeting.CountryCodes	Optional	422

Input field	Validation rule	HTTP code
	List size must be between 1 and 244	
	Country codes must be valid (ISO 3166-1 alpha-2)	

Asynchronous Errors

The following errors are specific to UpdateTargeting actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_BUYER_ACCOUNTS	Provide valid buyer accounts. Invalid accounts: [x].
INVALID_COUNTRY_CODES	Provide supported country codes.
INVALID_TARGETING	Use either negative or positive targeting on the same attribute.
INCOMPATIBLE_PRODUCT	Country-based targeting isn't supported for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide BuyerAccounts that are compatible with the ResaleAuthorization.
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is expired.
INCOMPATIBLE_TARGETING	Targeting can't be updated on a replacement offer. If the buyer isn't associated with the provided AgreementId, then create a new

Error code	Error message
	private offer by providing an AgreementId associated with the buyer.
TOO_MANY_BUYER_ACCOUNTS	Provide BuyerAccounts within the allowed limits.
INCOMPATIBLE_TARGETING	BuyerAccounts can't be removed after the offer is released.
INCOMPATIBLE_TARGETING	BuyerAccounts can't be added after the offer is released.
MISSING_COUNTRY_CODES	Provide PositiveTargeting with CountryCodes: [x].
INCOMPATIBLE_COUNTRY_CODES	Provide CountryCodes that are compatible.
INCOMPATIBLE_BUYER_ACCOUNTS	Provide BuyerAccounts that are compatible with the agreement.

Update refund policy

You can use the Catalog API to update the refund policy of your offer in AWS Marketplace.

This change doesn't affect existing agreements. The support terms that aren't included in the latest request will be removed from the offer.

To update the refund policy, call the `StartChangeSet` API operation with the `UpdateSupportTerms` change type, as shown in the following example.

Request Syntax

```
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"
      }
    ]
  }
}
```

Provide information for the fields to add the `UpdateSupportTerms` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **Terms** (array of structures) (required) – List of support terms that you would like to update. Accepted support terms are:
 - **SupportTerm** (object) (required) – Defines the customer support available for the acceptors when they purchase the software.
 - **Type** (string) (required) – Type of the term being updated. This is the object value: `"SupportTerm"`.
 - **RefundPolicy** (string) (required) – Free-text field about the refund policy description that will be shown to customers as is on the website and console.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

```
{
```

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to UpdateSupportTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
Terms[].RefundPolicy	Required Length must be between 1 and 500 Cannot lead or end with spaces	422
Terms[].Type	Required Can only be SupportTerm	422

Asynchronous Errors

The following errors are specific to UpdateSupportTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT	SupportTerm isn't supported in private offers for the product.
INCOMPATIBLE_TERMS	SupportTerm isn't supported for free trial offers.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.

Update legal resources

You can use the Catalog API to replace the existing legal documents, such as an end user license agreement (EULA). The legal terms that aren't included in the latest request will be removed from the offer.

To update legal resources of your offer, call the `StartChangeSet` API operation with the `UpdateLegalTerms` change type, as shown in the following example.

Request Syntax

```
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"
      }
    ]
  }
}

```

Provide information for the fields to add the UpdateLegalTerms change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always Offer@1.0.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **LegalTerm** (object) (required) – Defines the list of text agreements to be proposed to the acceptors. One example of such an agreement is the end user license agreement (EULA).
 - **Type** (string) (required) – Type of the term being updated. This is the object value: "LegalTerm".
 - **Documents** (array of structures) (required) – List of references to legal resources to be proposed to the buyers. One example of such a resource is the end user license agreement (EULA). Each reference is made up of a Type and a URL:
 - **Type** (string) (required) – Type of document. Available document types are:
 - **CustomEula** – A custom EULA provided by you as seller. Either a public S3 URL or a [presigned URL](#) is required for this document type.
 - **StandardEula** – Standard Contract For AWS Marketplace (SCMP). For more information about SCMP, see the AWS Marketplace Seller Guide. You don't provide a URL for this type because it is managed by AWS Marketplace.
 - **Url** (string) (conditionally required) – A URL to the legal document for buyers to read. Required when Type is one of the following [CustomEula].
 - **Version** (string) (conditionally required) – Version of standard contracts provided by AWS Marketplace. Required when Type is [StandardEula]. Available version:

- **2022-07-14** – This version of the Standard Contract for AWS Marketplace is available from this Amazon S3 bucket: <https://s3.amazonaws.com/aws-mp-standard-contracts/Standard-Contact-for-AWS-Marketplace-2022-07-14.pdf>

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdateLegalTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
	Only <code>LegalTerm</code> is allowed in the list	
	List size must be 1	
Terms[].Type	Required	422
	Can only be <code>LegalTerm</code>	

Input field	Validation rule	HTTP code
Terms[].LegalTerm.Documents	Required	422
Terms[].LegalTerm.Documents [].Type	Required Allowed values: • CustomEula • StandardEula	422
Terms[].LegalTerm.Documents[].Url	Required and must be a valid URL when Type is CustomEula	422
Terms[].LegalTerm.Documents [].Version	Required and must be a valid Version when Type is StandardE ula Valid StandardEula versions: ["2019-04-24", "2022-07-14"]	422

Asynchronous Errors

The following errors are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_LEGAL_DOCUMENTS	Provide URLs for legal documents stored in accessible S3 buckets.

Error code	Error message
INVALID_LEGAL_DOCUMENTS	Only the most recent version of StandardEula is supported for new offers.
INVALID_LEGAL_DOCUMENTS	Provide legal documents in the supported file formats.
INVALID_LEGAL_DOCUMENTS	Provide legal documents using the supported document types.
LIMIT_EXCEEDED_LEGAL_DOCUMENT_SIZE	Provide legal documents within the allowed size limits.
INVALID_LEGAL_DOCUMENTS	LegalTerm contains password-protected document(s). Provide accessible documents in LegalTerm.
INVALID_LEGAL_DOCUMENTS	LegalTerm contains invalid PDF document(s). Provide accessible documents in LegalTerm.

Update pricing

You can use the Catalog API to replace the existing pricing terms completely. The pricing terms that aren't included in the latest request will be removed from the offer.

To update pricing terms for your offer, call the `StartChangeSet` API operation with the `UpdatePricingTerms` change type, as shown in the following example.

Note

The following request syntax combines multiple examples. This combination doesn't work as a valid payload. For example, a `Terms` array can't include both the term type `FixedUpfrontPricingTerm` and the term type `ConfigurableUpfrontPricingTerm`. For examples of how different term types are combined for different pricing use cases, see [Manage offers with API](#) in the *AWS Marketplace seller workshop*.

Note

For SaaS products with Free pricing model, you must include either `UsageBasedPricingTerm` or `ConfigurableUpfrontPricingTerm` with at least one `RateCard` (dimension) where all prices are set to \$0.00. This requirement is unique to SaaS products.

Request Syntax

```
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
    }
  ]
}
]
}
}
]
}
}
}
}

```

Provide information for the fields to add the UpdatePricingTerms change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always Offer@1.0.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **PricingModel** (string) (required) – Pricing model for your offer. Possible values for pricing model are:
 - **Usage** – Usage-based pricing model where buyers will be billed for their usage of your product.
 - **Contract** – Contract-based pricing model where buyers are either billed in advance for the use of your product, or offered a flexible payment schedule. Buyers can also pay for an additional usage above their contract.
 - **Free** – Free pricing model where buyers will not be charged for usage of product. When using this pricing model no pricing terms or payment schedule term can have non-zero rates.

- **Byol** – Byol pricing model where buyers will bring their own license for usage of the product.
- **Terms** (array of structures) (required) – List of pricing terms that you want to update.
Supported pricing terms are:
 - **FreeTrialPricingTerm** (object) – Defines a short-term free pricing model where the buyers are not charged anything within a specified limit.
 - **Type** (string) – Type of the term being updated. This is the object value: "FreeTrialPricingTerm".
 - **Duration** (string) – Duration of the free trial period.
 - **Grants** (array of structures) – Entitlements that will be granted to the acceptor of a free trial as part of an agreement execution.
 - **DimensionKey** (string) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
 - **MaxQuantity** (integer) (optional) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If MaxQuantity is not provided, the buyer will be able to use an unlimited amount of the given dimension.
 - **UsageBasedPricingTerm** (object) – Defines a pay-as-you-go (PAYG) pricing model where the customers are charged based on product usage.
 - **Type** (string) (required) – Category of the term being updated. This is the object value: UsageBasedPricingTerm.
 - **CurrencyCode** (string) – Defines the currency for prices mentioned in this term. Currently, only USD is supported.
 - **RateCards** (array of structures) – List of rate cards.
 - **RateCard** (array of structures) – A rate card defines the per-unit rates for the product dimensions.
 - **DimensionKey** (string) – Dimension that the given entitlement applies. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
 - **Price** (string) – Per-unit price for the product dimension that will be used for calculating the amount to be charged to the buyer.
 - **ConfigurableUpfrontPricingTerm** (object) – Defines pre-paid payment model which allows buyers to configure the entitlements that they want to purchase and the duration of the

entitlements. You can update the list of rates for each contract duration and entitlements for each dimension.

- **Type** (string) (required) – Type of the term being updated. This is the object value: `ConfigurableUpfrontPricingTerm`.
- **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. For public offers, only USD is supported. For private offers, USD, AUD, EUR, GBP, and JPY are supported.
- **RateCards** (array of structures) (required) – List of rate cards.
 - **Selector** (object) (required) – Selector is used to differentiate between the mutually exclusive rate cards in the same pricing term, to be selected by the buyer.
 - **Type** (string) (required) – Category of Selector. At this time, only `Duration` is supported.
 - **Value** (string) (required) – Contract duration. This field supports the ISO 8601 format.
 - **RateCard** (array of structures) (required) – A rate card defines the per-unit rates for the product dimensions.
 - **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
 - **Price** (string) (required) – Per-unit price for the product dimension which will be used for calculating the amount to be charged to the buyer.
- **Constraints** (object) (required) – Defines constraints on how the term can be configured by acceptors.

 Note

Currently, **MultipleDimensionSelection** and **QuantityConfiguration** values need to be same.

- **MultipleDimensionSelection** (string) (required) – Determines if buyers are allowed to select multiple dimensions in the rate card. Possible values are `Allowed` and `Disallowed`.
- **QuantityConfiguration** (string) (required) – Determines if acceptors are allowed to configure quantity for each dimension in rate card. Possible values are `Allowed` and `Disallowed`.

- **ByolPricingTerm** (object) – Enables you and your customers to move your existing agreements to AWS Marketplace. The customer won't be charged for product usage in AWS Marketplace because they already paid for the product outside of AWS Marketplace.
- **Type** (string) (required) – Type of the term being updated. This is the object value: `ByolPricingTerm`.
- **RecurringPaymentTerm** (object) – Defines a pricing model where customers are charged a fixed recurring price at the end of each billing period.
 - **Type** (string) (required) – Type of the term being updated. This is the object value: `RecurringPaymentTerm`.
 - **BillingPeriod** (string) (required) – Defines the recurrence at which buyers are charged. Only `Monthly` is supported today.
 - **Price** (string) (required) – Amount charged to the buyer every billing period.
 - **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. Currently, only USD is supported.
- **FixedUpfrontPricingTerm** (object) – Defines a pre-paid pricing model where the customers are charged a fixed upfront amount.
 - **Type** (string) (required) – Type of the term being updated. This is the object value: `FixedUpfrontPricingTerm`.
 - **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. For public offers, only USD is supported. For private offers, USD, AUD, EUR, GBP, and JPY are supported.
 - **Price** (string) (required) – Fixed amount to be charged to the customer when this term is accepted.
 - **Grants** (array of structures) (required) – Entitlements that will be granted to the acceptor of fixed upfront as part of agreement execution.
 - **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
 - **MaxQuantity** (integer) (required) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If `MaxQuantity` is not provided, the buyer will be able to use an unlimited amount of the given dimension.
 - **Duration** (string) (optional) – Defines the duration that the term remains active. This field supports the ISO 8601 format.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdatePricingTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
PricingModel	Required Allowed pricing models: ["Byol", "Free", "Usage", "Contract"]	422
Terms	Required Allowed Terms: ["ConfigurableUpfrontPricingTerm", "ByolPricingTerm", "FreeTrialPricingTerm", "UsageBasedPricingTerm", "RecurringPaymentTerm", "FixedUpfrontPricingTerm"]	422
Terms[].ByolPricingTerm	Required	422

Input field	Validation rule	HTTP
Terms[].ByolPricingTerm.Type	Required Can only be "ByolPricingTerm"	422
Terms[].ConfigurableUpfrontPricingTerm	Required	422
Terms[].ConfigurableUpfrontPricingTerm.Type	Required Can only be "ConfigurableUpfrontPricingTerm"	422
Terms[].ConfigurableUpfrontPricingTerm.CurrencyCode	Required Supported currencies: ["USD", "AUD", "EUR", "GBP", "JPN"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards	Required List size must be between 1 and 5	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints	Required	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints.MultipleDimensionSelection	Required Allowed values: ["Allowed", "Disallowed"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Constraints.QuantityConfiguration	Required Allowed values: ["Allowed", "Disallowed"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard	Required List size must be between 1 and 800	422

Input field	Validation rule	HTTP
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard[].DimensionKey	Required Length must be between 1 and 100	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].RateCard[].Price	Required Data type is "String" Non-negative decimals with up to 3 decimal places supported	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Selector	Required	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Selector.Type	Required Allowed values: ["Duration"]	422
Terms[].ConfigurableUpfrontPricingTerm.RateCards[].Selector.Value	Required Expected format per Selector type: ISO 8601 duration	422
Terms[].FixedUpfrontPricingTerm	Required	422
Terms[].FixedUpfrontPricingTerm.Type	Required Can only be "FixedUpfrontPricingTerm"	422
Terms[].FixedUpfrontPricingTerm.CurrencyCode	Required Supported currencies: ["USD", "AUD", "EUR", "GBP", "JPN"]	422
Terms[].FixedUpfrontPricingTerm.Duration	Required Expected format per Selector type: ISO 8601 duration	422

Input field	Validation rule	HTTP
Terms[].FixedUpfrontPricingTerm.Grants	Required List size must be between 1 and 200	422
Terms[].FixedUpfrontPricingTerm.Grants[].DimensionKey	Required Length must be between 1 and 100	422
Terms[].FixedUpfrontPricingTerm.Grants[].MaxQuantity	RequiredValue must be greater than 0	422
Terms[].FixedUpfrontPricingTerm.Price	Required Data type is "String" Non-negative decimals with up to 3 decimal places supported	422
Terms[].FreeTrialPricingTerm	Required	422
Terms[].FreeTrialPricingTerm.Type	Required Can only be "FreeTrialPricingTerm"	422
Terms[].FreeTrialPricingTerm.Duration	Required Expected format: ISO 8601 duration	422
Terms[].FreeTrialPricingTerm.Grants	Required List size must be between 1 and 800	422
Terms[].FreeTrialPricingTerm.Grants[].DimensionKey	Required Length must be between 1 and 100	422
Terms[].FreeTrialPricingTerm.Grants[].MaxQuantity	Optional Value must be greater than 0	422

Input field	Validation rule	HTTP
Terms[].RecurringPaymentTerm	Required	422
Terms[].RecurringPaymentTerm.Type	Required Can only be "RecurringPaymentTerm"	422
Terms[].RecurringPaymentTerm.BillingPeriod	Required Allowed values: ["Monthly"]	422
Terms[].RecurringPaymentTerm.CurrencyCode	Required Supported currencies: ["USD"]	422
Terms[].RecurringPaymentTerm.Price	Required Data type is "String" Non-negative decimals with up to 3 decimal places supported	422
Terms[].UsageBasedPricingTerm	Required	422
Terms[].UsageBasedPricingTerm.Type	Required Can only be "UsageBasedPricingTerm"	422
Terms[].UsageBasedPricingTerm.CurrencyCode	Required Supported currencies: ["USD"]	422
Terms[].UsageBasedPricingTerm.RateCards	Required Must be size of 1	422
Terms[].UsageBasedPricingTerm.RateCards[].RateCard	Required List size must be between 1 and 800	422

Input field	Validation rule	HTTP
Terms[].UsageBasedPricingTerm.RateCards[].RateCard[].DimensionKey	Required Length must be between 1 and 100	422
Terms[].UsageBasedPricingTerm.RateCards[].RateCard[].Price	Required Data type is "String" Non-negative decimals with up to 8 decimal places supported	422

Asynchronous Errors

The following errors are specific to UpdatePricingTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
DUPLICATE_DIMENSION_KEYS	Provide Grants with a unique list of dimension keys in [x].
DUPLICATE_DIMENSION_KEYS	Provide RateCard with a unique list of dimension keys in [x].
DUPLICATE_SELECTORS	Provide a unique list of Selectors in ConfigurableUpfrontPricingTerm.
DUPLICATE_TERM_TYPES	Provide a unique list of term types.
INCOMPATIBLE_AGREEMENT	The following terms can't be removed from the replacement offer: [x, y, z].
INCOMPATIBLE_AGREEMENT	The following terms can't be added to the replacement offer: [x, y, z].

Error code	Error message
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the CurrencyCode.
INCOMPATIBLE_PRODUCT	Usage pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Contract pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Byol pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Free pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	[x] isn't supported in an offer for the product.
INCOMPATIBLE_PRODUCT	Provided payment and pricing terms are incompatible.
INCOMPATIBLE_PRODUCT	Use existing, available dimensions in the product in [x].
INCOMPATIBLE_PRODUCT	FreeTrialPricingTerm as the offer's only pricing term isn't supported for the product.
INCOMPATIBLE_PRODUCT	The following terms aren't supported for the product: [x,y,z].
INCOMPATIBLE_PRODUCT	Replacement offers are only supported for contract pricing model.
INCOMPATIBLE_PRODUCT	Provide pricing term(s) that are compatible with the product dimensions. Incompatible pricing terms: [x,y,z].

Error code	Error message
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Allowed in ConfigurableUpfrontPricingTerm for usage pricing model.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Disallowed in ConfigurableUpfrontPricingTerm for usage pricing model.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	QuantityConfiguration in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	MultipleDimensionSelection in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.
INCOMPATIBLE_RATES	Set all charge amounts and prices to zero (0) when using Free pricing model.
INCOMPATIBLE_RATES	Only zero (0) prices are allowed in UsageBasedPricingTerm for a free trial offer for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide the same CurrencyCode that is specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure Duration in FixedUpfrontPricingTerm matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [x, y, z].

Error code	Error message
INCOMPATIBLE_SELECTOR_DURATION	Durations aren't allowed to be removed from rate cards in ConfigurableUpfrontPricingTerm after the offer released.
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.
INCOMPATIBLE_TERMS	[x] isn't supported together with the following terms: [y,z].
INCOMPATIBLE_TERMS	The following terms can't be added after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	The following terms can't be removed after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	[x] isn't supported for private offers.
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm that grants unlimited usage: [x,y,z].
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm for the product: [x,y,z].
INCOMPATIBLE_TERMS	Provide zero (0) price for FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
INCOMPATIBLE_TERMS	The following terms aren't compatible with the PricingModel: [x,y,z].
INCOMPATIBLE_TERMS	FixedUpfrontPricingTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.

Error code	Error message
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide duration between [x] and [y] months.
INVALID_AGREEMENT_DURATION	Ensure duration granularity is at the day level for metered dimensions.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INVALID_DURATION	Ensure Duration in FreeTrialPricingTerm is within the allowed range.
INVALID_DURATION	Provide Duration in FixedUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
INVALID_DURATION	Provide duration between [x] and [y] months.
INVALID_DURATION	Ensure duration granularity is at the day level for metered dimensions.
INVALID_GRANTS	Provide the same MaxQuantity for all Grants in FreeTrialPricingTerm.
INVALID_GRANTS	Provide Grants for all available metered dimensions in FreeTrialPricingTerm.

Error code	Error message
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] can't be updated until [y] because you have requested a price increase in the past 120 days. To cancel your previous price increase request or for more information, contact the AWS Marketplace Managed Catalog Operations Team.
INVALID_PRICE_CHANGE	Price increase and dimension addition in [x] isn't supported in the same request. Add dimensions first.
INVALID_PRICE_CHANGE	Price increase and decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_PRICE_CHANGE	Price increase in RecurringPaymentTerm and price decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_PRICE_CHANGE	Price decrease in RecurringPaymentTerm and price increase in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.

Error code	Error message
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm is missing one or more-dimension keys for duration [x]. Provide prices for the same set of dimension keys for all durations.
INVALID_RATE_CARD	Provide a rate card for only metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Rates can't be removed from [x]. Provide prices for all dimensions in the existing rate card.
INVALID_RATE_CARD	Provide dimensions that have the same unit in [x].
INVALID_RATE_CARD	Provide either all metered or all entitled dimensions in [x].
INVALID_RATE_CARD	Provide only entitled dimensions in [x].
INVALID_RATE_CARD	Provide usage based rates for all available metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Provide usage based rates for all free trial dimensions.
INVALID_RATE_CARD	Provide prices with up to 8 decimal places in UsageBasedPricingTerm.
INVALID_RATE_CARD	The combination of Dimensions in rate card is invalid in UsageBasedPricingTerm for the product.
INVALID_SELECTOR_DURATION_VALUE	Provide duration between [x] and [y] months.

Error code	Error message
INVALID_SELECTOR_DURATION_VALUE	Ensure duration granularity is at the day level for metered dimensions.
INVALID_SELECTOR_DURATION_VALUE	Ensure Duration in ConfigurableUpfrontPricingTerm is within the allowed range.
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations.
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations or a single custom duration.
INVALID_SELECTOR_DURATION_VALUE	Provide Duration in ConfigurableUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
MISSING_DURATION	Provide Duration in FixedUpfrontPricingTerm.
MISSING_MANDATORY_TERMS	FixedUpfrontPricingTerm is only supported when paired with ByolPricingTerm or PaymentScheduleTerm.
MISSING_MANDATORY_TERMS	Provide at least one of [x,y,z].
MISSING_MANDATORY_TERMS	Provide a ByolPricingTerm when using Byol pricing model.
TOO_MANY_GRANTS	Provide up to [x] grants in [y].
TOO_MANY_RATE_CARDS	Only one rate card in ConfigurableUpfrontPricingTerm is allowed for the product.
TOO_MANY_RATE_CARDS	Up to [x] rate cards are allowed in ConfigurableUpfrontPricingTerm for the product.
TOO_MANY_RATES	Provide RateCards within the allowed limits in ConfigurableUpfrontPricingTerm.

Error code	Error message
TOO_MANY_RATES	Provide RateCards within the allowed limits in UsageBasedPricingTerm.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set both MultipleDimensionSelection and QuantityConfiguration to the same value (Allowed or Disallowed) in ConfigurableUpfrontPricingTerm.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Provide the same constraints for all rate cards in ConfigurableUpfrontPricingTerm.
INVALID_UPDATE_REQUEST	[x] can't be updated. To request pricing change or for more information, contact the AWS Marketplace Managed Catalog Operations Team.
INCOMPATIBLE_PRICING_MODEL	PricingModel can't change from [x] to [y].
INVALID_GRANTS	MaxQuantity for the FreeTrialPricingTerm is limited for the product. Provide a MaxQuantity less than or equal to [x]. For more information, contact the AWS Marketplace Managed Catalog Operations Team.
INVALID_GRANTS	Provide MaxQuantity for all Grants in FixedUpfrontPricingTerm.
INVALID_GRANTS	MaxQuantity isn't supported in FixedUpfrontPricingTerm for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure Grants in FixedUpfrontPricingTerm matches RateCards specified in the ResaleAuthorization.

Update the discoverability of the offer

You can use the Catalog API to control the discoverability of your offer in AWS Marketplace.

You can either choose to set a specific date in the future to restrict the discoverability of your offer or in the past to expire your offer. The `UpdateAvailability` change type doesn't affect existing agreements.

Note

- You can use the `UpdateAvailability` change type on a private offer that has already been [published](#) (also known as *released*). If buyers have already accepted the private offer, those existing agreements aren't affected.
- When modifying the `AvailabilityEndDate` of an existing private offer, the [constraints of the agreement duration](#) must be adhered to. If it's not, include an additional `UpdateValidityTerms` change type in this change set to modify the agreement duration to adhere to the new expiration. The `UpdateValidityTerms` change type can be used on a private offer that is either released or not yet released.
- When modifying the `AvailabilityEndDate` of an existing private offer, the [constraints of the payment schedule](#) must be adhered to. If it's not and the private offer is *not yet released*, include an additional `UpdatePaymentScheduleTerms` change type in this change set to modify the payment schedule to adhere to the new expiration. If the private offer is *already released*, you can only make changes to the `AvailabilityEndDate` as long as the new date adheres to the constraints of the payment schedule.

To control the discoverability of your offer, call the `StartChangeSet` API operation with the `UpdateAvailability` change type, as shown in the following example.

Request Syntax

```
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"
  }
}
```

Provide information for the fields to add the UpdateAvailability change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always Offer@1.0.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **AvailabilityEndDate** (string) (required) – This is the date until when the offer is discoverable and purchasable in AWS Marketplace. You can choose to set a specific date in the future to restrict the availability or in the past to expire the offer. Dates are represented in YYYY-MM-DD format.

A change set is created for your request. The response to this request gives you the ID and ARN for the change set and looks like the following.

Response Syntax

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

The change request is added to a queue and processed. It includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes. You can check the status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the DescribeChangeSet action.

Synchronous Validations

The following schema validations are specific to `UpdateAvailability` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
AvailabilityEndDate	Required Format: "YYYY-MM-DD"	422

Asynchronous Errors

The following errors are specific to `UpdateAvailability` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more details about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_AVAILABILITY_END_DATE	AvailabilityEndDate isn't supported for public offers.
INVALID_AVAILABILITY_END_DATE	Provide a future AvailabilityEndDate.
INVALID_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before AgreementEndDate.
MISSING_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before the agreement's end date.

Define the expiration date of agreements created using the offer

You can use the Catalog API to define the expiration date details of agreements created using the offer in AWS Marketplace.

This change type doesn't affect existing agreements.

Note

You can use the `UpdateValidityTerms` change type on a private offer that has already been [published](#) (also known as *released*). If buyers have already accepted the private offer, those existing agreements aren't affected.

For **AMI-based** and **container-based** products, if your private offer [pricing terms](#) include a term type that has a `Duration` (for example, the term types `FixedUpfrontPricingTerm` or `ConfigurableUpfrontPricingTerm`), your `AgreementDuration` set in this change type must be greater than the following: the number of days from today to the [expiration of the private offer](#) plus the number of days set in the `Duration` of those term types. This is because after a buyer accepts the private offer and the agreement is created, they can optionally purchase additional entitlements specified in those term types until the private offer expires. Furthermore, all additional entitlements must end before the agreement does. For example, if the buyer accepts the private offer on the first available day and then purchases entitlements on the last available day, those entitlements must not end after the agreement end date.

To define the expiration date details of agreements created using the offer, call the `StartChangeSet` API operation with the `UpdateValidityTerms` change type, as shown in the following example.

Request Syntax

```
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"
      }
    ]
  }
}

```

Provide information for the fields to add the `UpdateValidityTerms` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **Terms** (array of structures) – List of validity terms that you want to update. Supported validity terms are:
 - **ValidityTerm** (object) – Defines the conditions that will keep an agreement, created from this offer, valid.
 - **Type** (string) – Category of the term being updated. `ValidityTerm`
 - **AgreementDuration** (string) – Defines the duration that the agreement remains active. If `AgreementStartDate` isn't provided, agreement duration is relative to the agreement signature time. The duration is represented in the ISO_8601 format.
 - **AgreementStartDate** (string) – Defines the date when agreement starts. `AgreementStartDate` is represented in YYYY-MM-DD format. The agreement starts at 00:00:00.000 UTC on the date provided. If `AgreementStartDate` isn't provided, agreement start date is determined based on agreement signature time.
 - **AgreementEndDate** (string) – Defines the date when the agreement ends. The `AgreementEndDate` is represented in YYYY-MM-DD format. The agreement ends at 23:59:59.999 UTC on the date provided. If `AgreementEndDate` isn't provided, the agreement end date is determined by the validity of individual terms.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to UpdateValidityTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
Terms[].Type	Required Can only be "ValidityTerm"	422
Terms[].AgreementDuration	Optional Expected format per Selector type: ISO 8601 duration Can be stand alone or paired with AgreementStartDate	422
Terms[].AgreementEndDate	Optional Date must be formatted like "YYYY-MM-DD"	422
Terms[].AgreementStartDate	Optional	422

Input field	Validation rule	HTTP code
	Date must be formatted like "YYYY-MM-DD"	
	Can only be paired with Agreement EndDate and Agreement Duration	

Asynchronous Errors

The following errors are specific to UpdateValidityTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_AGREEMENT	AgreementStartDate can't be in the future when the current agreement to be replaced isn't future dated.
INCOMPATIBLE_AGREEMENT_END_DATE	AgreementEndDate can't be updated after the offer is released.
INCOMPATIBLE_AGREEMENT_START_DATE	AgreementStartDate can't be updated after the offer is released.
INCOMPATIBLE_PRODUCT	AgreementStartDate in the future isn't supported.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between Agreement StartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementStartDate is compatible with the ResaleAuthorization.

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between AgreementStartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementDuration matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_TERMS	ValidityTerm isn't supported for public offers.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide AgreementDuration that is greater than or equal to [x] days.
INVALID_AGREEMENT_END_DATE	Provide a future AgreementEndDate.
INVALID_AGREEMENT_END_DATE	Provide AgreementEndDate that is after or equal to [x].
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is after AvailabilityEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is before the AgreementEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is within [x] years from today.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementDuration and AgreementEndDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementStartDate and AgreementDuration isn't supported in an offer for the product.

Error code	Error message
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with AgreementStartDate isn't supported in an offer for the product.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with only AgreementStartDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementEndDate isn't supported unless it's used in combination with a future Agreement StartDate or for replacement offers.
INVALID_AGREEMENT_TIME_INTERVAL	Provide AgreementStartDate and Agreement EndDate where the difference is less than or equal to [x] years.
MISSING_AGREEMENT_START_DATE	Ensure AgreementStartDate is present in ValidityTerm when used along with ConfigurableUpfrontPricingTerm.
INVALID_AGREEMENT_END_DATE	Provide an AgreementEndDate that is within [x] years from today.
INCOMPATIBLE_AGREEMENT_START_DATE	Provide the same AgreementStartDate as defined in the agreement when the agreement has a future start date.
INCOMPATIBLE_AGREEMENT	AgreementStartDate can't be future dated when the agreement isn't future dated.

Update payment schedule details

You can use the Catalog API to update payment schedule details for your offer, such as flexible payment schedule, in AWS Marketplace.

Note

You cannot use the `UpdatePaymentScheduleTerms` change type on an offer that has already been [published](#) (also known as *released*).

The private offer can be accepted any day between the creation of the private offer and its [expiration](#) (set in the `AvailabilityEndDate`). Only one `ChargeDate` value of the payment schedule can be a date on or before the last day the buyer can accept the private offer (the private offer expiration date). The remaining values of `ChargeDate` must be after the private offer expiration, but no later than the end of the agreement if the private offer was accepted immediately. The end of the agreement is based on when the private offer is accepted (creating the agreement) plus the [duration of the agreement](#).

To update payment schedule details for your offer, call the `StartChangeSet` API operation with the `UpdatePaymentScheduleTerms` change type, as shown in the following example.

Request Syntax

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

```
}  
}  
]  
}
```

Provide information for the fields to add the `UpdatePaymentScheduleTerms` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **Terms** (array of structures) – List of payment terms that you want to update. Supported payment terms are:
 - **PaymentScheduleTerm** (object) – Defines an installment-based pricing model where customers are charged a fixed price on different dates during the agreement validity period.
 - **Type** (string) – Type of the term being updated. This is the object value: `"PaymentScheduleTerm"`.
 - **Schedule** (array of structures) – List of the payment schedule where each element defines one installment of payment. It contains the information necessary for calculating the price to be paid and the date on which the customer would be charged.
 - **ChargeDate** (string) – The date on which the customer would pay the price defined in this payment schedule term. `ChargeDate` is represented in YYYY-MM-DD format. Invoices are generated on the date provided.
 - **ChargeAmount** (string) – The price that the customer would pay on scheduled date (`ChargeDate`).

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdatePaymentScheduleTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input Field	Validation Rule	HTTP
Terms	Required Only <code>PaymentScheduleTerm</code> is allowed List size must be less than 2	422
Terms[].Type	Required Can only be <code>PaymentScheduleTerm</code>	422
Terms[].PaymentScheduleTerm.CurrencyCode	Required Supported currencies: ["USD", "AUD", "EUR", "GBP", "JPY"]	422
Terms[].PaymentScheduleTerm.Schedule[]	Required	422
Terms[].PaymentScheduleTerm.Schedule[].ChargeAmount	Required Date type is "String" Non-negative decimals with up to 2 decimal places supported	422

Input Field	Validation Rule	HTTP
Terms[].PaymentScheduleTerm .Schedule[].ChargeDate	Required Date must be formatted like "YYYY-MM-DD"	422

Asynchronous Errors

The following errors are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
DUPLICATE_CHARGE_DATES	Provide unique charge dates in PaymentScheduleTerm.
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.
INCOMPATIBLE_MARKUP_PERCENTAGE	PaymentScheduleTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the CurrencyCode.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the total ChargeAmounts in PaymentScheduleTerm is compatible with the ResaleAuthorization.
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.

Error code	Error message
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_CHARGE_DATES	Provide charge dates before Agreement EndDate.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
TOO_MANY_BACKDATED_CHARGES	Provide up to 1 scheduled payment before AvailabilityEndDate.
INVALID_CHARGE_DATES	Provide a last charge date that is before AgreementEndDate.
INVALID_CHARGE_DATES	Provide a first charge date that isn't in the past.
TOO_MANY_CHARGES	Provide up to [x] scheduled payments in PaymentScheduleTerm.

Modify renewal options

You can use the Catalog API to control renewal options of the agreements that are created using this offer in AWS Marketplace.

For offers created through Catalog API, auto-renewal remains disabled by default until you call the `UpdateRenewalTerms` change type to allow auto-renewal. This change does not affect existing agreements.

To control renewal options of the agreements that are created using this offer, call the `StartChangeSet` API operation with the `UpdateRenewalTerms` change type, as shown in the following example.

Request Syntax

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

Provide information for the fields to add the `UpdateRenewalTerms` change type:

- **Entity** (object) (required) – Your offer.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **Terms** (array of structures) – List of renewal terms that you want to update. Supported renewal terms are:
 - **RenewalTerm** (object) – Defines that on graceful termination (expiration of the `ValidityTerm`, not buyer or AWS Marketplace cancellation) of the agreement, a new agreement will be created using the accepted terms on the existing agreement. In other

words, the agreement will be renewed. Presence of `RenewalTerm` in the offer means that auto-renewal is allowed. Buyers will have the option to accept or decline auto-renewal at the offer acceptance/agreement creation.

- **Type** (string) – Type of the term being updated. `RenewalTerm`

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdateRenewalTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
Terms[].Type	Required Can only be "RenewalTerm"	422

Asynchronous Errors

The following errors are specific to `UpdateRenewalTerms` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more details about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT	RenewalTerm isn't supported in private offers for the product.
INCOMPATIBLE_TERMS	RenewalTerm isn't supported together with PaymentScheduleTerm.
INCOMPATIBLE_TERMS	RenewalTerm isn't supported with the PricingModel.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.

Publish an offer

You can use the Catalog API to merge the information collected from all update change types, and then publish the offer.

Offers remain in a `Draft` state, until `ReleaseOffer` is called. After the offer is released, it's discoverable in AWS Marketplace.

To publish your offer, call the `StartChangeSet` API operation with the `ReleaseOffer` change type, as shown in the following example.

Request Syntax

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

{
```



```
"Catalog": "AWSMarketplace",
"ChangeSet": [
  {
    "ChangeType": "ReleaseOffer",
    "Entity": {
      "Type": "Offer@1.0",
      "Identifier": "offer-123456789"
    },
    "DetailsDocument": {}
  }
]
```

Provide information for the fields to add the `ReleaseOffer` change type:

- **Entity** (object) – The named type of entity being created. The `Identifier` is your offer ID, and the `Type` is always `Offer@1.0`. For more information, see [Identifier](#).
- **DetailsDocument** (object) – The JSON value of specifics of the request. It must be empty for `ReleaseOffer`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `ReleaseOffer` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
DetailsDocument	Must be empty ({})

Asynchronous Errors

The following errors are specific to `ReleaseOffer` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the <code>CurrencyCode</code> .
INCOMPATIBLE_PRODUCT	First create a public offer for the product.
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.
INCOMPATIBLE_TARGETING	<code>PreExistingAgreement</code> is only supported for buyer targeted offers.
INCOMPATIBLE_TARGETING	<code>OfferSetId</code> is only supported for buyer targeted offers.
INVALID_TAX_INFORMATION	Your tax information is incomplete. To sell professional services on AWS Marketplace, you must complete the DAC7 tax questionnaire. Navigate to the Payment Information section, and select the DAC7 tax form. It can take up to two hours for your tax information to update.

Error code	Error message
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer is released.
MISSING_AGREEMENT_END_DATE	Provide an AgreementEndDate for replacement offers.
MISSING_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate for private offer.
MISSING_BUYER_ACCOUNTS	Provide PositiveTargeting with BuyersAccounts for offers created using ResaleAuthorization.
MISSING_BUYER_ACCOUNTS	All offers for the product must be private. Provide PositiveTargeting with BuyersAccounts.
MISSING_DESCRIPTION	Set Description before releasing the offer.
MISSING_MANDATORY_TERMS	Add [x] to the offer.
MISSING_MANDATORY_TERMS	Provide a FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
MISSING_NAME	Set Name before releasing the offer.
TOO_MANY_OFFERS	Only one public free trial offer can be created per product.
TOO_MANY_OFFERS	Only one public offer can be created per product.
MISSING_MANDATORY_TERMS	Provide a RenewalTerm for public offers with contract pricing for the product.
MISSING_AGREEMENT_END_DATE	Provide an AgreementEndDate for replacement offers.

Describe existing offer details

You can use the Catalog API to describe existing offer details in AWS Marketplace.

To describe existing offer details, call the `DescribeEntity` API operation with the `Offer@1.0` entity type, as shown in the following example.

Request Syntax

```
GET /DescribeEntity?catalog=<Catalog>&entityId=<EntityId> HTTP/1.1
```

Provide information for the fields to add the `DescribeEntity` change type:

- **catalog** (string) – The catalog related to the request. Fixed value: `AWSMarketplace`.
- **entityId** (string) – The unique ID of the offer to describe.

Response Syntax

The response to this request gives you the offer details and looks like the following.

```
{
  "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"
    }
  ]
}

```

The following is information about the fields you see in the `DescribeEntity` response.

- **EntityType** (string) – The named type of the entity, which is `Offer@1.0`.
- **EntityIdentifier** (string) – The identifier of the entity, in the format of `EntityId@RevisionId`.
- **EntityArn** (string) – The ARN associated to the unique identifier for the change set referenced in this request.
- **LastModifiedDate** (string) – The last modified date of the entity, in ISO 8601 format (for example: `2018-02-27T13:45:22Z`).
- **Details** (string) – This stringified JSON object includes the following details of the entity:
 - **Id** (string) – Unique identifier for an offer entity in AWS Marketplace and is generated during the creation of an offer.
 - **State** (string) – The status of the offer.
 - **Name** (string) – The name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.
 - **Description** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
 - **PreExistingAgreement** (string) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
 - **AcquisitionChannel** (string) – Indicates if the existing agreement was signed outside AWS Marketplace or within AWS Marketplace. Possible values: `External`, `AwsMarketplace`.
 - **PricingModel** (string) – Indicates which pricing model the existing agreement uses. Possible values: `Contract`, `Usage`, `ByoI`, `Free`.
 - **ProductId** (string) – The unique identifier of the product being offered.
 - **OfferSetId** (string) – The unique identifier of the offer set to associate this offer with.
 - **Terms** (array of structures) – List of terms.
 - **Rules** (array of structures) – List of rules.

Work with resale authorizations using the AWS Marketplace APIs

You can use the AWS Marketplace Catalog API to automate tasks for working with Resale Authorizations.

While the *product* describes what is being sold in AWS Marketplace, the *Resale Authorization* (also known as an opportunity) describes the terms and rules regarding how this product is authorized to be resold in AWS Marketplace. The *CPPO* is the target of the Resale Authorization.

A Resale Authorization has a collection of terms and rules to be accepted for a reseller agreement between manufacturers and channel partners. Accepting the terms of the Resale Authorization allows the reseller to create offers for the product per the conditions expressed in the terms.

There are two types of rules in a Resale Authorization:

- **AvailabilityRule** – Controls the lifecycle of the Resale Authorization in AWS Marketplace.
- **PartnerTargetingRule** – Specifies whether the Resale Authorization should be accessible to a specific set of channel partners.

See the following resources:

- For end-to-end labs with working code examples, see [Lab: Authorize a reseller](#) in the *AWS Marketplace seller workshop*.
- For code examples of API requests, see [Python](#) and [Java](#) examples in *AWS Samples* on GitHub.
- For a video on creating resale authorizations, see [Create Resale Authorizations Using the AWS Marketplace Catalog API](#) on YouTube.

The following topics describe how to use the Catalog API to create and update Resale Authorizations:

Topics

- [Resale Authorization prerequisites](#)
- [Create a new Resale Authorization](#)
- [Update buyer targeting](#)
- [Update availability](#)
- [Update the validity of a future dated agreement](#)
- [Update legal resources](#)

- [Update pricing](#)
- [Update payment schedule](#)
- [Update Resale Authorization details](#)
- [Restrict a Resale Authorization](#)
- [Release a Resale Authorization and make it visible to a Channel Partner](#)
- [Describe an existing Resale Authorization](#)

Resale Authorization prerequisites

To use Resale Authorization, both independent software vendors (ISVs) and AWS Marketplace Channel Partners must create a service-linked role that provides resource-sharing permissions to AWS. If both groups don't perform this prerequisite, AWS can't share the authorization resource from the ISV to the AWS Marketplace Channel Partner. For more information, see [Using roles for Resale Authorization for AWS Marketplace](#) in the *AWS Marketplace Seller Guide*.

Create a new Resale Authorization

You can use the Catalog API to create a new Resale Authorization in AWS Marketplace.

If your request is processed successfully, AWS Marketplace Catalog API generates a Resale Authorization in Draft state for you. It's an incomplete Resale Authorization and not visible to channel partners in AWS Marketplace.

Use the Update change types to complete the Resale Authorization. After the Resale Authorization is completed, use the `ReleaseResaleAuthorization` change type to complete the Resale Authorization creation process and release the Resale Authorization, which will validate the entire Resale Authorization and make your it visible to channel partners in AWS Marketplace.

To create a Resale Authorization in Draft state, call the `StartChangeSet` API operation with the `CreateResaleAuthorization` change type, as shown in the following example.

Request Syntax

```
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"
    }
  }
]
```

Provide information for the input fields to add the `CreateResaleAuthorization` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
- **DetailsDocument** (object) (required) – Specifics of the request.
 - **ProductId** (string) (required) – Product ID for which to create the resale authorization.
 - **Name** (string) (required) – Name associated with the `ResaleAuthorization` for better readability to you and your channel partners.
 - **Description** (string) (optional) – A free-form text field available to add details about the `ResaleAuthorization`.
 - **ResellerAccountId** (string) (required) – Add targeted channel partner's AWS account who can describe and use this `ResaleAuthorization` to create a private offer.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

```
{
  "ChangeSetId": "example123456789012abcdef",
```

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

The change request is added to a queue and processed.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

When the request is complete (if the Status is SUCCEEDED), a new ResaleAuthorization is generated. Although the SUCCEEDED status indicates that the CreateResaleAuthorization change type call is completed, the ResaleAuthorization status is still in Draft state.

The following shows the response from the [DescribeChangeSet](#) API operation.

```
{
  "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":
      []
    }
  ]
}
```

```
]
}
```

Synchronous Validations

The schema validations are specific to `CreateResaleAuthorization` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
ProductId	Required	422
	Must not be null or empty	
	Length must be between 1 and 50 characters	
ProductId	User must be authorized to create ResaleAuthorization for the given product	403
ProductId	Must be an existing product in the catalog and not in Draft state	404
	Product should be supported to resell	
Name	Required	422
	Must not be null or empty	
	Length must be between 1 and 100 characters	
	No special characters allowed	
Description	Optional	422
	Length must be between 1 and 255 characters	

Input field	Validation rule	HTTP code
	No special characters allowed	
ResellerAccountId	Required Must not be empty AWS account IDs must be in valid format (12-digit number)	422
BulkRequestId	Optional Length must be between 1 and 50 characters Must be in UUID format	422
An unknown property	No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to `CreateResaleAuthorization` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_RESELLER_ACCOUNT	Provide a valid reseller account.

Update buyer targeting

You can use the Catalog API to update buyers targeting your Resale Authorization in AWS Marketplace.

Any existing targeting options that aren't included in the latest request are removed from the Resale Authorization. This change type is optional for release of the Resale Authorization.

To update buyers targeting your Resale Authorization, call the `StartChangeSet` API operation with the `UpdateBuyerTargetingTerms` change type, as shown in the following example.

Request Syntax

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

Provide information for the fields to add the `UpdateBuyerTargetingTerms` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.

- **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request.
 - **Terms** (array of structures) (optional) – List of buyers targeting terms that you want to update. If the intentions aren't to target the `ResaleAuthorization` to any specific buyer, then terms field can be skipped. By default, `ResaleAuthorization` is targeted to all buyers. Supported terms are:
 - **BuyerTargetingTerms** (object) (optional) – Define buyer-specific targeting to your `ResaleAuthorization`.
 - **Type** (string) (required) – Category of the term being updated.
 - **PositiveTargeting** (object) (required) – Defines the criteria that any buyer's profile should fulfill to be allowed access to the `ResaleAuthorization`.
 - **BuyerAccounts** (array of strings) (optional) – List as optional. You can add the targeted buyer's AWS accounts. If the intention isn't to target `ResaleAuthorization` to specific buyers, then this field should be omitted. By default, all buyers are targeted. Targeted channel partners can choose to create a private offer and target a subset of buyers, if specified.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The schema validations are specific to `UpdateBuyerTargetingTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Terms	Optional Must not be null or empty Only "BuyerTargetingTerm" is allowed in the list List size must be 1 (there is no use case today that requires multiple buyer terms)
BuyerTargetingTerm.PositiveTargeting	Required Must not be empty
BuyerTargetingTerm.PositiveTargeting.BuyerAccounts	Optional AWS account IDs must be in valid format (12-digit number) Must not contain more than 25 accounts
An unknown property	No additional properties are allowed

Asynchronous Errors

The following errors are specific to `UpdateBuyerTargetingTerms` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_BUYER_TARGETING	At least one Buyer account must be present for ResaleAuthorization with PreExistingBuyerAgreement.

Update availability

You can use the Catalog API to limit the availability of how many private offers are created or until what specific time a private offer can be created.

By default, the value is unlimited usage of this Resale Authorization, although you can check the availability under the rule list.

To control the availability and usability of your Resale Authorization, call the `StartChangeSet` API operation with the `UpdateAvailability` change type, as shown in the following example.

Request Syntax

```
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
      }
    }
  ]
}
```

```
}
```

Provide information for the fields to add the `UpdateAvailability` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
 - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request.
 - **AvailabilityEndDate** (string) (optional) – Define the end date until Channel Partners can leverage the `ResaleAuthorization` to create an offer. Channel Partners can use this `ResaleAuthorization` multiple times until the specified end date. Dates are represented in ISO_8601 format.
 - **OffersMaxQuantity** (integer) (optional) – Define the maximum number of private offers that can be created using the `ResaleAuthorization`. This doesn't define the number of subscriptions.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The schema validations are specific to UpdateAvailability actions in the AWS Marketplace Catalog API. The validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response

Input field	Validation rule
OffersMaxQuantity	Optional Must be non-negative integer Allowed value only "1" (Currently no use case to support multiple quantity)
AvailabilityEndDate	Optional Must be ISO_8601 formatted Must be date in the future
Availability	Provide either OffersMaxQuantity or AvailabilityEndDate.
An unknown property	No additional properties are allowed

Asynchronous Errors

The following errors are specific to UpdateAvailability actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before all the ChargeDate in ResalePaymentScheduleTerms.
INVALID_AVAILABILITY_END_DATE	Provide a future AvailabilityEndDate.

Update the validity of a future dated agreement

You can use the Catalog API to modify and control a future dated service start date in AWS Marketplace.

This change set is not mandatory to release a Resale Authorization.

To modify and control the product agreement duration of your Resale Authorization, call the `StartChangeSet` API operation with the `UpdateBuyerValidityTerms` change type, as shown in the following example.

Note

Future-dated agreements are only supported for SaaS product types.

Request Syntax

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

```
]
}
```

Provide information for the input fields to add the `UpdateBuyerValidityTerms` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
 - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request.
 - **Terms** (array of structures) – List of agreement validity terms that you want to update. Supported terms are:
 - **BuyerValidityTerm** (object) – Defines the availabilities of a service for a product in your ResaleAuthorization.
 - **Type** (string) – Category of term being updated.
 - **MaximumAgreementStartDate** (string) (required) – Define the agreement start date for the product offered. Future dated offers can't exceed this service start date. Dates are represented in ISO_8601 format.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The schema validations are specific to `UpdateBuyerValidityTerms` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Terms	Required Must not be null or empty Only "BuyerValidityTerm" is allowed in the list List size must be 1 (there's no use case today that requires multiple service availability terms)
MaximumAgreementStartDate	Required Must not be null or empty Must be future date and shouldn't exceed more than 3 years from now Must be ISO_8601 formatted
An unknown property	No additional properties are allowed

Asynchronous Errors

The following errors are specific to `UpdateBuyerValidityTerms` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PRODUCT	BuyerValidityTerm isn't supported for the product.

Error code	Error message
INVALID_MAXIMUM_AGREEMENT_START_DATE	Provide a future MaximumAgreementStartDate with in allowed limit.

Update legal resources

You can use the Catalog API to replace the existing legal terms completely in AWS Marketplace.

The legal terms that aren't included in the latest request will be removed from the Resale Authorization. `BuyerLegalTerm` contains the EULA which will be included on the final buyer agreement and `LegalTerm` includes the Reseller Contract which will be included in the reseller agreement between the channel partner and the ISV.

To update legal terms of your `ResaleAuthorization`, call the `StartChangeSet` API operation with the `UpdateLegalTerms` change type, as shown in the following example.

Request Syntax

```
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"
            }
        ]
    }
]
}

```

Provide information for the fields to add the `UpdateLegalTerms` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
 - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request.
 - **Terms** (array of structures) (required) – List of legal terms. Supported legal terms are:
 - **BuyerLegalTerm** (object) (required) – Defines the list of text agreements to be proposed to acceptors. For example, the end user license agreement (EULA).
 - **Type** (string) (required) – Category of the term being updated.
 - **Documents** (array of structures) (required) – List of references to legal resources to be proposed to the buyers. For example, the EULA. Each reference is made up of a Type and a URL:
 - **Type** (string) (required) – Type of document. Available document types are:

- **StandardEula** – Standard Contract for AWS Marketplace. For more information, see [SCMP](#) in the *AWS Marketplace Seller Guide*. You don't need to provide a URL for this type because it's managed by AWS Marketplace.
- **EnterpriseEula** – Enterprise Contract for AWS Marketplace. For more information, see DSA in the *AWS Marketplace Seller Guide*. You don't need to provide a URL for this type because it's managed by AWS Marketplace.
- **CustomEula** – Custom EULA provided by you as a manufacturer. A URL for the EULA stored in an accessible S3 bucket is required for this document type.
- **Url** (string) (conditionally required) – A URL to the legal document for buyers to read. This is required when category Type is CustomEula.
- **ResaleLegalTerm** (object) (optional) – Defines the list of text agreements to propose only to channel partners. This term won't be available to buyers.
- **Type** (string) (required) – Category of term being updated.
- **Documents** (array of structures) (required) – List of references to the reseller legal resources to be proposed to the channel partners.
 - **Type** (string) (required) – Category of the document. Available document types are:
 - **StandardResellerContract** – Standard Reseller Contract for AWS Marketplace.
 - **CustomResellerContract** – A custom reseller contract by you as a manufacturer. A URL for the reseller contract is stored in an accessible S3 bucket and is required for this document type.
 - **Url** (string) (conditionally required) – URL to the reseller contract document for channel partners to read. It's required when the Type is CustomResellerContract.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The schema validations are specific to `UpdateLegalTerms` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required Must not be null or empty	422
Terms[].BuyerLegalTerm	Required Must not be null or empty	422
Terms[].ResaleLegalTerm	Optional Must not be null or empty if present	422
Terms[].BuyerLegalTerm.Documents	Required Must not be null or empty	422
Terms[].BuyerLegalTerm.Documents[].Type	Required Must not be null or empty Allowed values: - StandardEula - EnterpriseEula - CustomEula	422

Input field	Validation rule	HTTP code
Terms[].BuyerLegalTerm.Documents[].Url	Required and must be a valid URL when "Type" is "CustomEula" Must not be provided when "Type" is one of ["StandardEula", "EnterpriseEula"]	422
Terms[].ResaleLegalTerm.Documents	Required Must not be null or empty	422
Terms[].ResaleLegalTerm.Documents[].Type	Required Must not be null or empty Allowed values: • StandardEula • CustomResellerContract	422
Terms[].ResaleLegalTerm.Documents[].Url	Required and must be a valid URL when "Type" is "CustomResellerContract" Must not be provided when "Type" is one of ["StandardContract"]	422
An unknown property	No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to UpdateLegalTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet, after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_BUYER_LEGAL_DOCUMENTS	Provide URLs for buyer legal documents stored in accessible S3 buckets.
INVALID_RESALE_LEGAL_DOCUMENTS	Provide URLs for resale legal documents stored in accessible S3 buckets.
MISSING_MANDATORY_TERMS	Provide a BuyerLegalTerm.

Update pricing

You can use the Catalog API to replace the existing pricing terms completely in AWS Marketplace.

Pricing terms that aren't included in the latest request will be removed from the Resale Authorization. You can update the discounted pricing for your product through this API.

To update pricing details for your Resale Authorizations, call the `StartChangeSet` API operation with the `UpdatePricingTerms` change type, as shown in the following example.

Request Syntax

```
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
        }
    ]
}
]
}
}
]
}
}

```

Provide information for the fields to add the UpdatePricingTerms change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always ResaleAuthorization@1.0.
 - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request.
 - **PricingModel** (string) (required) – Pricing model for your offer. Possible values for pricing model are:
 - **Usage** – Usage-based pricing model where buyers will be billed for their usage of your product.
 - **Contract** – In the contract-based pricing model, buyers are either billed in advance for the use of your product or offered a flexible payment schedule. Buyers can also pay for additional usage above their contract. Channel partners can add their markup to this payment schedule and pricing for each dimension.

- **Terms** (array of structures) (required) – List of pricing terms that you want to update.
Supported pricing terms are:
 - **ResaleUsageBasedPricingTerm** (object) – Defines a pay-as-you-go (PAYG) pricing model where the customers are charged based on product usage.
 - **Type** (string) (required) – Category of the term.
 - **CurrencyCode** (string) – Defines the currency for prices mentioned in this term. Currently, only USD is supported.
 - **RateCards** (array of structures) – List of rate cards.
 - **RateCard** (array of structures) – A rate card defines the per-unit rates for the product dimensions.
 - **DimensionKey** (string) – Dimension for which the given entitlement applies. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
 - **Price** (string) – Per unit price for the product dimension which is used for calculating the amount to be charged.
 - **Constraints** (object) (optional) – Defines limits on how the term can be configured by acceptors.
 - **MultipleDimensionSelection** (string) (optional) – Determines if buyers are allowed to select multiple dimensions in the rate card. Possible values are Allowed and Disallowed. Default value is Allowed.
 - **QuantityConfiguration** (string) (optional) – Determines if acceptors are allowed to configure quantity for each dimension in rate card. Possible values are Allowed and Disallowed. Default value is Allowed.
 - **ResaleFixedUpfrontPricingTerm** (object) – Defines a pre-paid pricing model where the customers are charged a fixed upfront amount.
 - **Type** (string) (required) – Category of the term being updated.
 - **CurrencyCode** (string) – Defines the currency for prices mentioned in this term. Defines the currency for the prices mentioned in this term. USD, AUD, EUR, GBP, and JPY are supported.
 - **Price** (string) (required) – Fixed amount to be charged to the customer when this term is accepted.
 - **Duration** (string) (required) – Contract duration of the ResaleAuthorization. This field supports the ISO 8601 format.

- **Grants** (array of structures) (required) – Entitlements that will be granted to the acceptor of fixed upfront pricing as part of agreement execution.
- **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
- **MaxQuantity** (integer) (required) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If MaxQuantity is not provided, the buyer will be able to use an unlimited amount of the given dimension.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdatePricingTerms` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Terms	Required
	Must not be null or empty

Input field	Validation rule
	Each term must present only single time Allowed terms: • * ResaleUsageBasedPricingTerm • * ResaleConfigurableUpfrontPricingTerm • * ResaleFixedupfrontPricingTerm
Terms[].ResaleUsageBasedPricingTerm.CurrencyCode	Required Allowed values: USD
Terms[].ResaleUsageBasedPricingTerm.Validity	Required Must not be null or empty Expected format: ISO 8601 duration
Terms[].ResaleUsageBasedPricingTerm.RateCards	Required Must not be null or empty
Terms[].ResaleUsageBasedPricingTerm.RateCards[].DimensionKey	Required Must not be null or empty Length must be between 1 and 60
Terms[].ResaleUsageBasedPricingTerm.RateCards[].Price	Required Must not be null or empty Data type is "String" Must be non-negative Support up to 8 Decimal No special characters supported

Input field	Validation rule
Terms[].ResaleConfigurableUpfrontPricingTerm.CurrencyCode	Required Allowed values: ["USD", "AUD", "EUR", "GBP", "JPN"]
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Selector.Type	Required Must not be null or empty Allowed values: Duration
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Selector.Value	Required Must not be null or empty Expected format: ISO 8601 duration
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].RateCard.DimensionKey	Required Must not be null or empty Length must be between 1 and 60
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].RateCard.Price	Required Must not be null or empty Data type is "String" Must be non-negative Support up to 6 Decimal No special characters supported
Terms[].ResaleConfigurableUpfrontPricingTerm.RateCards[].Constraints	Optional

Input field	Validation rule
Terms[].ResaleFixedUpfrontPricingTerm.CurrencyCode	Required Allowed values: ["USD", "AUD", "EUR", "GBP", "JPN"]
Terms[].ResaleFixedUpfrontPricingTerm.Price	Required Must not be null or empty Data type is "String" Must be non-negative Support up to 6 Decimal No special characters supported Allowed values: 0.0
Terms[].ResaleFixedUpfrontPricingTerm.Duration	Required Must not be null or empty Expected format: ISO 8601 duration
Terms[].ResaleFixedUpfrontPricingTerm.Grants[].DimensionKey	Required Must not be null or empty Length must be between 1 and 60
Terms[].ResaleFixedUpfrontPricingTerm.Grants[].MaxQuantity	Required Must not be null or empty
An unknown property	No additional properties are allowed

Asynchronous Errors

The following errors are specific to `UpdatePricingTerms` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_CURRENCY_CODE	Provide the same <code>CurrencyCode</code> across all pricing and payment terms.
INCOMPATIBLE_PRODUCT	Use existing, available dimensions in the product in [x].
DUPLICATE_DIMENSION_KEYS	Provide rate card with a unique list of dimension keys in [x]
INVALID_RATE_CARD	Provide dimensions that have the same unit in [x]
INVALID_RATE_CARD	Provide a rate card for only metered dimensions in <code>ResaleUsageBasedPricingTerm</code> .
INVALID_RATE_CARD	Provide usage based rates for all available metered dimensions in <code>ResaleUsageBasedPricingTerm</code> .
TOO_MANY_RATES	Provide <code>RateCards</code> within the allowed limits in <code>ResaleUsageBasedPricingTerm</code> .
DUPLICATE_SELECTORS	Provide a unique list of <code>Selectors</code> in <code>ResaleConfigurableUpfrontPricingTerm</code> .
INVALID_RATE_CARD	<code>ConfigurableUpfrontPricingTerm</code> is missing one or more dimension keys for duration [x]. Provide prices for the same set of dimension keys for all durations.
INVALID_RATE_CARD	Provide either all metered or all entitled dimensions in [x].

Error code	Error message
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Disallowed in ResaleConfigurableUpfrontPricingTerm for the PricingModel.
TOO_MANY_RATE_CARDS	Only one rate card in ConfigurableUpfrontPricingTerm is allowed for the product.
INCOMPATIBLE_TERMS	The following terms aren't compatible with the PricingModel: [x,y,z].
TOO_MANY_RATES	Provide RateCards within the allowed limits in [x term].
TOO_MANY_GRANTS	Provide up to [N] grants in [x term].
INVALID_SELECTOR_DURATION_VALUE	Provide duration between [x] and [y] months in ResaleConfigurableUpfront
TOO_MANY_GRANTS	Provide duration between [x] and [y] months.
INVALID_SELECTOR_DURATION_VALUE	Ensure duration granularity is at the day level for metered dimensions in ResaleConfigurable Upfront
INVALID_DURATION	Ensure duration granularity is at the day level for metered dimensions in FixedUpfront.
INVALID_RATE_CARD	Provide only entitled dimensions in [x].
MISSING_DURATION	Provide a Duration in [x].
DUPLICATE_DIMENSION_KEYS	Provide Grants with a unique list of dimension keys in [x].
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the CurrencyCode.

Error code	Error message
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.

Update payment schedule

You can use the Catalog API to change payment-associated details, such as a flexible payment schedule, in AWS Marketplace.

To update payment-associated details for your Resale Authorization, call the `StartChangeSet` API operation with the `UpdatePaymentScheduleTerms` change type, as shown in the following example.

Request Syntax

```
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"
        }
      ]
  }
]
```

Provide information for the fields to add the UpdatePaymentScheduleTerms change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always ResaleAuthorization@1.0.
 - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request.
 - **Terms** (array of structures) – List of payment terms that you want to update. Supported payment terms are:
 - **ResalePaymentScheduleTerm** (object) – Defines an installment-based pricing model where the customers are charged a fixed price on different dates during the agreement validity period.
 - **Type** (string) – Category of the term being updated.
 - **CurrencyCode** (string) (required) – Defines the currency for the payment mentioned in the schedule. USD, AUD, EUR, GBP, and JPY are supported.

- **Schedule** (array of structures) – List of the payment schedule where each element defines one installment of payment. It contains the information necessary for calculating the price to be paid and the date on which the customer would be charged.
- **ChargeDate** (string) (required) – The date the customer would pay the price defined in this payment schedule term. This field supports the ISO 8601 format.
- **ChargeAmount** (string) (required) – The price the customer would pay on a scheduled date (ChargeDate).

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The schema validations are specific to `UpdatePaymentScheduleTerms` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
Terms.Type	Required	422
	Not supported for [x] product	

Input field	Validation rule	HTTP
	Allowed terms: ResalePaymentScheduleTerm	
Terms[].CurrencyCode	Required Allowed values: USD	422
Terms[].ResalePaymentScheduleTerm.Schedule	Required Length must be between 1 and 60	422
Terms[].ResalePaymentScheduleTerm.Shedule.ChargeDate	Required Must be in ISO 8601 format Date must be in the future	422
Terms[].ResalePaymentScheduleTerm.Shedule.ChargeAmount	Required Must be non-negative	422
An unknown property	No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. or more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_TERMS	OffersMaxQuantity and AvailabilityEndDate must be present with ResalePaymentScheduleTerm.
TOO_MANY_SCHEDULED_PAYMENTS	Provide up to 60 scheduled payments in ResalePaymentScheduleTerm.

Error code	Error message
DUPLICATE_CHARGE_DATES	Provide unique charge dates in ResalePaymentScheduleTerm.
INVALID_CHARGE_DATES	Provide a future ChargeDate.
INVALID_CHARGE_DATES	Provide a last charge date that is before [x].
MISSING_MANDATORY_TERMS	Provide a ResaleFixedUpfrontPricingTerm and ResalePaymentScheduleTerm together.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the CurrencyCode.
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.

Update Resale Authorization details

You can use the Catalog API to update Resale Authorization details in AWS Marketplace.

To update Resale Authorization details, call the `StartChangeSet` API operation with the `UpdateInformation` change type, as shown in the following example.

Note

The `UpdateInformation` change type only updates the sections provided in the request; all other information remains unchanged.

Request Syntax

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

Provide information for the fields to add the `UpdateInformation` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.

- **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Details of the request, including the information you want to update information for the Resale Authorization.
- **Name** (string) (optional) – The name associated with the ResaleAuthorization for better readability to you and your channel partners.
- **Description** (string) (optional) – The description is free-form text where you can add details about the ResaleAuthorization.
- **PreExistingBuyerAgreement** (object) (optional) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
 - **AcquisitionChannel** (string) (required) – Indicates if the existing buyer agreement was signed outside AWS Marketplace or in AWS Marketplace.

Possible values: External, AwsMarketplace

- **PricingModel** (string) (required) – Indicates which pricing model the exiting agreement uses.

Possible values: Contract, Usage, BYOL, Free

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet, and the request will fail with an HTTP error if the input does not meet the following requirements.

Input field	Validation rule	HTTP code
Name	Optional	422
	Must not be null or empty	
	Length must be between 1 and 100 characters	
	Pattern <code>^[A-Za-z0-9]*\$</code>	
Description	No special character or white space allowed	422
	Optional	
	Length must be between 1 and 255 characters	
	Pattern <code>^[A-Za-z0-9\\s]*\$</code>	
An unknown property	No special characters allowed	422
	No additional properties are allowed	

Asynchronous Errors

The following errors are specific to UpdateInformation actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing, or more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_BUYER_TARGETING	At least one Buyer account must be present for ResaleAuthorization with PreExistingBuyerAgreement.

Restrict a Resale Authorization

You can use the Catalog API to set restrict rules to a Resale Authorization in AWS Marketplace.

A restricted Resale Authorization can no longer be used by a channel partner to create a private offer. An existing private offer won't be impacted.

To restrict your Resale Authorization, call the StartChangeSet API operation with the RestrictResaleAuthorization change type, as shown in the following example.

Important

This is a non-reversible operation. After the Resale Authorization is marked as Restricted, it can't be in an Active state again.

Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

Provide information for the fields to add the `RestrictResaleAuthorization` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
 - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request. It must be an empty object for `RestrictResaleAuthorization`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The schema validations are specific to `RestrictResaleAuthorization` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
DetailsDocument	Must be empty	422
RestrictResaleAuth orization	Expired ResaleAuthorization can't be marked as Restricted	422

Input field	Validation rule	HTTP code
An unknown property	No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to `RestrictResaleAuthorization` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. or more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_STATUS	Expired <code>ResaleAuthorization</code> can't be marked as restricted.

Release a Resale Authorization and make it visible to a Channel Partner

You can use the Catalog API to initiate your `ResaleAuthorization` to an `Active` state.

`ReleaseResaleAuthorization` makes your `Resale Authorization` active so that a Channel Partner can use your `Resale Authorization` to create private offers.

To release your `Resale Authorization`, call the `StartChangeSet` API operation with the `ReleaseResaleAuthorization` change type, as shown in the following example.

Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "ReleaseResaleAuthorization",
      "Entity":
      {
```

```
    "Type": "ResaleAuthorization@1.0",
    "Identifier": "resaleauthz-123456789"
  },
  "DetailsDocument": {}
}
```

Provide information for the fields to add the `ReleaseResaleAuthorization` change type:

- **Entity** (object) (required) – Your Resale Authorization.
 - **Type** (string) (required) – The Type is always `ResaleAuthorization@1.0`.
 - **Identifier** (string) (required) – Your Resale Authorization ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – Specifics of the request. It must be empty for `ReleaseResaleAuthorization`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The schema validations are specific to `ReleaseResaleAuthorization` actions in the AWS Marketplace Catalog API. The validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
An unknown property	No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to `ReleaseResaleAuthorization` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more details about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
MISSING_MANDATORY_TERMS	Provide a <code>BuyerLegalTerm</code> .
MISSING_MANDATORY_TERMS	Provide a <code>PricingTerm</code> .
INCOMPATIBLE_PRODUCT	Use an active product in limited or public state.
INCOMPATIBLE_PRICING_TERM	<code>PaymentScheduleTerm</code> and <code>FixedUpfrontPricingTerm</code> must be present together.
INCOMPATIBLE_BUYER_TARGETING	At least one Buyer account must be present for <code>ResaleAuthorization</code> with <code>PreExistingBuyerAgreement</code> .
MISSING_MANDATORY_TERMS	Provide at least one of [x,y,z].
INCOMPATIBLE_STATUS	[x] request can't be performed after the resale authorization is released.

Describe an existing Resale Authorization

To describe Resale Authorization details, call the `DescribeEntity` API operation with the `ResaleAuthorization@1.0` entity type, as shown in the following example.

Request Syntax

```
GET /DescribeEntity?catalog=<Catalog>&entityId=<EntityId> HTTP/1.1
```

Provide information for the fields to add the DescribeEntity change type:

- **catalog** (string) – The catalog related to the request. Fixed value: AWSMarketplace.
- **entityId** (string) – The unique ID of the ResaleAuthorization to describe.

Response Syntax

The response to this request gives you the offer details and looks like the following.

```
{
  "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"
    }
]
}
}

```

The following is information about the fields you see in the `DescribeEntity` response.

- **EntityType** (string) – The named type of the entity, which is `ResaleAuthorization@1.0`.
- **EntityIdentifier** (string) – The identifier of the entity, in the format of `EntityId@RevisionId`.
- **EntityArn** (string) – The ARN associated to the unique identifier for the change set referenced in this request.
- **LastModifiedDate** (string) – The last modified date of the entity, in ISO 8601 format (2018-02-27T13:45:22Z).
- **DetailsDocument** (object) (required) – This JSON string includes the details of the entity.
 - **Name** (string) – Name associated with the `ResaleAuthorization` for better readability to you and your Channel Partners. It's displayed as part of the Agreement information.
 - **Description** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
 - **ProductId** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
 - **AgreementToken** (string) – Generated from content in `ResaleAuthorization`. It contains information about terms, rules, and proposer while creating an agreement. It's used for authorization checks and validations during procurement.
 - **Terms** (array of structures) – List of terms presented for acceptance.
 - **Rules** (array of structures) – List of rules or set of instructions.

Work with channel partner private offers using the AWS Marketplace APIs

You can use the AWS Marketplace Catalog API to automate tasks for working with channel partner private offers (CPPOs).

When you create or update a CPPO, the draft offer will contain the terms and rules from a Resale Authorization and will be invisible to the buyer. It's possible, but not required, to involve multiple personas in your organization to create a private offer.

For example, one persona can be responsible for updating prices while a second persona can be responsible for updating the payment schedule. Then, a third persona can be responsible for updating legal terms. You can give a persona permission to update certain parts of an offer. However, you can give only read permissions to Resale Authorizations.

As a prerequisite for calling change types, you must have received one or more Resale Authorizations and be familiar working with AWS Marketplace Catalog API.

For more information, see [Channel partner private offers](#) in the *AWS Marketplace Seller Guide*.

The following topics describe how to use the Catalog API to create and update CPPOs:

Topics

- [CPPO prerequisites](#)
- [Create a CPPO](#)
- [Create a channel partner private replacement offer](#)
- [Update markup](#)
- [Update targeting configuration](#)
- [Update legal resources](#)
- [Update the discoverability of the CPPO](#)
- [Define the expiration date of agreements](#)
- [Update pricing](#)
- [Update payment schedule details](#)
- [Publish the CPPO](#)
- [Define an existing CPPO](#)

CPPO prerequisites

Service-linked role for ResaleAuthorization (SLR) setup is a mandatory pre-requisite to use resale authorization to create a CPPO. To use Resale Authorization, both independent software vendors (ISVs) and AWS Marketplace Channel Partners must create a service-linked role that provides resource-sharing permissions to AWS. If both groups don't perform this prerequisite, AWS can't share the authorization resource from the ISV to the AWS Marketplace Channel Partner. For more information, see [Using roles for Resale Authorization for AWS Marketplace](#) in the *AWS Marketplace Seller Guide*.

Create a CPPO

You use a Resale Authorization targeted to you to create a channel partner private offer (CPPO) in Draft state in AWS Marketplace.

If your request is processed successfully, AWS Marketplace Catalog API generates an offer in Draft state for you with Resale Authorization terms. You can use `DescribeEntity` to see the terms applied to the draft offer from Resale Authorization. This is an incomplete offer and not visible to buyers in AWS Marketplace. You then use change types associated with the CPPO to complete the offer.

After the offer is completed, you use the `ReleaseOffer` change type to complete the offer creation process and release the offer. This will validate the entire offer and make your offer visible to buyers in AWS Marketplace.

To create a channel partner private offer, call the `StartChangeSet` API operation with the `CreateOfferUsingResaleAuthorization` change type, as shown in the following example.

Request Syntax

```
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"
    }
  }
]
```

Provide information for the fields to add the `CreateOfferUsingResaleAuthorization` change type:

- **Entity** (object) (required) – Your CPPO.
- **Type** (string) (required) – The Type is always `Offer@1.0`.
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **ResaleAuthorizationId** (string) (required) – The unique identifier that includes product, terms, and rules that are being offered. Channel partners can add additional terms and rules using update change types. ResaleAuthorization must be available and targeted to you as a partner.
 - **Name** (string) (optional) – The name associated with the offer for better readability. It is displayed as a part of the agreement information.
 - **OfferSetId** (string) (optional) – The ID of the offer set to associate this offer with. Only specify this field when creating an offer that will be part of an offer set. If OfferSetId is not provided, an individual offer will be created that can be purchased standalone. Note that specifying an OfferSetId during offer creation only indicates your intent to associate the offer with that offer set. To complete the association, you must [use the AssociateOffers change type](#) after the offer is created.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

The response to this request gives you the status of the request. If the status is SUCCEEDED, then a new OfferId is generated.

```
{
  "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": []
    }
  ]
}
```

You can use the GET DescribeEntity request to describe the draft offer rules and terms created from ResaleAuthorization in the AWS Marketplace Catalog API Reference. For more information, see [DescribeChangeSet](#).

Terms and rules from ResaleAuthorization

- **LegalTerms** – Provisions describing legal terms, such as the EULA in the ResaleAuthorization will be added to the draft offer. You can add legal terms using the UpdateLegalTerms change type. For more information, see [UpdateLegalTerms](#).

- **PricingTerms** – All the pricing terms (ConfigurableUpfrontPricingTerm, FixedUpfrontPricingTerm, UsageBasedPricingTerm, PaymentScheduleTerms) described by the Manufacturer in the ResaleAuthorization will be added to the draft offer. You can choose to increase the pricing (for each dimension) for your targeted buyers using the UpdateMarkup change type. For more information, see [UpdateMarkup](#) in this guide.
- **PaymentTerms** – If the manufacturer has defined the Future Payment Schedule in the ResaleAuthorization, then you will be able to see the payment terms in the draft offer. You can choose to increase the payment schedule amount for your targeted buyers using the UpdateMarkup change type. If you want to set the payment schedule for your buyers, you can use UpdatePaymentScheduleTerms. For more information, see [UpdatePaymentScheduleTerms](#).
- **TargetingRule** – If the ResaleAuthorization is targeted to specific buyers, then channel partners can give private offers to a subset of buyers using PositiveTargeting. By default, the rule will include all the buyers from ResaleAuthorization. You can select specific buyers and update the draft offer using the UpdateTargeting change type. For more information, see [UpdateTargeting](#).

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

Synchronous Validations

The following schema validations are specific to `CreateOfferUsingResaleAuthorization` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Name	Required	422
	Length must be between 1 and 150 characters	

Input field	Validation rule	HTTP code
Description	Required Length must be between 1 and 255 characters	422
ResaleAuthorizationId	Required Length must be between 1 and 50 characters	422
ResaleAuthorizationId	ResaleAuthorization must be targeted to the channel partner.	422
ResaleAuthorizationId	ResaleAuthorization must be active	422
Channel Partner	Channel Partner must be paid seller in AWS Marketplace	422

Asynchronous Errors

The following errors are specific to `CreateOfferUsingResaleAuthorization` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	Use a ResaleAuthorization in active state.
INCOMPATIBLE_PRODUCT	OfferSetId isn't supported in offers for the product.

Create a channel partner private replacement offer

You can use the Catalog API to create a channel partner private replacement offer in AWS Marketplace.

You use the `ResaleAuthorization` targeted to you and an Agreement of which you are the proposer to create a channel partner private replacement offer in Draft state in the Catalog API by calling `StartChangeSet` with the `CreateReplacementOfferUsingResaleAuthorization` change type, as shown in the following example. Replacement offers can be used to replace an agreement from a previous offer before it ends.

`CreateReplacementOfferUsingResaleAuthorization` will create a draft offer with the agreement acceptor in targeting. This targeting cannot be changed afterwards. The draft offer will also contain the source offer id of the agreement.

To create a channel partner private replacement offer, call the `StartChangeSet` API operation with the `CreateReplacementOfferUsingResaleAuthorization` change type, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the `CreateReplacementOfferUsingResaleAuthorization` change type:

- **Entity** (object) (required) – Your CPPO.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.

- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **ResaleAuthorizationId** (string) (required) – ResaleAuthorizationId is the unique identifier which includes product, terms and rules are being offered. Channel partners can add additional terms and rules using Update change types.

ResaleAuthorization must be available and targeted to you as a partner.

- **Name** (string) (optional) – Name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.
- **AgreementId** (string) (required) – AgreementId is the unique identifier of the agreement created when the targeted buyer accepted the previous offer you are trying to replace

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to CreateReplacementOfferUsingResaleAuthorization actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Name	Length must be between 1 and 150 characters

Input field	Validation rule
AgreementId	Provided agreement must be active Provided agreement must exist Provided agreement must be owned by Channel Partner
ResaleAuthorizationId	Required Length must be between 1 and 50 characters
ResaleAuthorizationId	ResaleAuthorization must be targeted to the channel partner.
ResaleAuthorizationId	ResaleAuthorization must be active
Channel Partner	Channel Partner must be paid seller in AWS Marketplace

Asynchronous Errors

The following errors are specific to `CreateReplacementOfferUsingResaleAuthorization` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	Use a ResaleAuthorization in active state.
INCOMPATIBLE_RESALE_AUTHORIZATION	The ResaleAuthorization must be for the same product that is associated with the agreement.
INCOMPATIBLE_RESALE_AUTHORIZATION	Use a ResaleAuthorization targeted to the acceptor of the agreement.

Update markup

You can use the Catalog API to update pricing terms by a percentage value in your offer in AWS Marketplace.

This will apply the given percentage markup on all pricing terms and payment terms (for future payment schedules) that are defined by the manufacturer in the ResaleAuthorization. Any existing markup will be overwritten. You can view updated pricing and payment terms using `DescribeEntity`.

To update markup, call the `StartChangeSet` API operation with the `UpdateMarkup` change type, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdateMarkup` change type:

- **Entity** (object) (required) – Your CPPO.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.

- **Percentage** (string) (required) – Percentage value will be added to the manufacturer pricing or payment terms.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdateMarkup` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Percentage	Required
	Data type is "String"
	Must be non-negative
	Allow up to 9 decimals

Asynchronous Errors

The following errors are specific to UpdateMarkup actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_MARKUP	The requested change can't be performed after the offer is released.
INCOMPATIBLE_RESALE_AUTHORIZATION	MarkupPercentage can't be updated when PaymentScheduleTerm or FixedUpfrontPricingTerm are present in offer and not present in ResaleAuthorization .
INCOMPATIBLE_TERMS	Use either UpdatePaymentScheduleTerms with specific payment amount or UpdateMarkup with a single markup percentage for the scheduled dates.
INVALID_MARKUP_PERCENTAGE	UpdateMarkup can only be invoked for offers created using ResaleAuthorization.

Update targeting configuration

You can use the Catalog API to replace the existing targeting configuration completely in AWS Marketplace.

Any existing targeting options that are not included in the latest request will be removed from the offer. Manufacturers can mention specific targeted buyers in ResaleAuthorization. Channel partners can give private offers to a subset of buyers using PositiveTargeting in the UpdateTargeting change type.

To update targeting configurations of your offer, call the StartChangeSet API operation with the UpdateTargeting change type, as shown in the following example.

Request Syntax

```
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"
          ]
        }
      }
    }
  ]
}

```

Provide information for the fields to add the `UpdateTargeting` change type:

- **Entity** (object) (required) – Your CPPO.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **PositiveTargeting** (object) (optional) – Positive targeting defines the criteria which any buyer's profile should fulfill in order to be allowed to access the offer. This field is optional, but at least one targeting option should be provided when this field is present.

- **CountryCodes** (array of strings) (optional) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a country, this field should be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code, using this format: ISO 3166-1 alpha-2.
- **BuyerAccounts** (array of strings) (optional) – List as an option to allow targeting based on AWS accounts (also known as, Private Offer). If the intention is to not target the offer to an AWS account, this field should be omitted.
- **NegativeTargeting** (object) (optional) – Negative targeting defines the criteria which any customer's profile should fulfill to be restricted to access the offer. Although this field is optional, at least one targeting option should be provided when this field is present.
- **CountryCodes** (array of strings) (required) – List as option for allowing targeting based on country. If the intention isn't to target the offer to a specific country, then this field should be omitted. If it's present, the list must contain at least one country code. Each element in this list should be a valid 2-letter country code using this format: ISO 3166-1 alpha-2.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdateTargeting` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Details	Required	422
PositiveTargeting	Optional	422
NegativeTargeting	Optional	422
PositiveTargeting.CountryCodes	Optional Country codes must be valid (ISO 3166-1 alpha-2)	422
PositiveTargeting.BuyerAccounts	Optional AWS account IDs must be in valid format (12-digit number) Must not contain more than 25 accounts	422
NegativeTargeting.CountryCodes	Optional Country codes must be valid (ISO 3166-1 alpha-2)	422
NegativeTargeting.BuyerAccounts	Must not be provided (negative targeting on BuyerAccounts isn't supported)	422

Asynchronous Errors

The following errors are specific to `UpdateTargeting` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_BUYER_ACCOUNTS	Provide valid buyer accounts. Invalid accounts: [x].
INVALID_COUNTRY_CODES	Provide supported country codes.
INVALID_TARGETING	Use either negative or positive targeting on the same attribute.
INCOMPATIBLE_PRODUCT	Country-based targeting isn't supported for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide BuyerAccounts that are compatible with the ResaleAuthorization.
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TARGETING	The requested change can't be performed after the offer is expired.
INCOMPATIBLE_TARGETING	Targeting can't be updated on a replacement offer. If the buyer isn't associated with the provided AgreementId, then create a new private offer by providing an AgreementId associated with the buyer.
TOO_MANY_BUYER_ACCOUNTS	Provide BuyerAccounts within the allowed limits.

Update legal resources

You can use the Catalog API to merge the Resale Authorization legal terms and replace the existing legal terms completely in AWS Marketplace.

This change doesn't affect existing agreements. The legal terms that aren't included in the latest request will be removed from the offer. You can view the merged legal terms by calling `DescribeEntity`.

To update legal terms of your offer, call the `StartChangeSet` API operation with the `UpdateLegalTerms` change type, as shown in the following example.

Request Syntax

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

Provide information for the fields to add the `UpdateLegalTerms` change type:

- **Entity** (object) (required) – Your CPPO.
- **Type** (string) (required) – The Type is always `Offer@1.0`.
- **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.

- **Terms** (array of structures) (required) – List of legal terms that you want to update. Supported legal terms are:
- **LegalTerm** (object) (required) – Defines the list of text agreements to be proposed to the acceptors. One example of such an agreement is the end user license agreement (EULA).
- **Type** (string) (required) – Category of term being updated.
- **Documents** (array of structures) (required) – List of references to legal resources to be proposed to the buyers. One example of such a resource is the end user license agreement (EULA). Each reference is made up of a Type and a URL:
 - **Type** (string) (required) – Type of document. Available document types are:
 - **CustomEula** – A custom EULA provided by you as seller. A URL for a EULA stored in an accessible S3 bucket is required for this document type.
 - **StandardEula** – Standard Contract For AWS Marketplace (SCMP). For more information about SCMP, see the AWS Marketplace Seller Guide. You don't provide a URL for this type because it is managed by AWS Marketplace.
 - **Url** (string) (conditionally required) – A URL to the legal document for buyers to read. Required when Type is one of the following [CustomEula].
 - **Version** (string) (conditionally required) – A version of standard contracts provided by AWS Marketplace. This is required when Type is StandardEula. Available versions are:
 - **2022-07-14** – This version of the Standard Contract for AWS Marketplace is available from this Amazon S3 bucket: <https://s3.amazonaws.com/aws-mp-standard-contracts/Standard-Contract-for-AWS-Marketplace-2022-07-14.pdf>

A change set is created for your request. The response to this request gives you the ID for the change set.

Response Syntax

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

The change request is added to a queue and processed. This includes validating information with the AWS Marketplace Seller Operations team to ensure it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours. You can check the

status of the request through the AWS Marketplace Management Portal, or in the Catalog API with the `DescribeChangeSet` action.

Synchronous Validations

The following schema validations are specific to `UpdateLegalTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required	422
	Only LegalTerm is allowed in the list	
	List size must be 1	
Terms[].LegalTerm.Documents	Required	422
Terms[].LegalTerm.Documents[].Type	Required	422
	Allowed values:	
	- CustomEula - StandardEula	
Terms[].LegalTerm.Documents[].Url	Required and must be a valid URL when "Type" is one of	422
	CustomEula	

Asynchronous Errors

The following errors are specific to `UpdateLegalTerms` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_LEGAL_DOCUMENTS	Provide URLs for legal documents stored in accessible S3 buckets.
INVALID_LEGAL_DOCUMENTS	Only the most recent version of StandardEula is supported for new offers.
INVALID_LEGAL_DOCUMENTS	Provide legal documents in the supported file formats.
INVALID_LEGAL_DOCUMENTS	Provide legal documents using the supported document types.
LIMIT_EXCEEDED_LEGAL_DOCUMENT_SIZE	Provide legal documents within the allowed size limits.

Update the discoverability of the CPPO

You can use the Catalog API to manage the discoverability of your offer in AWS Marketplace. This change type doesn't affect existing agreements.

You can either choose to set a specific date in the future to restrict the discoverability of your offer or in the past to expire your offer.

To manage the discoverability of your offer, call the `StartChangeSet` API operation with the `UpdateAvailability` change type, as shown in the following example.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the UpdateAvailability change type:

- **Entity** (object) (required) – Your CPPO.
 - **Type** (string) (required) – The Type is always Offer@1.0.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **AvailabilityEndDate** (string) – Date until when the offer is discoverable and purchasable in AWS Marketplace. You can choose to set a specific date in the future to restrict the availability or in the past to expire the offer. Dates are represented in YYYY-MM-DD format. Offer expires at 23:59:59.999 UTC on the date provided.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to UpdateAvailability actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
AvailabilityEndDate	Required	422
	Format: "YYYY-MM-DD"	

Asynchronous Errors

The following errors are specific to UpdateAvailability actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_AVAILABILITY_END_DATE	AvailabilityEndDate isn't supported for public offers.
INVALID_AVAILABILITY_END_DATE	Provide a future AvailabilityEndDate.
INVALID_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before AgreementEndDate.
MISSING_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate that is before the agreement's end date.

Define the expiration date of agreements

You can use the Catalog API to define the expiration date of the agreements that are created using this offer in AWS Marketplace.

This change does not affect existing agreements. The manufacturer could mention maximum agreement start date in a Resale Authorization. However, channel partners can't provide an agreement start date later than that date.

To define the expiration date of agreements, call the `StartChangeSet` API operation with the `UpdateValidityTerms` change type, as shown in the following example.

Request Syntax

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

Provide information for the fields to add the `UpdateValidityTerms` change type:

- **Entity** (object) (required) – Your CPPO.

- **Type** (string) (required) – The Type is always `Offer@1.0`.
- **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
- **Terms** (array of structures) – List of validity terms that you want to update. Supported validity terms are:
 - **ValidityTerm** (object) – Defines the conditions that will keep an agreement, created from this offer, valid.
 - **Type** (string) – Category of the term being updated.
 - **AgreementDuration** (string) – Defines the duration that the agreement remains active. If `AgreementStartDate` isn't provided, agreement duration is relative to the agreement signature time. The duration is represented in the ISO_8601 format.
 - **AgreementStartDate** (string) – Defines the date when agreement starts. `AgreementStartDate` is represented in YYYY-MM-DD format. The agreement starts at 00:00:00.000 UTC on the date provided. If `AgreementStartDate` isn't provided, agreement start date is determined based on agreement signature time.
 - **AgreementEndDate** (string) – Defines the date when the agreement ends. The `AgreementEndDate` is represented in YYYY-MM-DD format. The agreement ends at 23:59:59.999 UTC on the date provided. If `AgreementEndDate` isn't provided, the agreement end date is determined by the validity of individual terms.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to UpdateValidityTerms actions in the AWS Marketplace Catalog API. These validations are performed when you call StartChangeSet. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP code
Terms	Required Only "ValidityTerm" is allowed in the list Must be empty or contain only 1 term	422
Terms[].ValidityTerm	Supported use cases: 1. ValidityTerm with only Agreement Duration 2. ValidityTerm with only Agreement StartDate 3. ValidityTerm with only Agreement EndDate 4. ValidityTerm with both AgreementStartDate and Agreement EndDate	422
Terms[].ValidityTerm.AgreementDuration	Optional Represented in ISO_8601 format.	422
Terms[].ValidityTerm.AgreementStartDate	Optional Format: "YYYY-MM-DD"	422

Input field	Validation rule	HTTP code
Terms[].ValidityTerm.AgreementEndDate	Optional Format: "YYYY-MM-DD"	422

Asynchronous Errors

The following errors are specific to UpdateValidityTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_AGREEMENT	AgreementStartDate can't be in the future when the current agreement to be replaced isn't future dated.
INCOMPATIBLE_AGREEMENT_END_DATE	AgreementEndDate can't be updated after the offer is released.
INCOMPATIBLE_AGREEMENT_START_DATE	AgreementStartDate can't be updated after the offer is released.
INCOMPATIBLE_PRODUCT	AgreementStartDate in the future isn't supported.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between AgreementStartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementStartDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementEndDate is compatible with the ResaleAuthorization.

Error code	Error message
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the duration between AgreementStartDate and AgreementEndDate is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure AgreementDuration matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_TERMS	ValidityTerm isn't supported for public offers.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide AgreementDuration that is greater than or equal to [x] days.
INVALID_AGREEMENT_END_DATE	Provide a future AgreementEndDate.
INVALID_AGREEMENT_END_DATE	Provide AgreementEndDate that is after or equal to [x].
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is after AvailabilityEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is before the AgreementEndDate.
INVALID_AGREEMENT_START_DATE	Provide an AgreementStartDate that is within [x] years from today.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementDuration and AgreementEndDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with both AgreementStartDate and AgreementDuration isn't supported in an offer for the product.
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with AgreementStartDate isn't supported in an offer for the product.

Error code	Error message
INVALID_AGREEMENT_TIME_INTERVAL	ValidityTerm with only AgreementStartDate isn't supported.
INVALID_AGREEMENT_TIME_INTERVAL	AgreementEndDate isn't supported unless it's used in combination with a future Agreement StartDate or for replacement offers.
INVALID_AGREEMENT_TIME_INTERVAL	Provide AgreementStartDate and Agreement EndDate where the difference is less than or equal to [x] years.
MISSING_AGREEMENT_START_DATE	Ensure AgreementStartDate is present in ValidityTerm when used along with ConfigurableUpfrontPricingTerm.

Update pricing

You can use the Catalog API to replace the existing pricing terms completely.

The pricing terms that aren't included in the latest request will be removed from the offer. Channel partners can use this change type only to pass FixedUpFrontPricingTerm.

To update pricing terms for your offers, call the StartChangeSet API operation with the UpdatePricingTerms change type, as shown in the following example.

Request Syntax

```
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
            }
          ]
        }
      ]
    }
  ]
}

```

Provide information for the fields to add the `UpdatePricingTerms` change type:

- **Entity** (object) (required) – Your CPPO.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **Details** (string) (required) – Specifics of the request. This field is a JSON string field. It must be formatted properly for a single-line string field, including escaping characters (such as quotation marks) that can't be in a string.
- **PricingModel** (string) (required) – Pricing model for your offer. Possible values for pricing model are:
 - **Contract** – Contract-based pricing model where buyers are either billed in advance for the use of your product, or offered a flexible payment schedule. Buyers can also pay for an additional usage above their contract.
 - **Terms** (array of structures) (required) – List of pricing terms that you want to update. Supported pricing terms are:
 - **FixedUpfrontPricingTerm** (object) – Defines a pre-paid pricing model where the customers are charged a fixed upfront amount.

- **Type** (string) (required) – Type of the term being updated.
- **CurrencyCode** (string) (required) – Defines the currency for the prices mentioned in this term. For public offers, only USD is supported. For private offers, USD, AUD, EUR, GBP, and JPY are supported.
- **Price** (string) (required) – Fixed amount to be charged to the customer when this term is accepted.
- **Grants** (array of structures) (required) – Entitlements that will be granted to the acceptor of fixed upfront as part of agreement execution.
- **DimensionKey** (string) (required) – Unique dimension key defined in the product document. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.
- **MaxQuantity** (integer) (optional) – Maximum amount of capacity that the buyer can be entitled to the given dimension of the product. If `MaxQuantity` is not provided, the buyer will be able to use an unlimited amount of the given dimension.
- **Duration** (string) (optional) – Defines the duration that the term remains active. This field supports the ISO 8601 format.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdatePricingTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
<code>PricingModel</code>	Required Allowed pricing models: Contract
<code>Terms</code>	Required Allowed terms: <code>FixedUpfrontPricingTerm</code>
<code>Terms[].FixedUpfrontPricingTerm.CurrencyCode</code>	Required Allowed values: ["USD", "AUD", "EUR", "GBP", "JPN"] Allowed pricing models: Contract
<code>Terms[].FixedUpfrontPricingTerm.Price</code>	Required Data type is "String" Must be non-negative Support up to 6 Decimals No special character supported
<code>Terms[].FixedUpfrontPricingTerm.Duration</code>	Required Expected format: ISO 8601 duration
<code>Terms[].FixedUpfrontPricingTerm.Grants[].DimensionKey</code>	Required Length must be between 1 and 60

Input field	Validation rule
Terms[].FixedUpfrontPricingTerm.Grants[].MaxQuantity	Required

Asynchronous Errors

The following errors are specific to UpdatePricingTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
DUPLICATE_DIMENSION_KEYS	Provide Grants with a unique list of dimension keys in [x].
DUPLICATE_DIMENSION_KEYS	Provide RateCard with a unique list of dimension keys in [x].
DUPLICATE_SELECTORS	Provide a unique list of Selectors in ConfigurableUpfrontPricingTerm.
DUPLICATE_TERM_TYPES	Provide a unique list of term types.
INCOMPATIBLE_AGREEMENT	The following terms can't be removed from the replacement offer: [x, y, z].
INCOMPATIBLE_AGREEMENT	The following terms can't be added to the replacement offer: [x, y, z].
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the CurrencyCode.

Error code	Error message
INCOMPATIBLE_PRODUCT	Usage pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Contract pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Byol pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	Free pricing model isn't supported for the product.
INCOMPATIBLE_PRODUCT	[x] isn't supported in an offer for the product.
INCOMPATIBLE_PRODUCT	Provided payment and pricing terms are incompatible.
INCOMPATIBLE_PRODUCT	Use existing, available dimensions in the product in [x].
INCOMPATIBLE_PRODUCT	FreeTrialPricingTerm as the offer's only pricing term isn't supported for the product.
INCOMPATIBLE_PRODUCT	The following terms aren't supported for the product: [x,y,z].
INCOMPATIBLE_PRODUCT	Replacement offers are only supported for contract pricing model.
INCOMPATIBLE_PRODUCT	Provide pricing term(s) that are compatible with the product dimensions. Incompatible pricing terms: [x,y,z].
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Allowed in ConfigurableUpfrontPricingTerm for usage pricing model.

Error code	Error message
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	Set MultipleDimensionSelection and QuantityConfiguration to Disallowed in ConfigurableUpfrontPricingTerm for usage pricing model.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	QuantityConfiguration in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.
INCOMPATIBLE_RATE_CARD_CONSTRAINTS	MultipleDimensionSelection in ConfigurableUpfrontPricingTerm can't be changed after the offer is released.
INCOMPATIBLE_RATES	Set all charge amounts and prices to zero (0) when using Free pricing model.
INCOMPATIBLE_RATES	Only zero (0) prices are allowed in UsageBasedPricingTerm for a free trial offer for the product.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide the same CurrencyCode that is specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure Duration in FixedUpfrontPricingTerm matches duration specified in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [x, y, z].
INCOMPATIBLE_SELECTOR_DURATION	Durations aren't allowed to be removed from rate cards in ConfigurableUpfrontPricingTerm after the offer released.
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.

Error code	Error message
INCOMPATIBLE_TERMS	[x] isn't supported together with the following terms: [y,z].
INCOMPATIBLE_TERMS	The following terms can't be added after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	The following terms can't be removed after the offer is released: [x,y,z].
INCOMPATIBLE_TERMS	[x] isn't supported for private offers.
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm that grants unlimited usage: [x,y,z].
INCOMPATIBLE_TERMS	The following terms aren't supported with FreeTrialPricingTerm for the product: [x,y,z].
INCOMPATIBLE_TERMS	Provide zero (0) price for FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
INCOMPATIBLE_TERMS	The following terms aren't compatible with the PricingModel: [x,y,z].
INCOMPATIBLE_TERMS	FixedUpfrontPricingTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_AGREEMENT_DURATION	Provide duration between [x] and [y] months.
INVALID_AGREEMENT_DURATION	Ensure duration granularity is at the day level for metered dimensions.

Error code	Error message
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.
INVALID_DURATION	Ensure Duration in FreeTrialPricingTerm is within the allowed range.
INVALID_DURATION	Provide Duration in FixedUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
INVALID_DURATION	Provide duration between [x] and [y] months.
INVALID_DURATION	Ensure duration granularity is at the day level for metered dimensions.
INVALID_GRANTS	Provide the same MaxQuantity for all Grants in FreeTrialPricingTerm.
INVALID_GRANTS	Provide Grants for all available metered dimensions in FreeTrialPricingTerm.
INVALID_PRICE_CHANGE	[x] can't be updated until [y] because you have requested a price increase in the past 120 days. To cancel your previous price increase request or for more information, contact the AWS Marketplace Managed Catalog Operations Team.
INVALID_PRICE_CHANGE	Price increase and dimension addition in [x] isn't supported in the same request. Add dimensions first.
INVALID_PRICE_CHANGE	Price increase and decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.

Error code	Error message
INVALID_PRICE_CHANGE	Price increase in RecurringPaymentTerm and price decrease in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_PRICE_CHANGE	Price decrease in RecurringPaymentTerm and price increase in UsageBasedPricingTerm isn't supported in the same request. Decrease prices first.
INVALID_RATE_CARD	ConfigurableUpfrontPricingTerm is missing one or more-dimension keys for duration [x]. Provide prices for the same set of dimension keys for all durations.
INVALID_RATE_CARD	Provide a rate card for only metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Rates can't be removed from [x]. Provide prices for all dimensions in the existing rate card.
INVALID_RATE_CARD	Provide dimensions that have the same unit in [x].
INVALID_RATE_CARD	Provide either all metered or all entitled dimensions in [x].
INVALID_RATE_CARD	Provide only entitled dimensions in [x].
INVALID_RATE_CARD	Provide usage based rates for all available metered dimensions in UsageBasedPricingTerm.
INVALID_RATE_CARD	Provide usage based rates for all free trial dimensions.

Error code	Error message
INVALID_RATE_CARD	Provide prices with up to 8 decimal places in UsageBasedPricingTerm.
INVALID_SELECTOR_DURATION_VALUE	Provide duration between [x] and [y] months.
INVALID_SELECTOR_DURATION_VALUE	Ensure duration granularity is at the day level for metered dimensions.
INVALID_SELECTOR_DURATION_VALUE	Ensure Duration in ConfigurableUpfrontPricingTerm is within the allowed range.
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations.
INVALID_SELECTOR_DURATION_VALUE	Provide one or more supported contract durations or a single custom duration.
INVALID_SELECTOR_DURATION_VALUE	Provide Duration in ConfigurableUpfrontPricingTerm that matches the duration between AgreementStartDate and AgreementEndDate.
MISSING_DURATION	Provide Duration in FixedUpfrontPricingTerm.
MISSING_MANDATORY_TERMS	FixedUpfrontPricingTerm is only supported when paired with ByolPricingTerm or PaymentScheduleTerm.
MISSING_MANDATORY_TERMS	Provide at least one of [x,y,z].
MISSING_MANDATORY_TERMS	Provide a ByolPricingTerm when using Byol pricing model.
TOO_MANY_GRANTS	Provide up to [x] grants in [y].
TOO_MANY_RATE_CARDS	Only one rate card in ConfigurableUpfrontPricingTerm is allowed for the product.

Error code	Error message
TOO_MANY_RATE_CARDS	Up to [x] rate cards are allowed in ConfigurableUpfrontPricingTerm for the product.
TOO_MANY_RATES	Provide RateCards within the allowed limits in ConfigurableUpfrontPricingTerm.
TOO_MANY_RATES	Provide RateCards within the allowed limits in UsageBasedPricingTerm.

Update payment schedule details

You can use the Catalog API to change the payment schedule details, such as flexible payment schedule, in AWS Marketplace.

If the manufacturer has provided a payment schedule in Resale Authorization, Channel Partner can either:

- Use the UpdateMarkup change type to apply a uniform percentage markup to all payment schedules.
- Use the UpdatePaymentScheduleTerms change type to set custom payment amounts to a value greater than or equal to what is provided in Resale Authorization. Payment dates can't be changed. Both options can't be applied simultaneously. If a markup is applied first and you need to change it to apply a payment schedule, set the markup to 0 through the UpdateMarkup change type. If a payment schedule is applied first you can't revert the change to apply the markup.

To update payment-associated details for your offer, call the StartChangeSet API operation with the UpdatePaymentScheduleTerms change type, as shown in the following example.

Request Syntax

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

```

Provide information for the fields to add the `UpdatePaymentScheduleTerms` change type:

- **Entity** (object) (required) – Your CPPO.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).
- **DetailsDocument** (object) (required) – The JSON value of specifics of the request.
 - **Terms** (array of structures) – List of payment terms that you want to update. Supported payment terms are:
 - **PaymentScheduleTerm** (object) – Defines an installment-based pricing model where customers are charged a fixed price on different dates during the agreement validity period.
 - **Type** (string) – Type of the term being updated. This is the object value: `"PaymentScheduleTerm"`.

- **Schedule** (array of structures) – List of the payment schedule where each element defines one installment of payment. It contains the information necessary for calculating the price to be paid and the date on which the customer would be charged.
- **ChargeDate** (string) – The date on which the customer would pay the price defined in this payment schedule term. ChargeDate is represented in YYYY-MM-DD format. Invoices are generated on the date provided.
- **ChargeAmount** (string) – The price that the customer would pay on scheduled date (ChargeDate).

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `UpdatePaymentScheduleTerms` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule	HTTP
Terms	Required	422
	Only "PaymentScheduleTerm" is allowed	

Input field	Validation rule	HTTP
	List size must be less than 2	
Terms[].PaymentScheduleTerm.CurrencyCode	Required Supported currencies: ["USD", "AUD", "EUR", "GBP", "JPN"]	422
Terms[].PaymentScheduleTerm.Schedule[]	Required List size must be between 1 and 60, inclusive	422
Terms[].PaymentScheduleTerm.Schedule[].ChargeDate	Required Format: "YYYY-MM-DD"	422
Terms[].PaymentScheduleTerm.Schedule[].ChargeAmount	Required Data type is "String" Non-negative decimals with up to 2 decimal places supported No additional properties are allowed	422

Asynchronous Errors

The following errors are specific to UpdatePaymentScheduleTerms actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
DUPLICATE_CHARGE_DATES	Provide unique charge dates in PaymentScheduleTerm.

Error code	Error message
INCOMPATIBLE_MARKUP_PERCENTAGE	PaymentScheduleTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the total ChargeAmounts in PaymentScheduleTerm is compatible with the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the charge dates in PaymentScheduleTerm are same as the charge dates in the ResaleAuthorization.
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the charge amount in PaymentScheduleTerm is greater than or equal to the charge amount in the ResaleAuthorization for that date.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INCOMPATIBLE_TERMS	Use either UpdatePaymentScheduleTerms with specific payment amount or UpdateMarkup with a single markup percentage for the scheduled dates.
INVALID_CHARGE_DATES	Provide charge dates before AgreementEndDate.
TOO_MANY_BACKDATED_CHARGES	Provide up to 1 scheduled payment before AvailabilityEndDate.

Error code	Error message
DUPLICATE_CHARGE_DATES	Provide unique charge dates in PaymentScheduleTerm.
INCOMPATIBLE_CURRENCY_CODE	CurrencyCode can't be changed after the offer is released.
INCOMPATIBLE_MARKUP_PERCENTAGE	PaymentScheduleTerm isn't supported when MarkupPercentage is greater than zero (0).
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the CurrencyCode.
INCOMPATIBLE_RESALE_AUTHORIZATION	Provide term(s) that are compatible with the ResaleAuthorization. Incompatible terms: [PaymentScheduleTerm].
INCOMPATIBLE_RESALE_AUTHORIZATION	Ensure the total ChargeAmounts in PaymentScheduleTerm is compatible with the ResaleAuthorization.
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is released.
INCOMPATIBLE_TERMS	The requested change can't be performed after the offer is expired.
INVALID_CHARGE_DATES	Provide charge dates before AgreementEndDate.
INVALID_CURRENCY_CODE	Provide a supported CurrencyCode.
INVALID_CURRENCY_CODE	Provide the same CurrencyCode across all pricing and payment terms.

Error code	Error message
TOO_MANY_BACKDATED_CHARGES	Provide up to 1 scheduled payment before AvailabilityEndDate.

Publish the CPPO

You can use the Catalog API to merge the information collected from all update change types, and then publish the offer in AWS Marketplace.

Offers remain in a Draft state, until `ReleaseOffer` is called. After the offer is released, it's discoverable in AWS Marketplace.

To publish your offer, call the `StartChangeSet` API operation with the `ReleaseOffer` change type as shown in the following example.

Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseOffer",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-123456789"
      },
      "DetailsDocument": {}
    }
  ]
}
```

Provide information for the fields to add the `ReleaseOffer` change type:

- **Entity** (object) (required) – Your CPPO.
 - **Type** (string) (required) – The Type is always `Offer@1.0`.
 - **Identifier** (string) (required) – Your offer ID. For more information, see [Identifier](#).

- **DetailsDocument** (object) (required) – The JSON value of specifics of the request. It must be empty for `ReleaseOffer`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take anywhere from a few minutes to a few hours.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Synchronous Validations

The following schema validations are specific to `ReleaseOffer` actions in the AWS Marketplace Catalog API. These validations are performed when you call `StartChangeSet`. If the request doesn't meet the following requirements, it will fail with an HTTP response.

Input field	Validation rule
Details	Must be empty ({})

Asynchronous Errors

The following errors are specific to `ReleaseOffer` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_PAYMENT_SETTINGS	Update your payment settings to be compatible with the CurrencyCode.
INCOMPATIBLE_PRODUCT	First create a public offer for the product.
INCOMPATIBLE_SELLER_VERIFICATION	Complete all required seller verification processes.
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer is released.
MISSING_AGREEMENT_END_DATE	Provide an AgreementEndDate for replacement offers.
MISSING_AVAILABILITY_END_DATE	Provide an AvailabilityEndDate for private offer.
MISSING_MANDATORY_TERMS	Provide a FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
MISSING_BUYER_ACCOUNTS	Provide PositiveTargeting with BuyerAccounts for offers created using ResaleAuthorization.
MISSING_BUYER_ACCOUNTS	All offers for the product must be private. Provide PositiveTargeting with BuyerAccounts.
MISSING_DESCRIPTION	Set Description before releasing the offer.
MISSING_MANDATORY_TERMS	Add [x] to the offer.
MISSING_MANDATORY_TERMS	Provide a FixedUpfrontPricingTerm when the offer contains a PaymentScheduleTerm.
MISSING_NAME	Set Name before releasing the offer.

Error code	Error message
TOO_MANY_OFFERS	Only one public offer can be created per product.
TOO_MANY_OFFERS	Only one public free trial offer can be created per product.

Define an existing CPPO

You can use the Catalog API to define CPPO details in AWS Marketplace.

To define an existing CPPO, call the `DescribeEntity` API operation with the `Offer@1.0` entity type, as shown in the following example.

Request Syntax

```
GET /DescribeEntity?catalog=<Catalog>&entityId=<EntityId> HTTP/1.1
```

Provide information for the fields to add the `DescribeEntity` change type:

- **catalog** (string) –The catalog related to the request. Fixed value: `AWSMarketplace`.
- **entityId** (string) – The unique ID of the offer to describe.

Response Syntax

The response to this request gives you the offer details and looks like the following.

```
{
  "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"
    }
]
}
}

```

Provide information for the input fields for the `DescribeEntity` response:

- **EntityType** (string) – The named type of the entity, which is `Offer@1.0`.
- **EntityIdentifier** (string) – The identifier of the entity, in the format of `EntityId@RevisionId`.
- **EntityArn** (string) – The ARN associated to the unique identifier for the change set referenced in this request.
- **LastModifiedDate** (string) – The last modified date of the entity, in ISO 8601 format (`2018-02-27T13:45:22Z`).
- **DetailsDocument** (object) – The JSON object includes the details of the entity.
 - **Id** (string) – Unique identifier for an offer entity in AWS Marketplace and is generated during the creation of an offer.
 - **State** (string) – The status of the offer.
 - **Name** (string) – Name associated with the offer for better readability to you and your customers. It will be displayed as part of Agreement information as well.
 - **Description** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
 - **PreExistingAgreement** (string) – Determines if this offer is a renewal for an existing agreement with an existing customer for the same underlying product. The existing agreement can be within or outside AWS Marketplace. AWS may audit and verify your offer is a renewal. If AWS is unable to verify your offer, then AWS may revoke the offer and entitlements from your customer.
 - **AcquisitionChannel** (string) – Indicates if the existing agreement was signed outside AWS Marketplace or within AWS Marketplace. Possible values: `External`, `AwsMarketplace`.

- **PricingModel** (string) – Indicates which pricing model the existing agreement uses. Possible values: Contract, Usage, Byo1, Free.
- **ProductId** (string) – Description is a free-form text which is meant to be used only by you and will never be exposed to buyers.
- **ResaleAuthorizationId** (string) – ResaleAuthorization is used to create the private offer.
- **MarkupPercentage** (string) – Percentage value that the channel partner passed in the UpdateMarkup change type. This markup is already applied to the terms.
- **Terms** (array of structures) – List of terms.
- **Rules** (array of structures) – List of rules.

Work with renewals

Renewals in AWS Marketplace enable sellers, including independent software vendors (ISVs) and channel partners, to seamlessly extend existing agreements with customers. During the renewal process, sellers have the option to adjust terms by creating a new offer.

There are four types of renewals in AWS Marketplace:

- **Agreement based offers:** Sellers can replace an existing agreement to adjust pricing, duration, terms, and renew an existing contract before it ends. An agreement based offer can be extended to the customer beyond the current agreement's end date, granting new entitlements, discounts, or payment schedules based on the customers needs.

This support applies to software as a service (SaaS) products, including those with contract and consumption-based pricing (CCP), whether they offer flexible payment options or not.

- **Future dated offers:** Sellers can create future dated offers to start on a date in the future, and use them to pre-book renewals while existing terms are still ongoing. Once accepted by the customer, the agreement begins on a specified **future date, allowing to start after the current agreement ends**.

This support applies to software as a service (SaaS) products, including those with contract and consumption-based pricing (CCP), whether they offer flexible payment options or not.

- **New private offer:** Sellers can [create a new private offer](#) with new terms (for AMI hourly, AMI annual, and SaaS pay-as-you-go subscriptions) that can be accepted anytime to renew an existing agreement.

- **Auto-renewal:** Customers can enable auto-renewal for public offers and AWS Data Exchange products with private offers to automatically create a new agreement when a previous agreement ends.

Topics

- [Replacement offers](#)
- [Future dated agreements](#)
- [Resources](#)

Replacement offers

As a seller, you can offer renewals by replacing an active agreement that was originally created when the customer accepted your public offer or private offer. Using a replacement offer, you can extend a new offer to the customer that goes beyond the current agreements end date, grant new entitlements, offer pricing discounts, adjust payment schedules, change the payment schedule, or change the end user license agreement (EULA).

You can use the Catalog API to [create a replacement offer](#) (also known as an agreement-based offer) in AWS Marketplace for [supported product types](#). You will need to provide the unique identifier (agreement ID) of the current agreement you wish to replace. You can find this agreement ID in the [AWS Marketplace Management Portal](#) under the Agreements section or by using the [SearchAgreements](#) API operation.

Once the customer accepts the replacement offer, their current agreement will be replaced with a new agreement that can extend beyond the previous end date.

Note

You cannot create a replacement agreement that specifies a seller of record that differs from the original agreement.

For more information on replacement offers, see [Amending private offers in AWS Marketplace](#) in the *AWS Marketplace Seller Guide*.

Future dated agreements

Future-dated agreements are created when a customer accepts a private offer with a future service start date. To facilitate advance booking of upcoming renewals in AWS Marketplace, ISVs and channel partners can create private offers that start the day after the current agreement ends rather than immediately upon acceptance.

To set the start date of a renewal agreement as the day after the end date of the current agreement, you can use the [Catalog API UpdateValidityTerms change type](#) on a private offer that has already been published. Sellers can choose a service start date up to three years in the future.

Customers can review the terms and conditions of the private offer and accept it before it takes effect. Accepting a private offer with a start date in the future does not replace the current agreement. Instead, it creates a renewal agreement that begins immediately after the previous agreement ends.

For more information on product types that support future dated offers and agreements, see [Creating future dated agreements](#) in the *AWS Marketplace Seller Guide*.

Resources

- For end-to-end labs with working code examples, see:
 - [Lab: Create a private offer \(with a future service start date\)](#)
 - [Lab: Create a replacement private offer](#)
- For a video on creating replacement offers, see [Renew SaaS Contract Private Offers - AWS Marketplace](#) on YouTube.
- For a video on creating future dated offers, see [Create an AWS Marketplace Future Dated Private Offer](#) on YouTube.

Work with multi-product solutions

AWS Marketplace multi-product solutions combine multiple products and services to address specific customer needs. Two capabilities support this:

- **Solutions** – Dedicated pages that show customers how your AWS Marketplace products work together.
- **Offer sets** – Unified transactions that group multiple private offers, allowing customers to accept all offers with a single action.

You can use these capabilities independently or together. Use solutions to improve product discovery, offer sets to streamline procurement, or both for a complete multi-product experience.

Topics

- [Work with solutions](#)
- [Work with offer sets](#)

Work with solutions

A solution is a collection of products and services that work together to solve specific customer problems. Solutions can include AWS Marketplace products and non-AWS Marketplace products. Solutions help you showcase how multiple products integrate to address industry-specific use cases. The solution describes the customer problem, use cases, integration details, and related products.

Solutions in AWS Marketplace and AWS Partner Central

When you migrate from Partner Central 2.0 to Partner Central 3.0, you're prompted to create an AWS account and register as an AWS Marketplace seller. For more information, see [Register and create your seller profile](#). This registration is required to:

- Migrate your existing Partner Central solutions to the AWS Marketplace catalog (all solutions will be migrated to AWSMarketplace catalog)
- Enable your AWS Marketplace solutions to work with Partner Central APIs, such as [Selling APIs](#).

Understanding profile linking and migration

There are two distinct scenarios for connecting your Partner Central and AWS Marketplace accounts:

Profile linking only – If you link your AWS Partner Central and AWS accounts without migrating to Partner Central 3.0, your AWS Marketplace solutions and Partner Central solutions remain separate. You manage each platform independently.

Profile migration – When you migrate to Partner Central 3.0 with linked AWS Marketplace seller profiles:

- Your Partner Central 2.0 solutions migrate to AWS Marketplace

- You manage solutions through AWS Marketplace only
- Partner Central APIs can access migrated solutions

Note

If you have already linked your AWS Partner Central and AWS accounts (see [Linking AWS Partner Central and AWS accounts](#)), you don't need to create a new AWS account. However, verify that your existing AWS account is registered as an AWS Marketplace seller. To make your solution publicly available on AWS Marketplace buyer website, you need a public profile as an AWS Marketplace seller. For AWS Marketplace seller registration instructions, see [Register and create your seller profile](#).

Making pre-existing AWS Marketplace solutions available to Partner Central APIs

AWS Marketplace solutions that you created before the Partner Central 3.0 profile migration aren't automatically recognized by Partner Central APIs. This applies even after you complete the migration and link your AWS account with a Partner Central account.

To make an existing pre-migration solution visible to Partner Central APIs, update any field in the solution. You can enter the same information that's currently there. Repeat this process for each solution you created before the Partner Central 3.0 profile migration.

For example, use the `UpdateInformation` change type to update a solution's name with its current value. After you update a solution once, Partner Central APIs will automatically recognize any future updates to that solution.

Solution entity

A solution is a marketing document that helps buyers discover and understand your offerings. Solutions contain a title, description, use cases, AWS Marketplace products, and non-AWS Marketplace products. Solution is a separate entity type from product, offer, and offer sets. Each solution has its own discovery and listing experience that buyers can browse separately from individual products.

Key characteristics

- Group multiple products, including both AWS Marketplace products and non-AWS Marketplace products.

- Create a dedicated discovery and listing experience for buyers.
- Group products that you own with products from other sellers.

The solution entity type is `Solution@1.0`.

Entity identifiers

Solutions use the `soln-*` identifier format (for example, `soln-abc123def456`).

Entity ARN format

Solutions follow the standard AWS ARN format:

```
arn:{aws-partition}:aws-marketplace:{region}:{account-id}:{aws-marketplace-catalog}/  
Solution/{solution-id}
```

Example:

```
arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/soln-  
abc123def456
```

Getting started with solutions

This section provides detailed information about creating and managing solutions using AWS Marketplace Catalog API [change types](#).

Before you begin, make sure you have:

- Valid AWS Marketplace seller registration and a public profile. For more information, see [Register and create your seller profile](#).
- Access to the API and completed seller prerequisites. For more information, see [Access control for the AWS Marketplace Catalog API](#).
- Learned the basics of using the AWS Marketplace Catalog API, see [Using the AWS Marketplace Catalog API](#).

List your solutions

To list all solutions in your account, call the `ListEntities` API operation and set `EntityType` to `Solution`.

Request

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

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

Response

```
{
  "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
}
```

Filter solutions by visibility

You can filter solutions by visibility using the `FilterList` parameter.

Request

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

{
  "Catalog": "AWSMarketplace",
  "EntityType": "Solution",
  "FilterList": [
    {
      "Name": "Visibility",
      "ValueList": [
```

```
        "Public"
      ]
    }
  ]
}
```

Get solution details

To get detailed information about a specific solution, call the `DescribeEntity` API operation.

Note

The `DescribeEntity` response returns an `EntityIdentifier` that combines an `EntityId` with a `RevisionId`. For solutions, `EntityId` is the solution ID, and `RevisionId` can be used for optimistic locking. For more information, see [Identifiers](#).

Request

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=soln-abc123456 HTTP/1.1
```

Response

```
{
  "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"
}
```

List change history for a solution

To view the change history for a specific solution, call the `ListChangeSets` API operation with an entity filter.

Request Syntax

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

```
{
  "Catalog": "AWSMarketplace",
  "FilterList": [
    {
      "Name": "EntityId",
      "ValueList": [
        "soln-abc123456"
      ]
    }
  ]
}
```

Tag solutions and change sets

You can add tags to solutions and change sets during creation or update them later.

Tag during creation

You can add `EntityTags` only to the `CreateSolution` change type, not to other solution change types. You can add `ChangeSetTags` to any `StartChangeSet` payload.

```
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"  
      }  
    ]  
  }  
}
```

Manage tags on existing resources

Use the `TagResource` and `UntagResource` API operations to manage tags on existing solutions and change sets. Use `ListTagsForResource` to view current tags.

For more information, see [Managing tags on resources](#) and [Adding tags to an entity and change set during creation](#).

Create a solution

You can use the `CreateSolution` change type to create a new solution document with only a solution ID and name. Other values aren't yet filled in.

When your request is processed successfully, AWS Marketplace creates a solution in `Draft` status for you. This incomplete solution isn't visible to buyers on AWS Marketplace.

You then use other change types to complete the solution: `UpdateInformation`, `UpdateRelatedProducts`, `UpdateBuyerEngagementOptions` (optional for creating a Limited solution), and `ReleaseSolution`. The `ReleaseSolution` change type validates that all required fields needed for a Limited solution are present on the solution, and then moves it to Limited visibility.

Note

To move a solution to `Public` status or to change a solution's status, use the `UpdateVisibility` change type after providing the required values.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the `CreateSolution` change type:

- **Entity (object) (required)** – The named type of the solution entity being created.
 - **Type (string) (required)** – The Type must be `Solution@1.0`.
- **DetailsDocument (object) (required)** – The details of the request.
 - **Name (string) (required)** – AWS resource name for seller's own reference only (not visible to buyers). Max length: 100 characters. Must not contain leading or trailing whitespaces, linebreaks or control characters. You can also update the solution tile via the `UpdateInformation` change type.
 - **Title (string) (optional)** – The title for your solution. Max length: 255 characters. Must not contain leading or trailing whitespaces, linebreaks or control characters. You can also set or update the solution title via the `UpdateInformation` change type.

Response Syntax

A change set is created for your request. The response gives you the `ChangeSetId` and `ChangeSetArn`:

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

```
}
```

The change request is added to a queue and processed. You can check the status using the AWS Marketplace Management Portal or the `DescribeChangeSet` API operation.

When the request is completed successfully (if the `Status` is `SUCCEEDED`), a new solution ID is generated.

Asynchronous Errors

`CreateSolution` actions return specific errors, in addition to [common asynchronous errors](#). You receive these errors from `DescribeChangeSet` after a change set finishes processing. To learn how to get change request status, see [Working with change sets](#).

Error code	Error message
INVALID_NAME	The solution Name length isn't valid. Enter a Name that's between 1 and 100 characters and try again.
MISSING_NAME	A Name value is required. Specify a Name value and try again.
INVALID_TITLE	The solution Title length isn't valid. Enter a Title that's between 1 and 255 characters and try again.

Update solution information

If you already have a solution in AWS Marketplace, you can use the Catalog API to update the solution information.

To update solution information, call the `StartChangeSet` API operation with the `UpdateInformation` change type and your updated details. See the following example.

Note

The `UpdateInformation` change types behave like HTTP PATCH operations for top-level attributes. How fields are updated:

- If a top-level attribute is present in the `DetailsDocument`, the workflow validates your input and overwrites existing values.

- If a top-level field is not present in the DetailsDocument, the existing value is preserved.
- To unset an optional field, send an explicit JSON null for that field.

AWS Marketplace verifies and ingests logo and promotional media assets into an AWS Marketplace-owned location. DescribeEntity returns an accessible URL for the ingested files, not your original input from StartChangeSet. To see your original submitted URLs, use the DescribeChangeSet API.

Note

When you modify PromotionalMedia asset files (by providing new URLs) on a Public solution, UpdateInformation requires manual review from the AWS Marketplace Seller Operations team. This increases execution time. In this scenario, use UpdateInformation separately in its own change set.

Request Syntax

```
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"
    }
  ]
}

```

Provide information for the fields to add the UpdateInformation change type:

- **Entity (object) (required)** – The named type of entity being updated.
 - **Identifier (string) (required)** – The solution ID.
 - **Type (string) (required)** – Must be `Solution@1.0`.
- **DetailsDocument (object) (required)** – The details of the request including the information that you want to update for your solution. Each field is optional, but you must include at least one change to update.

- **Name (string)** — The resource name of the solution, visible only to the seller via Catalog API read operations, not visible to buyers. Max length: 100 characters. Must not contain leading or trailing whitespaces, linebreaks or control characters.
- **Title (string)** – The title of the solution to be displayed to buyers. Max length: 255 characters. Must not contain leading or trailing whitespaces, linebreaks or control characters.
- **ShortDescription (string)** – The value proposition description of key aspects of the solution to be displayed to buyers. This is usually 2–3 sentences. Max length: 1000 characters. Must not contain leading or trailing whitespaces or control characters.
- **LongDescription (string)** – The longer description of this solution to be displayed to buyers. This is usually 1–3 paragraphs. Max length: 5000 characters. Must not contain leading or trailing whitespaces or control characters.
- **LogoUrl (string)** – The URL to an image in a publicly accessible Amazon S3 bucket or a presigned S3 URL. Must be a direct S3 URL (not behind CloudFront or API Gateway). Max length: 2048 characters. For logo specifications, see [Company and product logo requirements](#).
- **PromotionalMedia (array of objects)** – The list of promotional images and videos. Max 15 items (5 videos and 10 images). For promotional media specifications, see [Enhance your AWS Marketplace product with promotional media](#).
 - **Type (string) (required)** – Either Image or Video.
 - **Url (string) (required)** – The URL to the media file. Must be a direct S3 URL or presigned S3 URL. Max length: 2048 characters. Must be an https URL.
 - **Title (string) (required)** – The title of the media. Max length: 100 characters. Must not contain leading or trailing whitespaces, linebreaks, or control characters.
 - **Description (string) (required)** – The description of the media. Max length: 200 characters. Must not contain leading or trailing whitespaces or control characters.
 - **PreviewUrl (string) (optional)** – For videos only, the URL to a preview image. Max length: 2048 characters. Must be an https URL.
- **AdditionalResources (array of objects)** – The list of references to additional resources. Max 8 items.
 - **Text (string)** – The name or title of the resource. Max length: 500 characters. Must not contain leading or trailing whitespaces, linebreaks, or control characters.
 - **Url (string)** – The URL to the resource. Max length: 2048 characters. Must be an https URL.
- **UseCases (array of objects)** – The list of use cases that this solution addresses. Max 3 items. For valid use case categories (a combination of Group, SubGroup, and Item), see [use case](#)

[categories list](#) for active accepted values and their [introduction labels](#). The combination of Group, SubGroup, and Item must be a valid item in the use case categories list.

- Group (string) – Industry or technology category. Must be one of the Group values in [use case categories list](#).
- SubGroup (string) – Use case group within the category. Must be one of the SubGroup values in [use case categories list](#).
- Item (string) – Specific use case. Must be one of the Item values in [use case categories list](#).
- Description (string) – Description of how this solution addresses the use case. Max length: 500 characters. Must not contain leading or trailing whitespaces or control characters.

For presigned URL information, see [Sharing objects with presigned URLs](#).

Note

When calling UpdateInformation change type, you can always provide a subset of fields to be updated in the DetailsDocument object. However, before moving a solution from Draft to Limited, the following fields must be set properly: Title, ShortDescription, LongDescription, and LogoUrl.

However, when you are updating existing fields on the solution, you can include only the attributes that need to be changed in the DetailsDocument object of the UpdateInformation change type.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed. To check request status, use the AWS Marketplace Management Portal or call the DescribeChangeSet API.

After triggering this change type with `PromotionalMedia` assets modified for a `Public` solution, it can take up to 37 days to complete. This includes the time the AWS Marketplace Seller Operations Team needs to review, audit, and approve.

Asynchronous Errors

In addition to [common asynchronous errors](#), the following errors are specific to `UpdateInformation` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_SHORT_DESCRIPTION	The solution ShortDescription length isn't valid. Enter a ShortDescription that's between 1 and 1000 characters and try again.
INVALID_LONG_DESCRIPTION	The solution LongDescription length isn't valid. Enter a LongDescription that's between 1 and 5000 characters and try again.
INVALID_NAME	The solution Name length isn't valid. Enter a Name that's between 1 and 100 characters and try again.
INVALID_TITLE	The solution Title length isn't valid. Enter a Title that's between 1 and 255 characters and try again.
INVALID_ADDITIONAL_RESOURCES	The solution AdditionalResources[%d].Text length isn't valid. Enter a Text that's between 1 and 500 characters and try again.
INVALID_ADDITIONAL_RESOURCES	The following URLs in the provided additional resources are inaccessible: [%s]. Provide valid and accessible URLs
INVALID_ADDITIONAL_RESOURCES	A Url value is required. Specify a Url value and try again.
INVALID_ADDITIONAL_RESOURCES	Text cannot be missing. Provide a Text

Error code	Error message
INVALID_USE_CASE	The solution UseCases[%d].Description length isn't valid. Enter a Description that's between 1 and 500 characters and try again.
INVALID_USE_CASE	UseCases[%d].Group: The provided Group (%s) is invalid. Provide a valid Group.
INVALID_USE_CASE	UseCases[%d].SubGroup: The provided Group (%s) is invalid. Provide a valid SubGroup.
INVALID_USE_CASE	UseCases[%d].Item: The provided Item (%s) is invalid. Provide a valid Item.
INVALID_USE_CASE	UseCases[%d].Item: The provided use case is inactive. Provide an active use case
INVALID_USE_CASE	UseCases[%d].Group: Group cannot be missing. Provide a Group.
INVALID_USE_CASE	UseCases[%d].SubGroup: SubGroup cannot be missing. Provide a SubGroup.
INVALID_USE_CASE	UseCases[%d].Item: SubGroup cannot be missing. Provide a Item.
INVALID_MEDIA	The solution PromotionalMedia[%d].Title length isn't valid. Enter a Title that's between 1 and 100 characters and try again.
INVALID_MEDIA	The solution PromotionalMedia[%d].Description length isn't valid. Enter a Description that's between 1 and 200 characters and try again.
INVALID_MEDIA	Invalid URL: %s Provide a new URL for media stored in S3.
INVALID_MEDIA	Media location not accessible: %s Provide a new, accessible URL for media stored in one of the following locations: [S3].

Error code	Error message
INVALID_MEDIA	Image size exceeds %s. Provide an image that is under %s.
INVALID_MEDIA	Video size exceeds %s. Provide an video that is under %s.
INVALID_MEDIA	Logo size exceeds %s. Provide an logo that is under %s.
INVALID_MEDIA	Malware detected in %s. Provide media without malware.
INVALID_MEDIA	Inappropriate content: %s detected. Provide media with no inappropriate content
INVALID_MEDIA	Explicit content in %s: '{ExplicitContent}' detected. Provide media without explicit content.
INVALID_MEDIA	PromotionalMedia[%s].Url: %s is not in a supported format (%s). Use a well-formed image in a supported format: [JPEG, PNG].
INVALID_MEDIA	PromotionalMedia[%s].Url: %s is not in a supported format (%s). Use a well-formed video in a supported format: [MP4, MOV].
INVALID_MEDIA	LogoUrl at %s has invalid aspect ratio. Provide an image with 1:1 (square) or 2:1 (wide) aspect ratio.
INVALID_MEDIA	Logo: %s has invalid dimensions. Provide an image between 120x120 and 1080x1080 pixels.
INVALID_MEDIA	PromotionalMedia[%d].Type is not supported. Provide PromotionalMedia in a supported type: %s
DUPLICATE_MEDIA	Duplicate media detected: [%s] Provide media items with no duplicates.
INVALID_MEDIA	PromotionalMedia[%d].Url: Malware detected in %s. Provide media without malware.
TOO_MANY_ITEMS	Provide no more than %d %s.

Update related products

You can use the Catalog API to update the products and services included in your solution, including AWS Marketplace products and non-AWS Marketplace products.

To update related products, call the `StartChangeSet` API operation with the `UpdateRelatedProducts` change type, as shown in the following example.

Note

The `UpdateRelatedProducts` change type behaves like an HTTP PATCH operation for top-level attributes. You can provide all current AWS Marketplace product types as `AwsMarketplaceProducts` in `UpdateRelatedProducts`.

Request Syntax

```
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"
    }
  ]
}

```

Provide information for the fields to add the `UpdateRelatedProducts` change type:

- **Entity (object) (required)** – The named type of entity being updated.
 - **Identifier (string) (required)** – The solution ID.
 - **Type (string) (required)** – Must be `Solution@1.0`.
- **DetailsDocument (object) (required)** – The details of the request. Each field is optional, but you must include at least one change to update.
 - **IntegrationDescription (string)** – Description of how the products work together in this solution. Max length: 5000 characters. Must not contain leading or trailing whitespaces or control characters.
 - **AwsMarketplaceProducts (array of objects)** – AWS Marketplace products included in this solution. Max 5 items
 - **ProductId (string)** – AWS Marketplace product ID (a UUID or a string beginning with `prod-`). Max length: 50 characters.
 - **NonAwsMarketplaceProducts (array of objects)** – Products not available on AWS Marketplace. Max 5 items.
 - **Type (string) (required)** – Type of the product. Possible values: `Software`, `Consulting`, `Hardware`, `Communication`, `Professional Service`, `Managed Service`, `Value-Added Resale`, `Training`
 - **Title (string) (required)** – Title of the product. Max length: 255 characters. Must not contain leading or trailing whitespaces, linebreaks, or control characters.
 - **Description (string) (optional)** – Description of the product. Max length: 220 characters. Must not contain leading or trailing whitespaces or control characters.
 - **Url (string) (optional)** – URL to more information about the product. Max length: 2048 characters. Must be an https url.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed.

To check request status, use the AWS Marketplace Management Portal or call the `DescribeChangeSet` API.

Asynchronous Errors

In addition to [common asynchronous errors](#), the following errors are specific to `UpdateRelatedProducts` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_OPERATION_UPDATE_RELATED_PRODUCTS_SOLUTION	The operation to update related products isn't supported for public solutions. To make this change, first change the solution's visibility to limited and try again.
INVALID_AWS_MARKETPLACE_PRODUCTS	AWS Marketplace product '%s' is not available for your account. Choose a Public product or a Limited product where you have allowlist access, or contact the seller to request access.
INVALID_AWS_MARKETPLACE_PRODUCTS	The specified product type '%s' isn't supported. Provide one of these valid product types: [AmiProduct, SaaSProduct, ContainerProduct, MachineLearningProduct, ProfessionalServicesProduct, DataProduct] and try again.

Error code	Error message
INVALID_AWS_MARKETPLACE_PRODUCTS	Product '%s' not found. Provide a valid product ID in the same catalog.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	The specified Non-AWS Marketplace product URLs aren't accessible: [%s]. Verify that the URLs are valid and publicly accessible, and try again.
INVALID_INTEGRATION_DESCRIPTION	Provide a solution IntegrationDescription between 1 and 5000 characters.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	The solution NonAwsMarketplaceProduct[%d].Title length isn't valid. Enter a Description that's between 1 and 255 characters and try again.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	The solution NonAwsMarketplaceProduct[%d].Description length isn't valid. Enter a Description that's between 1 and 220 characters and try again.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	The solution NonAwsMarketplaceProduct[%d].Type length isn't valid. Enter a Type that's between 1 and 50 characters and try again.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	NonAwsMarketplaceProduct[%d].Type is not a valid non-AWS Marketplace product type. Provide one of the following non-AWS Marketplace product types: [%s]
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	A NonAwsMarketplaceProduct[%d].Title value is required. Specify a Title value and try again.
INVALID_NON_AWS_MARKETPLACE_PRODUCTS	A NonAwsMarketplaceProduct[%d].Type value is required. Specify a Type value and try again.
TOO_MANY_PRODUCTS	The maximum number of AWS Marketplace products you can link is %d. Remove existing linked products before adding new ones.

Error code	Error message
TOO_MANY_PRODUCTS	The maximum number of none AWS Marketplace products you can add is %d. Remove existing none AWS Marketplace products before adding new ones.
INVALID_AWS_MARKETPLACE_PRODUCTS	Duplicate AWS Marketplace product IDs aren't allowed. Each product ID must be unique in your solution.
INVALID_REMOVE_OPERATION	Invalid operation, you can not remove all products.

Release a solution

You can use the Catalog API to publish a Draft solution into Limited visibility in AWS Marketplace.

The ReleaseSolution change type validates that all required fields for a Limited solution are present. It then moves the solution from Draft to Limited visibility. Once a solution is in Limited visibility, you cannot move it back to Draft visibility.

To release a solution, call the StartChangeSet API operation with the ReleaseSolution change type, as shown in the following example.

Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseSolution",
      "Entity": {
        "Type": "Solution@1.0",
        "Identifier": "soln-1234abcd"
      },
      "DetailsDocument": {}
    }
  ]
}
```

```
]
}
```

Provide information for the fields to add the ReleaseSolution change type:

- Entity (object) (required) – The named type of entity being released.
 - Identifier (string) (required) – The solution ID.
 - Type (string) (required) – Must be Solution@1.0.
- DetailsDocument (object) (required) – Must be an empty object. The change type ReleaseSolution doesn't accept any details.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed. You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the DescribeChangeSet API operation.

Asynchronous Errors

In addition to [common asynchronous errors](#), the following errors are specific to ReleaseSolution actions in the AWS Marketplace Catalog API. These errors are returned when you call DescribeChangeSet after a change set is processing. For more information about using DescribeChangeSet to get the status of a change request, see [Working with change sets](#).

Error code	Error message
MISSING_REQUIRED_FIELDS	The request is missing required parameters: [%s]. Specify values for all required parameters and try again.

Error code	Error message
INCOMPATIBLE_OPERATION_RELEASE_SOLUTION	The change type can only be invoked on Solutions in Draft status. Update the Solution's visibility to Draft and try again.

Update buyer engagement options

You can use the Catalog API to configure how buyers engage with you about your solution.

Note

To use this change type, you must have active enrollment in the [APN Customer Engagements \(ACE\)](#) program. You must also link your AWS account with your AWS Partner Central account that has enrolled in ACE. For instructions, see [Linking AWS Partner Central and AWS accounts](#).

To update buyer engagement options, call the `StartChangeSet` API operation with the `UpdateBuyerEngagementOptions` change type, as shown in the following example.

Request Syntax

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



```
    }  
  ]  
}  
]  
}
```

Provide information for the fields to add the `UpdateBuyerEngagementOptions` change type:

- **Entity (object) (required)** – The named type of entity being updated.
 - **Identifier (string) (required)** – The solution ID.
 - **Type (string) (required)** – Must be `Solution@1.0`.
- **DetailsDocument (array of objects) (required)** – The buyer engagement options to enable.
 - **EngagementOption (string) (required)** – The type of engagement option. Valid values: `RequestPrivateOffer`, `RequestProductDemo`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed.

To check request status, use the AWS Marketplace Management Portal or call the `DescribeChangeSet` API.

Asynchronous Errors

In addition to [common asynchronous errors](#), the following errors are specific to `UpdateBuyerEngagementOptions` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INELIGIBLE_SELLER	ACE eligibility validation isn't complete. To proceed: 1) Link your AWS Marketplace account to your APN account, 2) Complete the ACE eligibility requirements.

Update solution visibility

You can use the Catalog API to update the visibility of a solution in AWS Marketplace.

You can set target visibilities to `Limited`, `Public`, or `Restricted`.

- **Limited** – The solution is complete and has successfully completed the `ReleaseSolution` change type. You can preview buyer website details of the solution in this visibility. On the AWS Marketplace buyer website, the solution is visible only to the owning account, the owners of each AWS Marketplace product included in the solution, and the AWS Marketplace Seller Operations team.
- **Public** – The solution is visible on AWS Marketplace. Buyers can view and engage with the solution.
- **Restricted** – The solution is no longer visible to the public on the AWS Marketplace buyer website.

Note

To move a solution to `Public` visibility, you must meet the following requirements:

- You must have a public profile on AWS Marketplace.
- The solution must have no non-`Public` products.
- The solution must have at least 2 `Public` AWS Marketplace products.
- The solution must have at least 1 AWS Marketplace product from the same AWS account.
- The solution must have the `RequestPrivateOffer` buyer engagement option enabled.
- The AWS Marketplace Seller Operations team reviews all requests to move a solution to `Public` visibility. For support, see [Getting support for AWS Marketplace](#).

When you set `TargetVisibility` to `Public`, the `UpdateVisibility` change type requires manual review from the AWS Marketplace Seller Operations team. This increases execution time. Use `UpdateVisibility` separately in its own change set.

To update your solution's visibility, call the `StartChangeSet` API operation with the `UpdateVisibility` change type, as shown in the following example.

Request Syntax

For when `TargetVisibility` is `Public` or `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"
      }
    }
  ]
}
```

For when `TargetVisibility` is `Restricted`.

```
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"
  }
}
```

Provide information for the fields to add the UpdateVisibility change type:

- Entity (object) (required) – The named type of entity being updated.
 - Identifier (string) (required) – The solution ID.
 - Type (string) (required) – Must be Solution@1.0.
- DetailsDocument (object) (required) – The details required to run the change set.
 - TargetVisibility (string) (required) – The intended new visibility of the solution. Possible values: Public, Limited, and Restricted.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the DescribeChangeSet API operation.

After triggering this change type with TargetVisibility as Public, it can take up to 37 days to complete. This includes the time the AWS Marketplace Seller Operations Team needs to review, audit, and approve.

Asynchronous Errors

The following errors are specific to `UpdateVisibility` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_VISIBILITY_TRANSITION	The solution can't be moved back to Draft state. Once released, solutions can only transition between Limited, Public, and Restricted states.
TOO_FEW_USE_CASES	At least one use case is required to make a solution public. Add a use case and try again.
TOO_FEW_PUBLIC_PRODUCTS	A minimum of 2 public AWS Marketplace products are required to make a solution public. Add the required products and try again.
INCOMPATIBLE_PRODUCTS	The solution contains one or more non-public AWS Marketplace products. To make the solution public, remove all non-public products and try again.
TOO_FEW_SELF_OWNED_PRODUCTS	The solution requires at least one AWS Marketplace product owned by the solution owner. Add an AWS Marketplace product from your account and try again.
INCOMPATIBLE_BUYER_ENGAGEMENT_OPTIONS	The <code>RequestPrivateOffer</code> buyer engagement option must be enabled to make this solution public. Enable <code>RequestPrivateOffer</code> and try again.
INCOMPATIBLE_OPERATION_UPDATE_VISIBILITY	<code>UpdateVisibility</code> isn't supported for solutions in Draft state. Publish the solution first from Draft to Limited using <code>ReleaseSolution</code> , then use <code>UpdateVisibility</code> to change to Public or Restricted.
LIMIT_EXCEEDED_TOO_MANY_PUBLIC_SOLUTIONS	This account has reached its limit of 5 public solutions. Update an existing public solution to Limited or Restricted visibility and try again.

Error code	Error message
MISSING_SELLER_PROFILE_INFORMATION	A public seller profile is required to make solutions public. Create a public profile in your seller account and try again.

Common asynchronous error codes

There are some asynchronous validation that applies across all solution change types, such as checks for url validity and unsupported character, and their error codes and error messages are listed here.

Each URL field provided must use HTTPS protocol, and conform to RFC2396 or RFC 2732 standards.

Error code	Error message
INVALID_INPUT	Invalid '%s' field. Remove unsupported characters %s.
INVALID_INPUT	Inappropriate content '%s' found in %s field. Provide %s with no inappropriate content.
UNSUPPORTED_CATALOG	Requested catalog %s is not supported by this change type.
INVALID_REMOVE_OPERATION	Invalid operation, you can not remove %s
AUDIT_ERROR	(varies based on AWS Marketplace Seller Operations team audit result)

Solution lifecycle

Solutions progress through the following visibility values:

- **Draft** - Initial visibility after creation. You can progressively add information using other change types.
- **Limited** - Solution is complete and visible to you and product owners for preview. You achieve this by using `ReleaseSolution`.

- **Public** - Solution is visible to all buyers browsing the AWS Marketplace website. This requires additional validation.
- **Restricted** - Solution is no longer visible to new buyers.

Important

Once a solution moves from Draft to Limited visibility, you cannot move it back to Draft. ReleaseSolution validates that all required fields for a Limited solution are present before the transition to Limited visibility.

Minimum requirements for a complete solution

To create a complete solution and publish it beyond Draft status, you need the following change types and attributes:

- **CreateSolution**: Name
- **UpdateInformation**: Title, ShortDescription, LongDescription, LogoUrl
- **UpdateRelatedProducts**: IntegrationDescription, at least one item across AwsMarketplaceProducts and NonAwsMarketplaceProducts
- **ReleaseSolution**

The following example shows you how to create a complete solution with Limited visibility in a single change set:

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

```

IAM permissions for solutions

AWS Marketplace Catalog API uses standard IAM permissions. See [Access control for the AWS Marketplace Catalog API](#) for more details.

You can use `AWSMarketplaceSellerProductFullAccess` to perform all operations necessary for accessing Solution via Catalog API and Partner Central 3.0 in AWS Management Console.

Sample IAM policy for solutions

The following sample IAM policy allows specific actions (`DescribeEntity`, `ListEntities`, and `StartChangeSet`) on `Solution@1.0` entity. You can further narrow it down to restrict a principal's access to certain change types on certain entity IDs, or widen it up to allow a principal to also operate on another entity type supported by Catalog API.

```

{
  "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/*"
  ]
}
]
}
```

Restricting access to specific solutions

You can restrict access to specific solutions by using the solution ID in the resource ARN:

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "aws-marketplace:DescribeEntity",
        "aws-marketplace:StartChangeSet"
      ],
      "Resource": [
        "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/soln-specific123"
      ]
    }
  ]
}
```

Restricting access to specific change types

You can restrict access to specific change types using condition keys:

```

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

```
    ],  
    "Condition": {  
      "StringEquals": {  
        "catalog:ChangeType": [  
          "UpdateInformation",  
          "UpdateRelatedProducts"  
        ]  
      }  
    }  
  }  
]  
}
```

Work with offer sets

AWS Marketplace offer sets enable sellers to package multiple private offers into unified multi-product transactable packages for buyers. This capability addresses growing customer demand for comprehensive, end-to-end solutions in complex enterprise environments.

Traditionally, buyers had to discover and procure individual products from multiple sellers, resulting in fragmented experiences, integration challenges, and longer procurement cycles. Offer sets simplify this by allowing AWS Marketplace partners to combine multiple offers together, streamlining the transaction process through coordinated private offers where customers can review and accept all components with one-time approval.

Offer sets work together with solutions to provide comprehensive multi-product offerings. Solutions provide rich marketing content including detailed descriptions, architecture diagrams, and use case documentation, while offer sets handle the transactable packaging and commercial terms for unified procurement.

Offer set entity

An offer set is a container that groups 2-7 private offers into a single transactable package. Each offer within an offer set maintains its own distinct pricing, payment terms, duration, and End User License Agreement (EULA), while the offer set provides a unified discovery and acceptance experience for buyers.

Key characteristics:

- Groups multiple private offers (minimum 2, maximum 7)

- Enables unified buyer acceptance of all offers with a single action
- Maintains flexibility with distinct terms for each offer
- Creates separate agreements for each product, allowing independent management after purchase
- Can be optionally associated with a solution for enhanced marketing content

The offer set entity type is `OfferSet@1.0`.

Entity identifiers

Offer sets use the `offerset-*` identifier format (for example, `offerset-abc123def456`).

Entity ARN format

Offer sets follow the standard AWS ARN format:

```
arn:aws:aws-marketplace:{region}:{account-id}:AWSMarketplace/OfferSet/{offerset-id}
```

Example:

```
arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/OfferSet/offerset-abc123def456
```

Getting started with offer sets

This section provides detailed information about creating and managing offer sets using AWS Marketplace Catalog API [change types](#).

Before you begin, make sure you have:

- Valid AWS Marketplace seller registration and a public profile. For more information, see [Register and create your seller profile](#).
- Access to the API and completed seller prerequisites. For more information, see [Access control for the AWS Marketplace Catalog API](#).
- Learned the basics of using the AWS Marketplace Catalog API, see [Using the AWS Marketplace Catalog API](#).

List offer sets

To list all offer sets in your account, call the `ListEntities` API operation with `EntityType` set to `OfferSet`.

Request

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

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

Response

```
{
  "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
}
```

Filter offer sets by state

You can filter offer sets by state using the `EntityTypeFilters` parameter.

Request

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

{
  "Catalog": "AWSMarketplace",
  "EntityType": "OfferSet",
  "EntityTypeFilters": {
    "OfferSetFilters": {
      "State": {
        "ValueList": [
          "Released"
        ]
      }
    }
  }
}
```

Filter offer sets by solution ID

You can filter offer sets by solution ID using the `EntityTypeFilters` parameter.

Request

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

{
  "Catalog": "AWSMarketplace",
  "EntityType": "OfferSet",
  "EntityTypeFilters": {
    "OfferSetFilters": {
      "SolutionId": {
        "ValueList": [
          "soln-abc987"
        ]
      }
    }
  }
}
```

```
}  
}
```

Filter offer sets by release date

You can filter offer sets by release date and sort the results using the `EntityTypeFilters` and `EntityTypeSort` parameters.

Request

```
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"  
    }  
  }  
}
```

Describe an offer set

To get detailed information about a specific offer set, call the `DescribeEntity` API operation.

Request

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=offerset-xyz123 HTTP/1.1
```

Response

```
{
```



```

"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"
    }
  ]
}
}

```

Create an offer set

Use the `CreateOfferSet` change type to create a new offer set entity in Draft state. This is the first step in the offer set lifecycle, establishing the foundational entity that will later be configured with offer associations and metadata before releasing to buyers.

Request Syntax

```

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"
    }
  }
]
```

Provide information for the fields to add the CreateOfferSet change type:

- Entity (object) (required) – The entity type being created.
 - Type (string) (required) – Must be OfferSet@1.0.
- DetailsDocument (object) (required) – The details of the request.
 - Name (string) (required) – The name associated with the offer set for better readability to you and your customers. Minimum length: 1 character. Maximum length: 150 characters.

Response Syntax

A change set is created for your request. The response to this request gives you the ChangeSetId and ChangeSetArn for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

After successful processing, AWS Marketplace generates an offer set in Draft state with a unique identifier prefixed with offer set-.

Update offer set information

Use the `UpdateInformation` change type to modify the details of an existing offer set, including its name and buyer notes.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the `UpdateInformation` change type:

- **Entity (object) (required)** – The entity being updated.
 - **Identifier (string) (required)** – Your offer set ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – Must be `OfferSet@1.0`.
- **DetailsDocument (object) (required)** – At least one field must be provided.
 - **Name (string) (optional)** – Display name visible to buyers. Minimum length: 1 character. Maximum length: 150 characters.
 - **BuyerNotes (string) (optional)** – Detailed information about the offer set that helps buyers understand its purpose and contents. Minimum length: 1 character. Maximum length: 1,000 characters.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Associate offers with an offer set

Use the `AssociateOffers` change type to associate multiple AWS Marketplace offers with an offer set. This creates the relationship between offers and the offer set, making the offers part of the offer set's transactable configuration.

The operation is idempotent and allows associating offers regardless of their current state, enabling complete configuration before individual offers are released. Each offer can only belong to one offer set.

Important

The `OfferSetId` on individual offers is immutable and can only be set during offer creation. Before associating offers with an offer set using `AssociateOffers`, you must first create the individual offers and specify the `OfferSetId` during offer creation. If you need to include an existing offer that doesn't have the correct `OfferSetId`, you must create a new offer with the correct `OfferSetId` specified.

For information about required IAM permissions, see [IAM permissions for offer sets](#).

Request Syntax

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

Provide information for the fields to add the AssociateOffers change type:

- Entity (object) (required) – The entity being updated.
 - Identifier (string) (required) – Your offer set ID. For more information, see [Identifier](#).
 - Type (string) (required) – Must be OfferSet@1.0.
- DetailsDocument (object) (required) – The details of the request.
 - Offers (array) (required) – List of AWS Marketplace offers to associate. Minimum: 1 item. Maximum: 7 items.
 - OfferId (string) (required) – The identifier of the AWS Marketplace offer. Minimum length: 1 character. Maximum length: 36 characters.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Asynchronous Errors

The following errors are specific to `AssociateOffers` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
TOO_MANY_OFFERS	Associate up to 7 offers to the offer set.
INCOMPATIBLE_OFFER_SET_REFERENCE	Ensure all offers were created specifically for this offer set.
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer set released.

Disassociate offers from an offer set

Use the `DisassociateOffers` change type to remove multiple AWS Marketplace offers from an offer set. This idempotent operation removes the association relationship while preserving the

offer's `OfferSetId` attribute. Disassociated offers become hidden from buyer discovery until re-associated with the same offer set. They cannot be associated with a different offer set.

Request Syntax

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

Provide information for the fields to add the `DisassociateOffers` change type:

- **Entity (object) (required)** – The entity being updated.
 - **Identifier (string) (required)** – Your offer set ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – Must be `OfferSet@1.0`.
- **DetailsDocument (object) (required)** – The details of the request.
 - **Offers (array) (required)** – List of offers to disassociate. Minimum: 1 item. Maximum: 7 items.
 - **OfferId (string) (required)** – The identifier of the AWS Marketplace offer. Minimum length: 1 character. Maximum length: 36 characters.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Asynchronous Errors

The following errors are specific to `DisassociateOffers` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer set released.

Associate a solution with an offer set

Use the `AssociateSolution` change type to associate a single AWS Marketplace solution with an offer set. This creates a relationship that allows buyers to discover the offer set's connection to the solution and access the solution's rich marketing content such as detailed descriptions, architecture diagrams, and use case documentation.

Note

This is a loosely coupled association. AWS Marketplace doesn't enforce consistency between the solution and offer set. Solutions can contain no AWS Marketplace products, different

products than those in the offer set, or overlapping products. You have complete flexibility to associate any solution you own for marketing and discovery purposes.

For information about required IAM permissions, see [IAM permissions for offer sets](#).

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the AssociateSolution change type:

- Entity (object) (required) – The entity being updated.
 - Identifier (string) (required) – Your offer set ID. For more information, see [Identifier](#).
 - Type (string) (required) – Must be OfferSet@1.0.
- DetailsDocument (object) (required) – The details of the request.
 - SolutionId (string) (required) – The identifier of the AWS Marketplace solution to associate. Minimum length: 1 character. Maximum length: 50 characters. Must match pattern `^soln-[A-Za-z0-9]+$`.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Asynchronous Errors

The following errors are specific to `AssociateSolution` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INCOMPATIBLE_SOLUTION_ASSOCIATION	Disassociate the existing solution before associating a new solution.
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer set released.

Disassociate a solution from an offer set

Use the `DisassociateSolution` change type to remove the association between a solution and an offer set. This removes the solution's marketing content from the offer set while maintaining the offer set's transactable functionality.

Request Syntax

```
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"
      }
    }
  ]
}
```

Provide information for the fields to add the `DisassociateSolution` change type:

- **Entity (object) (required)** – The entity being updated.
 - **Identifier (string) (required)** – Your offer set ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – Must be `OfferSet@1.0`.
- **DetailsDocument (object) (required)** – The details of the request.
 - **SolutionId (string) (required)** – The identifier of the solution to disassociate. Minimum length: 1 character. Maximum length: 50 characters.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Asynchronous Errors

The following errors are specific to `DisassociateSolution` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer set released.

Release an offer set

Use the `ReleaseOfferSet` change type to make an offer set available to buyers. Once released, the offer set transitions from Draft to Released state, becomes discoverable through AWS Marketplace, and associated offers can only be found as part of the unified package.

Released offer sets generate notifications at two levels: individual offers and the offer set itself. All individual offer notifications include the `OfferSetId` in existing notification channels ([email notifications](#) and [Amazon EventBridge events](#)). Offer set-level notifications are sent for key events such as when the offer set is released.

Request Syntax

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "ReleaseOfferSet",
```

```
    "Entity": {
      "Type": "OfferSet@1.0",
      "Identifier": "offerset-abc123"
    },
    "DetailsDocument": {}
  }
]
```

Provide information for the fields to add the `ReleaseOfferSet` change type:

- **Entity (object) (required)** – The entity being updated.
 - **Identifier (string) (required)** – Your offer set ID. For more information, see [Identifier](#).
 - **Type (string) (required)** – Must be `OfferSet@1.0`.
- **DetailsDocument (object) (required)** – Empty object for this operation.

Response Syntax

A change set is created for your request. The response to this request gives you the `ChangeSetId` and `ChangeSetArn` for the change set and looks like the following.

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

The change request is added to a queue and processed. This includes validating information to ensure that it meets the AWS Marketplace guidelines. The validation process can take a few minutes.

You can check the status of the request through the AWS Marketplace Management Portal, or directly through Catalog API using the [DescribeChangeSet](#) API operation.

Asynchronous Errors

The following errors are specific to `ReleaseOfferSet` actions in the AWS Marketplace Catalog API. These errors are returned when you call `DescribeChangeSet` after a change set is processing. For more information about using `DescribeChangeSet` to get the status of a change request, see [Working with change sets](#).

Error code	Error message
MISSING_BUYER_NOTES	Provide BuyerNotes before releasing the offer set.
MISSING_OFFERS	Associate at least two offers to the offer set before releasing it.
TOO_MANY_OFFERS_PER_PRODUCT	Associate only one offer per product to the offer set.
INCONSISTENT_OFFER_CURRENCY_CODE	Ensure all associated offers have matching CurrencyCode.
INCONSISTENT_OFFER_AVAILABILITY_END_DATE	Ensure all associated offers have matching AvailabilityEndDate.
INCONSISTENT_OFFER_TARGETING_RULE	Ensure all associated offers have matching TargetingRule.
INCOMPATIBLE_OFFER_TARGETING_RULE	Only buyer-targeted offers can be associated to an offer set.
EXPIRED_OFFERS	Disassociate expired offers from the offer set or extend their AvailabilityEndDates.
DRAFT_OFFERS	Release all associated offers before releasing the offer set.
INCOMPATIBLE_SOLUTION_STATE	Provide a solution in limited or public state.
INVALID_UPDATE_REQUEST	The requested change can't be performed after the offer set released.

Offer set lifecycle

The offer set lifecycle consists of the following stages:

Creation phase

Create the offer set entity in Draft state using the `CreateOfferSet` change type. The offer set receives a unique identifier and begins in Draft state.

Configuration phase

During configuration, you:

- Add or update metadata using `UpdateInformation` (name, buyer notes)
- Create individual private offers, specifying the `OfferSetId` during offer creation
- Associate offers with the offer set using `AssociateOffers`
- Optionally associate a solution using `AssociateSolution` for enhanced marketing content

Important

The `OfferSetId` on individual offers is immutable and can only be set during offer creation. Before associating offers with an offer set using `AssociateOffers`, you must first create the individual offers and specify the `OfferSetId` during offer creation. If you need to include an existing offer that doesn't have the correct `OfferSetId`, you must create a new offer with the correct `OfferSetId` specified.

Validation phase

Before releasing an offer set, ensure:

- The offer set contains between 2 and 7 offers
- All associated offers must be:
 - In Released state
 - Active
 - Using the same currency
 - Targeting the same buyer AWS account ID(s)
 - Having identical expiration dates

Release phase

When ready, use the `ReleaseOfferSet` change type to transition the offer set from Draft to Released state. This makes the offer set discoverable to buyers through AWS Marketplace.

Post-release management

Expiration management:

- The effective expiration of an offer set is calculated as the earliest expiration date among all associated offers
- You can modify individual offer expiration dates using existing offer management capabilities (`UpdateAvailability` change type)

Offer set modifications:

When buyers request changes to a released offer set, use the recreation workflow:

1. Create a new offer set entity
2. For offers that don't require changes, clone existing offers through the AWS Marketplace Management Portal
3. For offers requiring modifications, create new offers with the requested changes, specifying the new `OfferSetId`
4. Associate all offers (cloned and new) with the new offer set using `AssociateOffers`
5. Release the new offer set using `ReleaseOfferSet`
6. Expire the original offer set by setting the availability end date of associated offers

IAM permissions for offer sets

To work with offer sets using the AWS Marketplace Catalog API, you need specific IAM permissions. This section describes the required permissions for offer set operations and cross-entity authorization requirements.

Required IAM actions

The following IAM actions are required for offer set operations:

- `aws-marketplace:StartChangeSet` – Required for all offer set change types including create, update, associate, and release operations
- `aws-marketplace:DescribeChangeSet` – Required to check the status and results of change set executions
- `aws-marketplace:ListEntities` – Required to list offer sets in your account

- `aws-marketplace:DescribeEntity` – Required to retrieve detailed information about an offer set

Resource permissions

Use the following ARN patterns to grant permissions on specific resources:

- **Offer sets** – `arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/OfferSet/*`
- **Individual offers** – `arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Offer/*`
- **Solutions** – `arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/Solution/*`

For more restrictive permissions, replace the wildcard (*) with specific entity identifiers.

Cross-entity authorization

Some offer set operations require permissions on both the offer set and the entities being associated:

AssociateOffers

- `aws-marketplace:StartChangeSet` permission on the offer set entity
- `aws-marketplace:StartChangeSet` permission with `AssociateWithOfferSet` change type on each individual offer entity being associated

AssociateSolution

- `aws-marketplace:StartChangeSet` permission on the offer set entity
- `aws-marketplace:StartChangeSet` permission with `AssociateWithOfferSet` change type on the solution entity being associated

Note

Having only `DescribeEntity` (read-only) permission on offers or solutions is not sufficient for association operations. You must have `StartChangeSet` permission with the `AssociateWithOfferSet` change type on the entities being associated with the offer set.

Example IAM policies

Basic offer set management

This policy provides permissions for basic offer set operations without cross-entity associations:

```
{
  "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/*"
      ]
    }
  ]
}
```

Cross-entity permissions for AssociateOffers

This policy demonstrates the specific permissions and conditions required for `AssociateOffers`:

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

```

For more restrictive permissions, replace the wildcard (*) with specific entity identifiers.

Cross-entity permissions for AssociateSolution

This policy demonstrates the specific permissions and conditions required for AssociateSolution:

```

{
    "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"]
        }
    }
}
]
}

```

For more restrictive permissions, replace the wildcard (*) with specific entity identifiers.

Build interfaces with catalog data

You can use the AWS Marketplace Discovery API to programmatically access the catalog and display product and pricing information on your own platforms. This section describes how to build custom interfaces such as storefronts, partner portals, and integrated sales experiences for your customers to navigate, using the Discovery API.

Architecture overview

A typical integration follows this pattern:

1. **Browse experience** — Call `SearchListings` to retrieve listing summaries based on search text and filters, and `SearchFacets` to populate category navigation and filter options with counts.
2. **Product detail pages** — Call `GetListing` or `GetProduct` to display full product details, media, and reviews.

3. **Pricing display** — Call `ListPurchaseOptions`, `GetOffer`, and `GetOfferTerms` to show pricing options.

Build a browse experience

Use `SearchListings` to retrieve listing summaries based on search text and filters, and `SearchFacets` to build dynamic filter navigation. Facets include categories, fulfillment types, pricing models, pricing units, publishers, and deployment status. Each facet value includes a count of matching listings, which you can display alongside filter options.

```
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']}")
```

Build product detail pages

Use `GetListing` or `GetProduct` to display full product details, media, and reviews. Use `GetListing` when you need the listing overview including badges, categories, and pricing summaries. Use `GetProduct` when you need detailed product information such as descriptions, highlights, and media. Call `ListFulfillmentOptions` to retrieve deployment options for a specific product.

```
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
    }
```

Display pricing information

Use `ListPurchaseOptions` to find available offers for a product, then `GetOffer` and `GetOfferTerms` to retrieve detailed pricing. Display pricing models, rate cards, and term details to help your customers evaluate options.

```
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
```

Best practices

- **Cache responses** — Product and pricing data changes infrequently. Cache API responses for 15–60 minutes to reduce API calls and improve page load times.
- **Handle pagination** — Use `nextToken` to retrieve all results from paginated operations. Pagination tokens expire after 24 hours.
- **Implement retry logic** — Use exponential backoff for `ThrottlingException` (HTTP 429) responses.
- **Display attribution** — When displaying AWS Marketplace data on your platform, follow the [AWS Trademark Guidelines](#).
- **Minimize API calls** — Use `SearchListings` for summary data and only call detail APIs (`GetProduct`, `GetOffer`, `GetOfferTerms`) when a user navigates to a specific product.

Note

The Discovery API supports all product types on AWS Marketplace, including SaaS, AI agents and tools, AMI, containers, and machine learning models. Your interface can display the full breadth of the catalog.

Using the AWS Marketplace API as a buyer

The following sections provide information on how to use the API as a buyer.

Topics

- [Discover products and pricing](#)
- [Work with the Agreement API as a buyer](#)
- [Work with the private marketplace using the AWS Marketplace API](#)

Discover products and pricing

You can use the AWS Marketplace Discovery API to programmatically search the catalog, retrieve detailed product information, and access pricing details including public and private offers. This section describes common workflows for discovering products and evaluating pricing.

Browse the catalog

Use `SearchFacets` to retrieve available filter values and build a browsing experience. Facets include categories, pricing models, fulfillment types, publishers, and deployment status. Each facet value includes a count of matching listings.

```
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)")
```

Search for listings

Use `SearchListings` to find listings by keyword and filters. You can combine text search with facet-based filters, and sort results by relevance or customer rating.

```
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']}")

```

Get listing details

Use `GetListing` to retrieve a listing's full details, including descriptions, categories, badges, pricing model summaries, reviews, and associated products and offers.

```

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']}")

```

Get product details and fulfillment options

Use `GetProduct` to get detailed product information including descriptions, highlights, categories, and seller engagements. Use `ListFulfillmentOptions` to see how the product can be deployed (AMI, SaaS, Container, Helm, etc.).

```

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']}")

```

Find available offers for a product

Use `ListPurchaseOptions` to find all available purchase options (offers and offer sets) for a product. You can filter by product ID, seller, purchase option type, visibility scope, and availability status.

```
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']}")
```

Get detailed pricing terms

Use `GetOfferTerms` to retrieve the full commercial structure of an offer. Each offer contains one or more terms — pricing terms (usage-based, fixed upfront, configurable upfront, BYOL, free trial), legal terms, payment schedules, validity terms, support terms, and renewal terms.

```
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']}")
```

Get offer set details

When a purchase option returned by `ListPurchaseOptions` has a `purchaseOptionType` of `OFFERSET`, it represents a grouped collection of private offers for a multi-product solution.

Use `GetOfferSet` to retrieve the details of the offer set, including the associated products and offers, seller of record, availability dates, and buyer notes. You can then call `GetOffer` and `GetOfferTerms` for each individual offer within the set.

```
# 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']}")
```

Access private offer pricing

To access private offer pricing, the calling IAM principal must have the appropriate permissions. Private offers include negotiated pricing, custom terms, and organization-specific offers. Use the `VISIBILITY_SCOPE` filter with value `PRIVATE` in `ListPurchaseOptions` to find private offers available to your account.

```
private_options = client.list_purchase_options(
    filters=[
        {'filterType': 'PRODUCT_ID', 'filterValues': ['prod-gitpod-enterprise']},
        {'filterType': 'VISIBILITY_SCOPE', 'filterValues': ['PRIVATE']}
    ]
)
```

>

Integrate with procurement tools

The Discovery API is designed to integrate with enterprise procurement platforms. Common integration patterns include:

- Integrating the AWS Marketplace catalog into your procurement tool's product catalog
- Displaying product details and pricing within purchase request workflows
- Enabling comparison shopping across AWS Marketplace offers

Note

The Discovery API provides the product and pricing data you need to build integrated procurement experiences. Combine it with your existing procurement workflows to streamline purchasing decisions.

Work with the Agreement API as a buyer

An agreement is a document that binds two parties, including the proposer and the acceptor (commonly, the buyer) and defines the terms and conditions applicable between them.

Buyer creates an `AgreementRequest`, which generates a quote. This quote includes all relevant information, such as the estimated charges that will be incurred over the lifetime of the agreement, which is critical for the buyer's purchase decision. If the buyer is satisfied with the quote, they can accept the `AgreementRequest` before it expires. The Agreement is created upon acceptance of the `AgreementRequest`. During this process, the buyer admin or buyer persona can associate purchase order to charges. Finally, based on the terms and conditions of the active agreement, the buyer is invoiced and granted a License to use the product.

Prerequisites: Discover products and offers

Before creating an agreement, use the AWS Marketplace Discovery API to discover the product and obtain the `agreementProposalId`, `pricingModel`, and term details needed to construct a `CreateAgreementRequest`.

Step	Discovery API action	Output	Used for
1	ListPurchaseOptions	offerId	Finding available offers for a product
2	GetOffer	agreementProposalId , pricingModel	Constructing agreement ProposalIdentifier
3	GetOfferTerms	Term IDs and pricing details	Constructing requested Terms

For detailed information about discovering products and pricing, see [Discover products and pricing](#).

An acceptor can perform the following tasks using this API:

Task	Description	Action(s)	Intent
Generate a quote	Acceptor can generate quote that includes all relevant information, such as the Charges that will be incurred over the lifetime of the agreement , which is critical for the buyer's purchase decision.	CreateAgreementRequest	NEW
Accept an offer	The Acceptor can accept terms proposed by the proposer that creates a new agreement. This acceptance can involve passing parameters for certain terms, such as selecting the quantity or duration, adding purchase order etc. Acceptor can also create a new agreement where the product usage begins at a future date. The agreement sign date will be when the offer is accepted and when the agreement is created. The Agreement start date is future date when the	CreateAgreementRequest , AcceptAgreementRequest	NEW

Task	Description	Action(s)	Intent
	product usage begins. This is the date when license/entitlement gets activated. To retrieve the latest status of your usage entitlement, refer to the GetAgreementEntitlements API.		
Replace an existing agreement	The Acceptor can perform a mid-term upgrade on their agreement to either switch to more favorable terms or switch seller for their agreements. This action terminates the existing Agreement passed as input and creates net new Agreement. This action is logically equivalent to a CANCEL followed by a NEW agreement, but ensures entitlement continuity for the Acceptor and at no point is the Acceptor left without entitlements.	CreateAgreementRequest , AcceptAgreementRequest	REPLACE
Amend an existing agreement	The acceptor is only permitted to modify the configuration for the accepted terms. For example, they can enable or disable auto-renewal, or modify the purchased quantity, as long as the price change post-modification does not result in refunds. We will only support amending charge if agreement has not started. Note: Proposer is allowed to modify prices for pay-as-you-go pricing term. Any increase in pay as you go prices takes 90 days to go into effect after the buyer is notified about the price increase. Any decrease in pay as you go prices takes effect immediately.	CreateAgreementRequest , AcceptAgreementRequest	AMEND

Task	Description	Action(s)	Intent
Enable or disable auto-renewal	The Acceptor can turn ON/Off the auto renew flag on their agreement if the seller has enabled renewal offer terms. In case its enabled, the renew agreement will be created by the Agreement service on the date of expiry of the original agreement using latest offer revision. The start date of this Agreement created will be same as the end date of the original agreement .	CreateAgreementRequest , AcceptAgreementRequest	AMEND
Cancel an agreement	Acceptor can cancel usage agreement . For everything else, buyer must reach out to seller to initiate cancellation. When you cancel your agreement, your license and entitlement gets deactivated.	CancelAgreement	N/A
Search agreements	Acceptor can perform search across all agreements that they participated in as acceptor in AWS Marketplace. The search returns a list of agreements with basic agreement information.	SearchAgreements	N/A
Describe an agreement	Acceptor can view details about an agreement, such as the proposer, acceptor, start date, and end date.	DescribeAgreement	N/A
Get agreement terms	Acceptor can obtain details about the terms in an agreement that they participated in as acceptor.	GetAgreementTerms	N/A

Task	Description	Action(s)	Intent
Get agreement entitlements	Acceptor can obtain agreement level view of status and details of entitlements tied to their agreement - for example, whether it's in the process of being granted or if it has been rejected (and if so, for what reasons), what's the entitlements being granted to customers.	GetAgreementEntitlements	N/A
Get a registration token	Registration token is a short-lived token required by acceptors to setup account with proposers. This token is used for both externally metered and entitled dimension types. The token is only valid for 30 minutes after creation and currently only applicable for SaaS purchase agreements.	GetAgreementEntitlements	N/A
List agreement charges	Acceptor can view charges and Purchase Order details associated with them for their Agreement.	ListAgreementCharges	N/A
Update purchase orders	Acceptor can add a purchase order number after subscribing to a product. Once a purchase order is associated with a charge, the invoice generated for that charge will include the purchase order number.	UpdatePurchaseOrders	N/A
List cancellation requests	Acceptor can list all cancellation requests for agreements they participate in. The list can be filtered by agreement, status, and other criteria.	ListAgreementCancellationRequests	N/A

Task	Description	Action(s)	Intent
Get cancellation request details	Accepter can retrieve detailed information about a specific cancellation request initiated by the seller (proposer), including status, timestamps, and reason codes.	GetAgreementCancellationRequest	N/A
Accept a cancellation request	Accepter can approve a cancellation request initiated by the seller (proposer) for an active agreement. After approval, the agreement cancellation workflow executes asynchronously and the agreement status changes to cancelled. Note: Users also need <code>CancelAgreement</code> permission because approving the cancellation request leads to agreement cancellation.	AcceptAgreementCancellationRequest, CancelAgreement	N/A
Reject a cancellation request	Accepter can reject a cancellation request initiated by the seller (proposer). After rejection, the agreement remains active and the cancellation request enters a terminal state. The seller can create a new cancellation request if needed.	RejectAgreementCancellationRequest	N/A
List payment requests	Accepter can list all payment requests for agreements they participate in. The list can be filtered by agreement, status, and other criteria.	ListAgreementPaymentRequests	N/A

Task	Description	Action(s)	Intent
Get payment request details	Acceptor can retrieve detailed information about a specific payment request initiated by the seller (proposer), including status, timestamps, and associated charges.	GetAgreementPaymentRequest	N/A
Accept a payment request	Acceptor can approve a payment request initiated by the seller (proposer) for an active agreement.	AcceptAgreementPaymentRequest	N/A
Reject a payment request	Acceptor can reject a payment request initiated by the seller (proposer). After rejection, the payment request enters a terminal state. The seller can create a new payment request if needed.	RejectAgreementPaymentRequest	N/A

Generate a quote

Use `CreateAgreementRequest` to generate a quote. You must provide:

- `agreementProposalIdentifier` — the `agreementProposalId` from the [GetOffer](#) response.
- `requestedTerms` — constructed from the terms in the [GetOfferTerms](#) response. Each term requires its `id`, and some require additional configuration. See [Constructing requestedTerms](#).

```
response = client.create_agreement_request(
    agreementProposalIdentifier='at-edhtjnbilupjv3xqbphtom77y',
    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']}")

```

Constructing requestedTerms

To construct requestedTerms, you need to determine two things:

1. **Which terms to include** — determined by the pricing model of the offer. See [Required terms by pricing model](#).
2. **Which terms need configuration** — most terms only need the id, but three term types require additional buyer-supplied configuration. See [Term configurations](#).

Required terms by pricing model

The pricingModel.[PricingModel](#) field in the [GetOffer](#) response determines which terms must be included in your CreateAgreementRequest.

The following terms must be included in **every** agreement when present in the offer: LegalTerm, SupportTerm, ValidityTerm, and RenewalTerm.

Note

FreeTrialPricingTerm can only be accepted **once** per product — CreateAgreementRequest will return an error if the buyer has already used a free trial.

CONTRACT

All terms returned by [GetOfferTerms](#) are required in a **single** CreateAgreementRequest.

USAGE

Agreement creation depends on the terms present in the offer:

1. **First agreement** — Accept UsageBasedPricingTerm and/or RecurringPaymentTerm (depending on what the offer includes) along with the mandatory terms. This grants product usage entitlements.
2. **Subsequent agreement(s)** (optional) — If the offer includes ConfigurableUpfrontPricingTerm, FixedUpfrontPricingTerm, or PaymentScheduleTerm, you can optionally create a separate agreement to purchase annual to get discounted pricing.

Note

If you create both agreements and later want to cancel, cancel the subsequent agreement (with ConfigurableUpfrontPricingTerm) first, then cancel the first agreement (with UsageBasedPricingTerm and/or RecurringPaymentTerm).

BYOL

Accept ByolPricingTerm along with the mandatory terms.

Term configurations

Most terms require only the id from [GetOfferTerms](#). The following term types require a buyer-supplied configuration in addition to the id:

Term type	Configuration	What the buyer provides
ConfigurableUpfrontPricingTerm	ConfigurableUpfrontPricingTermConfiguration	Agreement duration and number of units per dimension
RenewalTerm	RenewalTermConfiguration	Whether to auto-renew when the agreement expires

Term type	Configuration	What the buyer provides
VariablePaymentTerm	VariablePaymentTermConfiguration	How payment requests from the seller are approved

ConfigurableUpfrontPricingTermConfiguration

Use the example below to locate the values needed for ConfigurableUpfrontPricingTermConfiguration.

Example GetOfferTerms response:

```
{
  "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	Value / source
ConfigurableUpfrontPricingTermConfiguration.selectorValue	offerTerms[].configurableUpfrontPricingTerm.rateCards[].selector.value
Dimension.dimensionKey	offerTerms[].configurableUpfrontPricingTerm.rateCards[].rateCard[].dimensionKey
Dimension.dimensionValue	Buyer-supplied integer (number of units to purchase for that dimension)

 **Note**

The constraints field in the rate card governs how you may craft the ConfigurableUpfrontPricingTermConfiguration — for example, whether you may select multiple dimensions or only one. See the [Constraints](#) data type for details.

Example RequestedTerm payload for ConfigurableUpfrontPricingTerm:

```
{
  "id": "term-43ab7a2445f89dfcb7e1a6a81c92fe08c124125ffad126bea5e5b7915c065166",
  "configuration": {
    "configurableUpfrontPricingTermConfiguration": {
      "selectorValue": "P12M",
      "dimensions": [
        {
          "dimensionKey": "AdminUsers",
          "dimensionValue": 5
        }
      ]
    }
  }
}
```

```
    },
    {
      "dimensionKey": "ReadOnlyUsers",
      "dimensionValue": 10
    }
  ]
}
}
```

RenewalTermConfiguration

Field	Value / source
enableAutoRenew	Buyer-supplied boolean (true or false) — required

Example RequestedTerm payload for RenewalTerm:

```
{
  "id": "term-ffcc0100ce0c5468426dc07d0b05cff09ab8447640108d9c95ce1591f0949738",
  "configuration": {
    "renewalTermConfiguration": {
      "enableAutoRenew": false
    }
  }
}
```

VariablePaymentTermConfiguration

Field	Value / source
RequestedTerm.configuration.variablePaymentTermConfiguration.paymentRequestApprovalStrategy	Buyer-supplied: AUTO_APPROVE_ON_EXPIRATION or WAIT_FOR_APPROVAL — required

Field	Value / source
RequestedTerm.configuration.variablePaymentTermConfiguration.expirationDuration	Buyer-supplied duration (for example, P10D); required only when strategy is AUTO_APPROVE_ON_EXPIRATION

Example RequestedTerm payload for VariablePaymentTerm:

```
{
  "id": "term-9ab34376c51cd5b4dc995aa0f410b657d58bd577b3f09b34d38afb6964aaaf12",
  "configuration": {
    "variablePaymentTermConfiguration": {
      "paymentRequestApprovalStrategy": "AUTO_APPROVE_ON_EXPIRATION",
      "expirationDuration": "P10D"
    }
  }
}
```

Accept an offer

First, generate a quote as described in [Generate a quote](#). Then, use `AcceptAgreementRequest` to accept the quote and create the agreement. You must call `AcceptAgreementRequest` before the agreement request expires. You can optionally associate a purchase order number during acceptance.

```
# 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']}")
```


Replace an existing agreement

Construct a `CreateAgreementRequest` as described in [Generate a quote](#), but set `intent` to `REPLACE` and provide `sourceAgreementIdentifier` with the agreement to replace. Then call `AcceptAgreementRequest` to complete the replacement.

```
response = client.create_agreement_request(
    agreementProposalIdentifier='at-edhtjnbilupjv3xqbptom77y',
    intent='REPLACE',
    sourceAgreementIdentifier='agmt-existing123'
)

print(f"Agreement Request ID: {response['agreementRequestId']}")
```

Amend an existing agreement

To modify term configuration on an existing agreement, set `intent` to `AMEND` and provide `sourceAgreementIdentifier`. Unlike a new agreement, no `agreementProposalIdentifier` is needed. Only include the terms whose configuration you want to change. Then call `AcceptAgreementRequest` to apply the amendment.

```
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']}")
```

Enable or disable auto-renewal

Use `CreateAgreementRequest` with the `AMEND` intent to toggle auto-renewal.

```
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']}")
```

Cancel an agreement

Use `CancelAgreement` to cancel a usage agreement. For other agreement types, the buyer must reach out to the seller to initiate cancellation.

```
client.cancel_agreement(
    agreementId='agmt-abc123'
)
```

Search agreements

Use `SearchAgreements` to search across all agreements that you participate in as an acceptor. The search returns a list of agreements with basic agreement information.

```
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']}")
```

Describe an agreement

Use `DescribeAgreement` to view details about an agreement, such as the proposer, acceptor, start date, and end date.

```
response = client.describe_agreement(
    agreementId='agmt-abc123'
)

print(f"Status: {response['status']}")
print(f"Start: {response['startTime']}")
print(f"End: {response['endTime']}")
```

Get agreement terms

Use `GetAgreementTerms` to obtain details about the terms in an agreement that you participate in as an acceptor.

```
response = client.get_agreement_terms(
    agreementId='agmt-abc123'
)

for term in response['acceptedTerms']:
    print(f"  Term: {term}")
```

Get agreement entitlements

Use `GetAgreementEntitlements` to obtain an agreement-level view of the status and details of entitlements tied to your agreement.

```
response = client.get_agreement_entitlements(
    agreementId='agmt-abc123'
)

for ent in response['agreementEntitlements']:
    print(f"  {ent['resource']}: {ent['status']}")
```

Get a registration token

Use `GetAgreementEntitlements` to obtain a registration token. This short-lived token (valid for 30 minutes) is required by acceptors to set up an account with proposers. Currently only applicable for SaaS purchase agreements.

```
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
```

List agreement charges

Use `ListAgreementCharges` to view charges and purchase order details associated with your agreement.

```
response = client.list_agreement_charges(
    agreementId='agmt-abc123'
)

for charge in response['items']:
    print(f"  {charge['id']} (rev {charge['revision']}): ${charge['amount']}")
    print(f"  PO: {charge.get('purchaseOrderReference', 'N/A')}")
```

Update purchase orders

Use `UpdatePurchaseOrders` to add a purchase order number after subscribing to a product. Use the `chargeId` and `chargeRevision` from the [List agreement charges](#) response. Once a purchase order is associated with a charge, the invoice generated for that charge includes the purchase order number.

```
client.update_purchase_orders(
    agreementId='agmt-abc123',
    purchaseOrders=[
        {
            'chargeId': 'charge-xyz',
```

```
        'chargeRevision': 1,  
        'purchaseOrderReference': 'PO-2024-001'  
    }  
]  
)
```

List cancellation requests

Use `ListAgreementCancellationRequests` to list all cancellation requests for agreements you participate in. You can filter by agreement, status, and other criteria.

```
response = client.list_agreement_cancellation_requests(  
    partyType='Acceptor',  
    agreementId='agmt-abc123'  
)  
  
for req in response['agreementCancellationRequests']:  
    print(f" {req['agreementCancellationRequestId']}: {req['status']}")
```

Get cancellation request details

Use `GetAgreementCancellationRequest` to retrieve detailed information about a specific cancellation request, including status, timestamps, and reason codes.

```
response = client.get_agreement_cancellation_request(  
    agreementId='agmt-abc123',  
    agreementCancellationRequestId='acr-abc123'  
)  
  
print(f"Status: {response['status']}")  
print(f"Reason: {response['reasonCode']}")
```

Accept a cancellation request

Use `AcceptAgreementCancellationRequest` to approve a cancellation request initiated by the seller. After approval, the agreement cancellation workflow executes asynchronously. You also need `CancelAgreement` permission because approving the cancellation request leads to agreement cancellation.

```
client.accept_agreement_cancellation_request(  

```

```
    agreementId='agmt-abc123',  
    agreementCancellationRequestId='acr-abc123'  
)
```

Reject a cancellation request

Use `RejectAgreementCancellationRequest` to reject a cancellation request initiated by the seller. After rejection, the agreement remains active and the seller can create a new cancellation request if needed.

```
client.reject_agreement_cancellation_request(  
    agreementId='agmt-abc123',  
    agreementCancellationRequestId='acr-abc123'  
)
```

List payment requests

Use `ListAgreementPaymentRequests` to list all payment requests for agreements you participate in. You can filter by agreement, status, and other criteria.

```
response = client.list_agreement_payment_requests(  
    partyType='Acceptor',  
    agreementId='agmt-abc123'  
)  
  
for req in response['agreementPaymentRequests']:  
    print(f" {req['agreementPaymentRequestId']}: {req['status']}")
```

Get payment request details

Use `GetAgreementPaymentRequest` to retrieve detailed information about a specific payment request, including status, timestamps, and associated charges.

```
response = client.get_agreement_payment_request(  
    paymentRequestId='apr-abc123',  
    agreementId='agmt-abc123'  
)  
  
print(f"Status: {response['status']}")  
print(f"Amount: ${response['amount']}")
```

Accept a payment request

Use `AcceptAgreementPaymentRequest` to approve a payment request initiated by the seller for an active agreement.

```
client.accept_agreement_payment_request(  
    paymentRequestId='apr-abc123',  
    agreementId='agmt-abc123'  
)
```

Reject a payment request

Use `RejectAgreementPaymentRequest` to reject a payment request initiated by the seller. After rejection, the seller can create a new payment request if needed.

```
client.reject_agreement_payment_request(  
    paymentRequestId='apr-abc123',  
    agreementId='agmt-abc123'  
)
```

Work with the private marketplace using the AWS Marketplace API

You can use the AWS Marketplace Catalog API to manage a *private marketplace* for your AWS account or [organization](#).

All change types can be called only from the organization's management account or by a member account that is a delegated administrator for private marketplace. If you're a current private marketplace customer without the AWS Organizations integration for private marketplace, you can create and manage a private marketplace from any account in your organization that has the `AWSPprivateMarketplaceAdminFullAccess` IAM policy.

For more information about private marketplaces, see [Private marketplaces](#) in the *AWS Marketplace Buyer Guide*.

The following table details a set of tasks to manage private marketplaces and the change types that apply to each task.

Task	Action	Change types
the section called “Creating a private marketplace”	StartChangeSet	CreateExperience CreateProcurementPolicy
the section called “Changing the branding of a private marketplace experience”	StartChangeSet	CreateBrandingSettings UpdateBrandingSettings
the section called “Enabling or disabling a private marketplace experience”	StartChangeSet	UpdateExperience
the section called “Enabling or disabling user requests”	StartChangeSet	UpdateProcurementPolicy
the section called “Getting a list of products in a private marketplace experience”	DescribeEntity	Not applicable
the section called “Adding or removing products from a private marketplace”	StartChangeSet	AllowProductProcurement DenyProductProcurement
the section called “Finding products”	Not applicable	Not applicable
the section called “Working with private marketplaces for AWS Organizations”	Not applicable	Not applicable
the section called “Associating principals to experiences”	StartChangeSet	AssociateAudience

Task	Action	Change types
		DisassociateAudience
Archiving and reactivating a private marketplace experience	StartChangeSet	RestrictExperience ReviveExperience

Creating a private marketplace

A private marketplace for an AWS account can be thought of as a list of products that users are allowed to procure in that account, and branding for the marketplace. In an organization with multiple accounts, you can use the grouping from AWS Organizations called [organizational unit \(OU\)](#) to associate to an experience. For example, you could have one set of products that all accounts in the organization are allowed to procure, or you could have a different list of products for each OU in the organization. You can also have a different list of products for individual accounts in the organization. Each list of approved products and branding is called a *procurement experience*.

In the AWS Marketplace Catalog API, four entities represent an experience:

- Experience entity – This entity is at the highest level of the experience and contains two child entities.
- ProcurementPolicy entity – This entity represents the products that have been allowed and denied in your private marketplace.
- BrandingSettings entity – You can also create a BrandingSettings entity to define how your private marketplace looks to your users.
- Audience entity – You must also associate one or more Audience entities, which define the set of AWS accounts, OUs, or organization that the experience applies to.

The steps to create a procurement experience are as follows:

1. Create the Experience entity.
2. Create a ProcurementPolicy entity to store the list of products that are allowed or denied for the experience.

3. (Optional) Create a `BrandingSettings` entity to customize the look of your marketplace experience.
4. Associate principals with your experience. A principal can be an AWS account, OU, or the organization.
5. Enable the experience.

Note

If your account is part of an organization in AWS Organizations, see [Working with private marketplaces for AWS Organizations](#).

Create the Experience entity

To create the Experience entity, use the `StartChangeSet` action with the `CreateExperience` value for the `ChangeType` parameter to request that the experience be created by AWS Marketplace. See the following code example.

```
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"
}
```

In this action, `Entity` is a template for the entity that you want to create. It is assigned an `EntityId` when it is created. `ChangeSetName` identifies the change to help you find it later.

The response looks like the following.

```
{
  "ChangeSetArn": "arn:...:AWSMarketplace/ChangeSet/abcd1234example5678frjzkz",
  "ChangeSetId": "abcd1234example5678frjzkz"
}
```

The response includes a `ChangeSetId` that you can use to get the status of your change request as it is processed with `DescribeChangeSet`. You can also use `ListEntities` to find your Experience entity without the `ChangeSetId`. For more information about change sets, see [Working with change sets](#).

A newly created Experience entity doesn't have a procurement policy by default. It is also created with default settings for branding. For more information about branding settings, including how to customize them, see [Changing the branding of a private marketplace experience](#).

Create a `ProcurementPolicy` entity

You must create a `ProcurementPolicy` entity. By default, a new Experience entity is disabled, so you can create the procurement policy before enabling it.

Note

An Experience entity with *no* procurement policy (null) allows all products to be procured in your private marketplace. An Experience entity with an *empty* procurement policy has no products available to users to procure.

To allow and deny products in your private marketplace, you must create the procurement policy. To do this, you again call `StartChangeSet`, but this time with the `ChangeType` of `CreateProcurementPolicy`. The following code example creates an empty procurement policy.

```
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"
    }
  }
]
```

The Entity you provide in this action is the Experience entity that you want the procurement policy created within, so you must include the identifier for the entity that you created earlier. Use `ListEntities` to find the Experience entity. You can also return the identifier by using `DescribeChangeSet` with the change set identifier from the `CreateExperience` action

 Note

This example shows the identifier with a revision of 1. For more information about revisions for identifiers, see [Identifier](#).

You can again use `DescribeChangeSet` on the `CreateProcurementPolicy` change type to follow the processing of your request.

 Note

The names you give the Experience and ProcurementPolicy objects do not appear in AWS Marketplace. The names are only for your ease of finding the entities in the API.

After you have created the procurement policy, your private marketplace displays in the AWS Management Console. (You can go to the [Private Marketplace page](#) to see it.) After you have

completed these steps, your private marketplace will be disabled, have default branding, have an empty procurement policy, and will not be associated with any principals in your organization. You can update the branding and add any products that you want in it, associate the experience with one or more accounts, and then enable your private marketplace.

The following sections describing managing your private marketplace with the AWS Marketplace Catalog API.

Changing the branding of a private marketplace experience

You can customize the look of your private marketplace for your users. Without customization, your private marketplace will have the default branding settings, which are described below. Aspects of branding that you can change in a private marketplace include the following:

- **Title** – The name displayed for your private marketplace. This is the same as the **Name** field in the private marketplace **Profile settings** screen. If you set the Title to **Example**, then the text displayed is **Example Private Marketplace**. The default is **Private Marketplace**.
- **Information** – The paragraph displayed under the name in your private marketplace. This is the same as the **Description** field in **Profile settings**. The default is no information, in which case a general description of private marketplaces is displayed.
- **ThemeColor** – The color displayed in the banner of your private marketplace. This is a color in RGB hexadecimal format. This value is the same as the **Theme color** field in **Profile settings**. The default value is #232F3E.
- **LogoUrl** – The URL that points to an image file to be used as the logo on your private marketplace. The URL must be publicly available (for example, a signed Amazon S3 URL). The file must be either a .png or .svg file and be under 500kb. If necessary, the image file will be resized to a maximum height of 30 pixels and a maximum width of 100 pixels. This is the same value as the **Logo Select** in **Profile Settings**. The default is to not show a logo.

To set these values, you must first create a **BrandingSettings** entity with the **CreateBrandingSettings** change type. You can then request an **UpdateBrandingSettings** change to set or change the branding. You only need to create a **BrandingSettings** object once. To create this object, call **StartChangeSet** with the **CreateBrandingSettings** change type, as shown in the following code example.

```
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"
      }
    }
  ]
}
```

This example modifies the Experience entity by adding the BrandingSettings object to it. The revision of the entity identifier has incremented to 2. For more information about revisions for identifiers, see [Identifier](#).

 Note

You can specify all the details of the branding settings in the call to create the branding settings entity. The details facet is the same for CreateBrandingSettings and UpdateBrandingSettings.

You modify the settings by calling StartChangeSet with the UpdateBrandingSettings change type. The settings are part of the Configuration of the DetailsDocument object.

```
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

The URL for the logo is used to make a copy during the update change. After the change is complete, if you remove or change the URL at that path, it will not affect your private marketplace unless you again request `UpdateBrandingSettings`.

Enabling or disabling a private marketplace experience

When a private marketplace is enabled (and has a procurement policy), users in associated accounts can only purchase products that you have approved. When no private marketplace experience is enabled for an account, users can purchase products across the full AWS Marketplace catalog.

To enable a private marketplace, use the `StartChangeRequest` action with the `UpdateExperience` change type.

```
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"
      }
    }
  ]
}
```

Similarly, you can use the same action and `ChangeType`, but change the `Status` in `DetailsDocument` to `Disabled` to disable a private marketplace.

Note

Disabling a private marketplace keeps your list of both allowed and denied products, as well as customizations, such as branding. When a private marketplace is disabled, users no longer see the private marketplace (although they may still be governed by the default experience for the organization). If there are no private marketplace experiences enabled for an account, then all restrictions are removed, and users are able to procure any products in the public AWS Marketplace.

Enabling or disabling user requests

Users in your organization can view the full public AWS Marketplace, but they can only subscribe to the products that you have allowed. By default, they can request that a product that is not in the private marketplace be added to it. These requests show up in the private marketplace administrator page ([Private Marketplace](#)), where you can decide whether to accept or deny the request (and whether to block further requests for the same product). You cannot see or respond to the requests by using the Catalog API.

You can enable or disable the ability for users to create requests for your private marketplace experience. Use `StartChangeSet` with the `UpdateProcurementPolicy` change type. The ability to make requests is disabled in the following code example.

```
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"
      }
    }
  ]
}
```

To enable the change request capability for users, use `Allow` instead of `Deny` in `PolicyResourceRequests`.

To learn how to get the current status of this setting, see the next section, [Getting a list of products in a private marketplace experience](#).

Getting a list of products in a private marketplace experience

The products allowed (and denied) in a private marketplace are part of the procurement policy in the Experience entity. To get the details about the procurement policies in a private marketplace, you first get the procurement policy identifier from the Experience entity, and then call `DescribeEntity` with that identifier.

To get the procurement policy identifier, use `DescribeEntity` on the Experience entity that you are interested in, as shown in the following command.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=exp-example01
```

Following is an example response.

```
{
  "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

The `DetailsDocument` attribute contains the entity details as a JSON object. The legacy `Details` attribute contains the same JSON object as a string.

You can use the returned `EntityId` for the procurement policy to get the details, as shown in the following command.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=procpolicy-123example456
```

This returns the full details of the policy, including both allowed and denied products. Following is an example response.

```
{
  "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"
}

```

In this example, the procurement policy has two allowed products and one denied product. The policy allows user resource requests.

Adding or removing products from a private marketplace

By default, a private marketplace does not have any approved products in it. Use change requests to add or remove a product. To add a product, use the `AllowProductProcurement` change type. To remove a product, use the `DenyProductProcurement` change type.

The following code example shows the `AllowProductProcurement` change type with the `StartChangeSet` action to add a product to a private marketplace.

```

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"
}
]
```

You add the product to the Experience entity for a private marketplace by using `AllowProductProcurement`. The syntax to remove a product from a private marketplace is identical, with the exception that you use the `DenyProductProcurement` `ChangeType` instead of `AllowProductProcurement`. The products are added to the allow (or deny) list of the `ProcurementPolicy` entity that is contained by your Experience entity.

Note

The list of products in the `DetailsDocument` of your change is an array of `Ids`, so you can add (or remove) multiple products with one call by including a list of product identifiers. The limit is 50 products in a single request.

The `Notes` field for the list of `Ids` is not required. However, you can use it to record why a decision to allow or deny a set of products was made.

Finding products

By getting the details of your procurement policy, you can find the product IDs for the products that are already in a private marketplace. However, the AWS Marketplace Catalog API does not provide a way to find the product IDs for other products. There are two ways to get product IDs to use with the Catalog API service:

- **Public marketplace** – After you find a product in the public marketplace, choose **Continue to Subscribe** to see a details page about the product (it will not subscribe you to the product). The URL will include the product ID as a parameter. For example, in the URL `https://aws.amazon.com/marketplace/fulfillment?productId=ab1234cd-1234-abcd-5678-90abcdef1234&ref_=dtl_psb_continue`, *ab1234cd-1234-abcd-5678-90abcdef1234* is the product ID.
- **AWS Marketplace Discovery API** – Programmatically, you can access the full list of products in the AWS Marketplace by using the Discovery API. The Discovery API is a private API. You must request access to be able to use it. For more information, see [Access control for the AWS Marketplace Discovery API](#).

Working with private marketplaces for AWS Organizations

Whether you are working with a private marketplace for your account or your organization, you use the same API. However, there are differences when working within your organization:

- Before you can use private marketplace feature in an organization, you must [enable trusted access](#) to provide private marketplace service (`private-marketplace.marketplace.amazonaws.com`) access to your AWS Organizations data. You must also [create the private marketplace service-linked role](#) in the management account. This role includes all the permissions that private marketplace requires to describe AWS Organizations and update private marketplace resources on your behalf. These actions can only be performed by the management account. It is recommended to perform this enablement using private marketplace administrator page. If you are a new customer, see [Private marketplaces](#) in the *AWS Marketplace Buyer Guide*. If you are an existing customer, see [Creating and managing a private marketplace](#) in the *AWS Marketplace Buyer Guide*.
- Private marketplace resources in an organization are created in the management account and shared with the member account that is a delegated administrator for private marketplace.
- When listing objects in a private marketplace from a member account that is a delegated administrator for private marketplace, you must specifically request them with the `SharedWithMe` filter. This applies to both `ListEntities` and `ListChangeSets` actions.

To list Experience objects in your own account, call `ListEntities` as shown in the following code example.

```
POST /ListEntities HTTP/1.1
```

```
Content-Type: application/json

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

However, to list the entities that have been shared with you, you must add a `FilterList` with a `Scope` of `SharedWithMe`, as shown in the following code example. As a result, AWS Marketplace searches outside of your own account to find entities that are shared with you.

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

{"Catalog": "AWSMarketplace",
 "EntityType": "Experience",
 "FilterList":
  [{
    "Name": "Scope",
    "ValueList":
      ["SharedWithMe"]
  ]}]
```

In this case, only entities outside of your account (the ones for your organization) are returned.

Similarly, to call `ListChangeSets`, you must set the scope, as shown in the following code example.

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

{"Catalog": "AWSMarketplace",
 "FilterList":
  [{
    "Name": "Scope",
    "ValueList":
      ["SharedWithMe"]
  ]}]
```

This returns change sets that apply to a shared private marketplace for your organization.

Associating principals to experiences

A private marketplace experience must have one or more principals associated with it in order to have any effects in your organization. For a single AWS account, you must associate the account with the experience to use the private marketplace. In an organization, you can have multiple experiences apply to different principals.

Note

- The experience that is associated with the organization is the default for all other accounts in the organization. Associating a member account or OU with a different experience directly sets a different experience for the member account or child accounts of the OU.
- If you are a current private marketplace customer without the AWS Organizations integration for private marketplace, the experience that is associated with the management account is the default for all other accounts in the organization.

To associate a principal to an experience, use the `AssociateAudience` change type with the `StartChangeSet` action, as shown in the following code example.

```
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"
}
```

The *audience* is the list of *principals* that are associated with the Experience. A principal is an AWS account, organizational unit, or organization defined by its ID. *Principals* is a list, so you can include multiple principals to be associated with the experience. After the first call, subsequent calls to the `AssociateAudience` change type will add principals to the association for the experience.

You can also remove accounts from an experience. Use the `DisassociateAudience` change type to do this, as shown in the following code example.

```
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

A principal can only be directly associated with one experience. To move a principal from being directly associated with one experience to another experience, you must disassociate it from the initial experience, then associate it with the second.

Archiving and reactivating a private marketplace experience

You can remove a private marketplace experience by archiving it. Archived experiences can't be updated or used to govern accounts in your organization. If you have audiences associated with an archived experience, you can associate them with a different experience. If you decide to use the experience at a later time, you can always reactivate it. Administrators from the management account or a member account that is a delegated administrator for private marketplace have permissions to archive and reactivate experiences. If you're a current private marketplace customer without the AWS Organizations integration for private marketplace, administrators from the account that created the experience have permissions to archive and reactivate experiences.

Note

Before archiving an experience, you must disable it. For information about disabling an experience, see [Configuring your private marketplace](#) in the *AWS Marketplace Buyer Guide*.

To archive an experience, use the `RestrictExperience` change type with the `StartChangeSet` action, as shown in the following code example.

```
POST /StartChangeSet HTTP/1.1  
Content-type: application/json  
  
{  
  "Catalog": "AWSMarketplace",  
  "ChangeSet":
```

```
[
  {
    "ChangeType": "RestrictExperience",
    "DetailsDocument":
    {},
    "Entity":
    {
      "Type": "Experience@1.0",
      "Identifier": "exp-1234example"
    }
  }
]
```

To reactivate an experience, use the `ReviveExperience` change type with the `StartChangeSet` action, as shown in the following code example.

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

{
  "Catalog": "AWSMarketplace",
  "ChangeSet":
  [
    {
      "ChangeType": "ReviveExperience",
      "DetailsDocument":
      {},
      "Entity":
      {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example"
      }
    }
  ]
}
```

Errors in the private marketplace API

The following errors are specific to the private marketplace actions in the AWS Marketplace Catalog API.

Change type	Error code	Error message	Description
Errors returned directly by the StartChangeSet action			
All	422	Document not valid JSON format	Invalid JSON input used, check your syntax.
AllowProductProcurement, DenyProductProcurement	422	Values in Ids array must be unique	You can't include the same product multiple times in a single change request.
AllowProductProcurement, DenyProductProcurement	422	Cumulative number of values in Ids array must be less than or equal to 50	You can allow or deny up to 50 products in a single change request.
Errors found by calling the DescribeChangeSet action			
CreateBrandingSettings, UpdateBrandingSettings	INVALID_URL	Image could not be fetched from the input URL	You must specify a valid, reachable URL for the logo field in BrandingSettings .
CreateBrandingSettings, UpdateBrandingSettings	INVALID_IMAGE	Image verification for type, content, or file size failed. Only .png and .svg file types with sizes less than or equal to 500KB are supported.	Your image file must match the logo requirements for branding settings.
AllowProductProcurement, DenyProductProcurement	ENTITY_NOT_FOUND	Procurement policy missing from Experience	You must create a ProcurementPolicy before

Change type	Error code	Error message	Description
			allowing or denying products.
CreateProcurementPolicy	ENTITY_ALREADY_EXISTS	Procurement policy exists for Experience	You can only have a single procurement policy for a private marketplace.
UpdateProcurementPolicy	ENTITY_NOT_FOUND	Procurement policy missing from Experience	You must create a ProcurementPolicy before updating the procurement policy.
CreateBrandingSettings	ENTITY_ALREADY_EXISTS	Branding settings exists for Experience	You can only have a single branding settings for a private marketplace.
UpdateBrandingSettings	ENTITY_NOT_FOUND	Branding settings missing from Experience	You must create a BrandingSettings entity before updating the branding settings.
AssociateAudience	CALLER_NOT_AUTHORIZED	Caller not authorized to execute the action	You must have permissions to call the action. The accounts being added must be in the same organization.
CreateExperience	CALLER_NOT_AUTHORIZED	Caller not authorized to create experience.	You must have permissions to create an experience.

Change type	Error code	Error message	Description
AssociateAudience	ENTITY_ALREADY_EXISTS	An experience is already associated with the account {accountId}. Disassociate previous experience before updating	You can only associate a single experience with an account. Disassociate the current experience before associating a new one.
AssociateAudience, DisassociateAudience	ENTITY_IN_USE	There is already a conflicting change in progress for the selected account. Try again later	You can't change the association with an account while another change request to change the association is already in progress.

Entity types defined by private marketplace

The following table lists the private marketplace entity types, purpose, and actions on which each can be specified. Each entity type can be used to specify a resource Amazon Resource Name (ARN) that can be used in the AWS Identity and Access Management (IAM) policy. For more details on ARN formats, see [the section called “Catalog API entities”](#).

Entity	Purpose	Actions
Experience	Stores the top-level settings for a private marketplace	StartChangeSet DescribeEntity
BrandingSettings	Stores the branding settings for a private marketplace	DescribeEntity
ProcurementPolicy	Stores the procurement settings and lists of products in a private marketplace	DescribeEntity

Entity	Purpose	Actions
Audience	Stores the details of principals associated with a private marketplace	DescribeEntity

Using the AWS Marketplace API to share resources

AWS Marketplace Catalog API integrates with AWS Resource Access Manager (AWS RAM) to enable resource sharing. A *resource* is an entity that users can work with in AWS Marketplace, such as a product, an offer, or a resale authorization. With AWS RAM, you can share some AWS Marketplace Catalog API resources with other AWS accounts. You share resources that you own by creating a *resource share*. A resource share specifies the resources that you want to share and the consumers with whom to share them.

Contents

- [Prerequisites to share AWS Marketplace entities](#)
- [Share an AWS Marketplace entity](#)

Prerequisites to share AWS Marketplace entities

Before sharing entities in AWS Marketplace Catalog API, you must meet the following prerequisites:

- You can only have one resource policy attached to your AWS Marketplace entity.
- To share an AWS Marketplace entity, you must own it in your AWS account. This requirement means that the entity must be allocated or provisioned in your account. You can't share an AWS Marketplace entity that has been shared with you.

Share an AWS Marketplace entity

With AWS Marketplace resource sharing, entity owners can share their entities with other AWS accounts in AWS Marketplace. Entity-owners can be ISVs and channel partners. Entities that can be shared are products, offers, and resale authorizations.

Note

At this time, you can only share entities. Entities in AWS Marketplace include `AmiProduct`, `Audience`, `BrandingSettings`, `ContainerProduct`, `Experience`, and `ProcurementPolicy`.

For more information about AWS RAM, see the [AWS RAM User Guide](#). For more information about managing your shared resources, see [Using shared AWS resources](#) in the *AWS RAM User Guide*.

As a *sharing account*, you can set read-only or both read/write on the resources that you want to share. These permissions determine what operations a *consuming account* can perform on the resources that are shared with them.

- **Sharing account** – The resource that is shared and in which the AWS RAM administrator creates the AWS resource share by using AWS RAM.
- **Consuming account** – The AWS account to which a resource is shared. The resource share can specify an entire account as the principal, or for some resource types, individual roles or users in the account.

To share an AWS Marketplace entity, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share an entity using the AWS Marketplace console, you add it to an existing resource share. To add the AWS Marketplace entity to a new resource share, you must first create the resource share using the [AWS RAM console](#).

You can share an AWS Marketplace entity that you own using the AWS Marketplace console, AWS RAM console, or the AWS Command Line Interface (AWS CLI).

To share an AWS Marketplace entity that you own using the AWS RAM console

See [Creating a Resource Share](#) in the *AWS RAM User Guide*.

To share an AWS Marketplace entity that you own using the AWS CLI

Use the [create-resource-share](#) command.

Note

For resource types such as entities that support resource-based policies, you can use AWS RAM to share resources to use additional AWS RAM features. For more information, see [Resource-based policy](#) in the *AWS RAM User Guide*. AWS RAM uses the AWS Marketplace Catalog API to automatically construct the resource policy from permissions in a resource share and manages that resource policy for you.

For information about how to set, view, or delete AWS resource-based policies on your AWS Marketplace entity through AWS RAM, see [Allowing actions on all resources](#) in the *AWS RAM User Guide*.

Differences between sharing an entity through AWS RAM and the AWS Marketplace Catalog API

In addition to sharing your entity through AWS RAM, you can also set, view, or delete AWS resource-based policies on your entities through the AWS Marketplace Catalog API. However, there are a few differences between sharing your entity through AWS RAM and through the AWS Marketplace Catalog API.

When you share an entity through AWS RAM:

- If you share your entity with accounts that are outside of AWS Organizations, the consuming account must first accept your sharing request before the entity is shared.
- The consuming account can discover the shared entity through `ListEntities` with `OwnershipType` set to `SHARED`.
- You must adhere to several resource quotas. For more information, see [Service quotas for AWS RAM](#) in the *AWS RAM User Guide*.

When you share an entity through the AWS Marketplace Catalog API:

- Your entity will be shared as soon as the `PutResourcePolicy` request succeeds with no input from the consuming account.
- The consuming account can't discover the shared entity through `ListEntities` with `OwnershipType` set to `SHARED`. Instead, the owner of the sharing account must inform the consuming account of the shared entity ID.

Note

If your use case requires sharing resources that might exceed AWS RAM service quotas, or if you want to share resources without direct input from the consuming account, consider sharing through the AWS Marketplace Catalog API. For all other use cases, consider using AWS RAM to share AWS Marketplace resources.

The following sections detail how you can set, view, or delete AWS resource-based policies on your entities through the AWS Marketplace Catalog API.

Topics

- [Attach read-only policy to your resource](#)
- [Attach read and write resource policy to your resource](#)
- [View resource policy set on your resource](#)
- [Delete resource policy on your resource](#)
- [View all resources owned by you and shared with you](#)

Attach read-only policy to your resource

You can create a read-only resource-based policy on your shared resource using a sharing account. With this policy, the principal can only view the details of the resource that is shared with them.

Request

```
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"
      ]
    }
  }
}
```

```
}
```

Response

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

{}
```

Attach read and write resource policy to your resource

As a sharing account, you can create a read and write resource-based policy on your shared resource. With this policy, the principal can view the details and perform write operations on the resource that is shared with them.

Request

```
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"
      ]
    }
  }
}
```

Response

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

{}
```

View resource policy set on your resource

As a sharing account, you can view the resource policy that is set on your shared resource.

Request

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

{
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/
  AmiProduct/example2-abcd-1234-5ef6"
}
```

Response

```
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"
      ]
    }
  }
}
```

```
    }  
  }  
}
```

Delete resource policy on your resource

As a sharing account, you can delete the resource policy that is set on your shared resource.

Request

```
POST /DeleteResourcePolicy HTTP/1.1  
Content-type: application/json  
  
{  
  "ResourceArn": "arn:aws:aws-marketplace:us-east-1:123456789012:AWSMarketplace/  
  AmiProduct/example2-abcd-1234-5ef6"  
}
```

Response

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

View all resources owned by you and shared with you

As a consuming account, you can view the resources that are shared with you.

Note

You can view the resources shared with you only if the resources were shared through AWS RAM.

Request

```
POST /ListEntities HTTP/1.1
```

```
Content-type: application/json
{
  "Catalog": "AWSMarketplace",
  "EntityType": "AmiProduct",
  "FilterList": [
    {
      "Name": "EntityId",
      "ValueList": [ "example2-abcd-1234-5ef6" ]
    }
  ],
  "OwnershipType": "SHARED"
}
```

Response

```
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": ""
}
```

Using this service with an AWS SDK

AWS software development kits (SDKs) are available for many popular programming languages. Each SDK provides an API, code examples, and documentation that make it easier for developers to build applications in their preferred language.

SDK documentation	Code examples
AWS SDK for C++	AWS SDK for C++ code examples
AWS CLI	AWS CLI code examples
AWS SDK for Go	AWS SDK for Go code examples
AWS SDK for Java	AWS SDK for Java code examples
AWS SDK for JavaScript	AWS SDK for JavaScript code examples
AWS SDK for Kotlin	AWS SDK for Kotlin code examples
AWS SDK for .NET	AWS SDK for .NET code examples
AWS SDK for PHP	AWS SDK for PHP code examples
AWS Tools for PowerShell	AWS Tools for PowerShell code examples
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) code examples
AWS SDK for Ruby	AWS SDK for Ruby code examples
AWS SDK for Rust	AWS SDK for Rust code examples
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP code examples
AWS SDK for Swift	AWS SDK for Swift code examples

 Example availability

Can't find what you need? Request a code example by using the **Provide feedback** link at the bottom of this page.

Code examples for AWS Marketplace using AWS SDKs

The following code examples show how to use AWS Marketplace with an AWS software development kit (SDK).

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Code examples

- [Code examples for AWS Marketplace Catalog API using AWS SDKs](#)
 - [AMI products for AWS Marketplace Catalog API](#)
 - [Add a dimension to an existing AMI product and update the offer pricing terms using an AWS SDK](#)
 - [Add a region where an AMI product is deployed using an AWS SDK](#)
 - [Create a public or limited AMI product and a public offer with hourly annual pricing using an AWS SDK](#)
 - [Create a public or limited AMI product and public offer with hourly monthly pricing using an AWS SDK](#)
 - [Create a public or limited AMI product and public offer with hourly pricing using an AWS SDK](#)
 - [Create a draft AMI product with a draft public offer using an AWS SDK](#)
 - [Restrict a region where an AMI product is deployed using an AWS SDK](#)
 - [Restrict product visibility using an AWS SDK](#)
 - [Specify whether AMI assets are deployed in new regions using an AWS SDK](#)
 - [Channel partner offers for AWS Marketplace Catalog API](#)
 - [Create a draft CPPO for any product type using an AWS SDK](#)
 - [Create a resale authorization replacement private offer from an existing agreement with contract pricing using an AWS SDK](#)
 - [List all CPPOs created by a channel partner using an AWS SDK](#)
 - [List all shared resale authorizations available to a channel partner using an AWS SDK](#)
 - [Publish a CPPO and append a buyer EULA using an AWS SDK](#)

- [Publish a CPPO using one-time resale authorization and update price markup using an AWS SDK](#)
- [Publish a draft CPPO and update price markup using an AWS SDK](#)
- [Update the expiration date of a CPPO using an AWS SDK](#)
- [Container products for AWS Marketplace Catalog API](#)
 - [Create a draft container product with a draft public offer using an AWS SDK](#)
 - [Create a limited container product with public offer, contract pricing using an AWS SDK](#)
- [Entities for AWS Marketplace Catalog API](#)
 - [Describe all entities in a single call using an AWS SDK](#)
 - [List and describe all offers associated with a product using an AWS SDK](#)
- [Offers for AWS Marketplace Catalog API](#)
 - [Create a custom dimension for a SaaS product and create a private offer using an AWS SDK](#)
 - [Create a draft private offer for an AMI or SaaS product using an AWS SDK](#)
 - [Create a private offer with contract and Pay-As-You-Go pricing for a SaaS product using an AWS SDK](#)
 - [Create a private offer with contract pricing and a flexible payment schedule for a SaaS product using an AWS SDK](#)
 - [Create a private offer with contract pricing for a Container product using an AWS SDK](#)
 - [Create a private offer with contract pricing for an AMI product using an AWS SDK](#)
 - [Create a private offer with hourly annual pricing and a flexible payment schedule for an AMI product using an AWS SDK](#)
 - [Create a private offer with hourly annual pricing for an AMI product using an AWS SDK](#)
 - [Create a private offer with hourly pricing for an AMI product using an AWS SDK](#)
 - [Create a private offer with subscription pricing for a SaaS product using an AWS SDK](#)
 - [Create a private offer with tiered contract pricing for a SaaS product using an AWS SDK](#)
 - [Create a public free trial offer with subscription pricing for a SaaS product using an AWS SDK](#)
 - [Create a replacement private offer from an existing agreement with contract pricing using an AWS SDK](#)
- [Describe a public offer using an AWS SDK](#)

- [Expire a private offer using an AWS SDK](#)
- [List all private offers using an AWS SDK](#)
- [List released public and private offers for a specific product ID using an AWS SDK](#)
- [Update an offer to apply a contract with Pay-As-You-Go pricing using an AWS SDK](#)
- [Update an offer to apply hourly annual pricing using an AWS SDK](#)
- [Update an offer to apply targeting to specific geographic regions using an AWS SDK](#)
- [Update name and description of a public offer using an AWS SDK](#)
- [Update the EULA of an offer using an AWS SDK](#)
- [Update the expiration date of a private offer to a future date using an AWS SDK](#)
- [Update the free trial duration of a public free trial offer for a SaaS product using an AWS SDK](#)
- [Update the refund policy of an offer using an AWS SDK](#)
- [Products for AWS Marketplace Catalog API](#)
 - [Describe an AMI, SaaS, or Container product using an AWS SDK](#)
 - [List all AMI, SaaS, or Container products and associated public offers using an AWS SDK](#)
- [Resale authorization for AWS Marketplace Catalog API](#)
 - [Create draft resale authorization for any product type using an AWS SDK](#)
 - [Describe a resale authorization using an AWS SDK](#)
 - [Publish a one-time resale authorization with a private offer using an AWS SDK](#)
 - [Publish multi-use resale authorization with an expiration date using an AWS SDK](#)
 - [Publish multi-use resale authorization with an expiration date and a EULA to be sent to the buyer using an AWS SDK](#)
 - [Publish multi-use resale authorization with an expiration date and add reseller contract documentation using an AWS SDK](#)
 - [Publish multi-use resale authorization with expiration and add a specific buyer account for the resale using an AWS SDK](#)
 - [Publish multi-use resale authorization without an expiration date using an AWS SDK](#)
 - [Publish multi-use resale authorization without an expiration date and add a custom EULA using an AWS SDK](#)
 - [Publish multi-use resale authorization without an expiration date and add reseller contract documentation using an AWS SDK](#)

- [Publish multi-use resale authorization without expiration date and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish one-time resale authorization and add Flexible payment schedule using an AWS SDK](#)
- [Publish one-time resale authorization for any product type and add a EULA using an AWS SDK](#)
- [Publish one-time resale authorization and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish one-time resale authorization for any product type and add reseller contract documentation using an AWS SDK](#)
- [Publish one-time resale authorization for and add whether it is a renewal using an AWS SDK](#)
- [Restrict resale authorization using an AWS SDK](#)
- [Update name and description of one-time or multi-use resale authorization using an AWS SDK](#)
- [SaaS products for AWS Marketplace Catalog API](#)
 - [Create a draft SaaS product with a draft public offer using an AWS SDK](#)
 - [Create a public or limited SaaS product and public offer with contract pricing using an AWS SDK](#)
 - [Create a public or limited SaaS product and public offer with contract with Pay-As-You-Go pricing using an AWS SDK](#)
 - [Create a public or limited SaaS product and public offer with subscription pricing using an AWS SDK](#)
 - [Publish a SaaS product and associated public offer using an AWS SDK](#)
 - [Publish a SaaS product and associated public offer from an existing draft using an AWS SDK](#)
 - [Update dimensions on an AMI or SaaS product using an AWS SDK](#)
- [Utilities for AWS Marketplace Catalog API](#)
 - [Utilities to start a changeset using an AWS SDK](#)
- [Code examples for AWS Marketplace Agreement API using AWS SDKs](#)
 - [Agreements for AWS Marketplace Agreement API](#)
 - [Accept an agreement cancellation request using an AWS SDK](#)

- [Accept an agreement payment request using an AWS SDK](#)
- [Amend a SaaS contract agreement renewal term using an AWS SDK](#)
- [Amend an AMI ConfigurableUpfrontPricingTerm for usage pricing model using an AWS SDK](#)
- [Create a new AMI free trial agreement using an AWS SDK](#)
- [Create a new SaaS contract agreement with upfront payment using an AWS SDK](#)
- [Get all agreement IDs using an AWS SDK](#)
- [Get all agreements using an AWS SDK](#)
- [Get customer ID from an agreement using an AWS SDK](#)
- [Get details of an agreement cancellation request using an AWS SDK](#)
- [Get details of an agreement payment request using an AWS SDK](#)
- [Get financial details from an agreement using an AWS SDK](#)
- [Get free trial details from an agreement using an AWS SDK](#)
- [Get information about an agreement using an AWS SDK](#)
- [Get product and offer details from an agreement using an AWS SDK](#)
- [Get the EULA of an agreement using an AWS SDK](#)
- [Get the auto renewal terms of an agreement using an AWS SDK](#)
- [Get the dimensions purchased in an agreement using an AWS SDK](#)
- [Get the instances of each dimension purchased in an agreement using an AWS SDK](#)
- [Get the payment schedule of an agreement using an AWS SDK](#)
- [Get the pricing per dimension in an agreement using an AWS SDK](#)
- [Get the pricing type of an agreement using an AWS SDK](#)
- [Get the product type of an agreement using an AWS SDK](#)
- [Get the status of an agreement using an AWS SDK](#)
- [Get the support terms of an agreement using an AWS SDK](#)
- [Get the terms of an agreement using an AWS SDK](#)
- [List agreement cancellation requests using an AWS SDK](#)
- [List agreement payment requests using an AWS SDK](#)
- [Reject an agreement cancellation request using an AWS SDK](#)
- [Reject an agreement payment request using an AWS SDK](#)

- [Replace a SaaS contract agreement with an agreement-based offer using an AWS SDK](#)
- [Replace a SaaS free trial with Contract with Consumption Pricing using an AWS SDK](#)
- [Replace a SaaS usage-based pricing term and cancel the previous agreement using an AWS SDK](#)
- [Replace an AMI usage-based pricing term but keep the existing ConfigurableUpfrontPricingTerm using an AWS SDK](#)
- [Search for agreements by account ID using an AWS SDK](#)
- [Search for agreements by agreement ID using an AWS SDK](#)
- [Search for agreements by end date using an AWS SDK](#)
- [Search for agreements by offer ID using an AWS SDK](#)
- [Search for agreements by product ID using an AWS SDK](#)
- [Search for agreements by status using an AWS SDK](#)
- [Search for agreements with one custom filter using an AWS SDK](#)
- [Search for agreements with two custom filters using an AWS SDK](#)
- [Update purchase orders after agreement acceptance using an AWS SDK](#)

Code examples for AWS Marketplace Catalog API using AWS SDKs

The following code examples show how to use AWS Marketplace Catalog API with an AWS software development kit (SDK).

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

AWS Marketplace Catalog API

- [AMI products for AWS Marketplace Catalog API](#)
 - [Add a dimension to an existing AMI product and update the offer pricing terms using an AWS SDK](#)
 - [Add a region where an AMI product is deployed using an AWS SDK](#)
 - [Create a public or limited AMI product and a public offer with hourly annual pricing using an AWS SDK](#)

- [Create a public or limited AMI product and public offer with hourly monthly pricing using an AWS SDK](#)
- [Create a public or limited AMI product and public offer with hourly pricing using an AWS SDK](#)
- [Create a draft AMI product with a draft public offer using an AWS SDK](#)
- [Restrict a region where an AMI product is deployed using an AWS SDK](#)
- [Restrict product visibility using an AWS SDK](#)
- [Specify whether AMI assets are deployed in new regions using an AWS SDK](#)
- [Channel partner offers for AWS Marketplace Catalog API](#)
 - [Create a draft CPPO for any product type using an AWS SDK](#)
 - [Create a resale authorization replacement private offer from an existing agreement with contract pricing using an AWS SDK](#)
 - [List all CPPOs created by a channel partner using an AWS SDK](#)
 - [List all shared resale authorizations available to a channel partner using an AWS SDK](#)
 - [Publish a CPPO and append a buyer EULA using an AWS SDK](#)
 - [Publish a CPPO using one-time resale authorization and update price markup using an AWS SDK](#)
 - [Publish a draft CPPO and update price markup using an AWS SDK](#)
 - [Update the expiration date of a CPPO using an AWS SDK](#)
- [Container products for AWS Marketplace Catalog API](#)
 - [Create a draft container product with a draft public offer using an AWS SDK](#)
 - [Create a limited container product with public offer, contract pricing using an AWS SDK](#)
- [Entities for AWS Marketplace Catalog API](#)
 - [Describe all entities in a single call using an AWS SDK](#)
 - [List and describe all offers associated with a product using an AWS SDK](#)
- [Offers for AWS Marketplace Catalog API](#)
 - [Create a custom dimension for a SaaS product and create a private offer using an AWS SDK](#)
 - [Create a draft private offer for an AMI or SaaS product using an AWS SDK](#)
 - [Create a private offer with contract and Pay-As-You-Go pricing for a SaaS product using an AWS SDK](#)
 - [Create a private offer with contract pricing and a flexible payment schedule for a SaaS product using an AWS SDK](#)

- [Create a private offer with contract pricing for a Container product using an AWS SDK](#)
- [Create a private offer with contract pricing for an AMI product using an AWS SDK](#)
- [Create a private offer with hourly annual pricing and a flexible payment schedule for an AMI product using an AWS SDK](#)
- [Create a private offer with hourly annual pricing for an AMI product using an AWS SDK](#)
- [Create a private offer with hourly pricing for an AMI product using an AWS SDK](#)
- [Create a private offer with subscription pricing for a SaaS product using an AWS SDK](#)
- [Create a private offer with tiered contract pricing for a SaaS product using an AWS SDK](#)
- [Create a public free trial offer with subscription pricing for a SaaS product using an AWS SDK](#)
- [Create a replacement private offer from an existing agreement with contract pricing using an AWS SDK](#)
- [Describe a public offer using an AWS SDK](#)
- [Expire a private offer using an AWS SDK](#)
- [List all private offers using an AWS SDK](#)
- [List released public and private offers for a specific product ID using an AWS SDK](#)
- [Update an offer to apply a contract with Pay-As-You-Go pricing using an AWS SDK](#)
- [Update an offer to apply hourly annual pricing using an AWS SDK](#)
- [Update an offer to apply targeting to specific geographic regions using an AWS SDK](#)
- [Update name and description of a public offer using an AWS SDK](#)
- [Update the EULA of an offer using an AWS SDK](#)
- [Update the expiration date of a private offer to a future date using an AWS SDK](#)
- [Update the free trial duration of a public free trial offer for a SaaS product using an AWS SDK](#)
- [Update the refund policy of an offer using an AWS SDK](#)
- [Products for AWS Marketplace Catalog API](#)
 - [Describe an AMI, SaaS, or Container product using an AWS SDK](#)
 - [List all AMI, SaaS, or Container products and associated public offers using an AWS SDK](#)
- [Resale authorization for AWS Marketplace Catalog API](#)
 - [Create draft resale authorization for any product type using an AWS SDK](#)
 - [Describe a resale authorization using an AWS SDK](#)
 - [Publish a one-time resale authorization with a private offer using an AWS SDK](#)
 - [Publish multi-use resale authorization with an expiration date using an AWS SDK](#)

- [Publish multi-use resale authorization with an expiration date and a EULA to be sent to the buyer using an AWS SDK](#)
- [Publish multi-use resale authorization with an expiration date and add reseller contract documentation using an AWS SDK](#)
- [Publish multi-use resale authorization with expiration and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish multi-use resale authorization without an expiration date using an AWS SDK](#)
- [Publish multi-use resale authorization without an expiration date and add a custom EULA using an AWS SDK](#)
- [Publish multi-use resale authorization without an expiration date and add reseller contract documentation using an AWS SDK](#)
- [Publish multi-use resale authorization without expiration date and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish one-time resale authorization and add Flexible payment schedule using an AWS SDK](#)
- [Publish one-time resale authorization for any product type and add a EULA using an AWS SDK](#)
- [Publish one-time resale authorization and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish one-time resale authorization for any product type and add reseller contract documentation using an AWS SDK](#)
- [Publish one-time resale authorization for and add whether it is a renewal using an AWS SDK](#)
- [Restrict resale authorization using an AWS SDK](#)
- [Update name and description of one-time or multi-use resale authorization using an AWS SDK](#)
- [SaaS products for AWS Marketplace Catalog API](#)
 - [Create a draft SaaS product with a draft public offer using an AWS SDK](#)
 - [Create a public or limited SaaS product and public offer with contract pricing using an AWS SDK](#)
 - [Create a public or limited SaaS product and public offer with contract with Pay-As-You-Go pricing using an AWS SDK](#)
 - [Create a public or limited SaaS product and public offer with subscription pricing using an AWS SDK](#)
 - [Publish a SaaS product and associated public offer using an AWS SDK](#)
 - [Publish a SaaS product and associated public offer from an existing draft using an AWS SDK](#)

- [Update dimensions on an AMI or SaaS product using an AWS SDK](#)
- [Utilities for AWS Marketplace Catalog API](#)
- [Utilities to start a changeset using an AWS SDK](#)

AMI products for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Add a dimension to an existing AMI product and update the offer pricing terms using an AWS SDK](#)
- [Add a region where an AMI product is deployed using an AWS SDK](#)
- [Create a public or limited AMI product and a public offer with hourly annual pricing using an AWS SDK](#)
- [Create a public or limited AMI product and public offer with hourly monthly pricing using an AWS SDK](#)
- [Create a public or limited AMI product and public offer with hourly pricing using an AWS SDK](#)
- [Create an draft AMI product with a draft public offer using an AWS SDK](#)
- [Restrict a region where an AMI product is deployed using an AWS SDK](#)
- [Restrict product visibility using an AWS SDK](#)
- [Specify whether AMI assets are deployed in new regions using an AWS SDK](#)

Add a dimension to an existing AMI product and update the offer pricing terms using an AWS SDK

The following code examples show how to add a dimension to an existing AMI product and update the offer pricing terms.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Add a region where an AMI product is deployed using an AWS SDK

The following code examples show how to add a region where an AMI product is deployed.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.


```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-11111111111111"
      },
      "DetailsDocument": {
        "Regions": [
          "us-east-2",
          "us-west-2"
        ]
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AddRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-11111111111111"
      }
    }
  ]
}
```

```

        },
        "DetailsDocument": {
            "Regions": [
                "us-east-2",
                "us-west-2"
            ]
        }
    ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a public or limited AMI product and a public offer with hourly annual pricing using an AWS SDK

The following code examples show how to create a public or limited AMI product and a public offer with hourly annual pricing. This example creates either a standard or custom EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProduct",
      "ChangeName": "CreateProductChange",
```

```

        "Entity": {
            "Type": "AmiProduct@1.0"
        },
        "DetailsDocument": {}
    },
    {
        "ChangeType": "UpdateInformation",
        "Entity": {
            "Type": "AmiProduct@1.0",
            "Identifier": "$CreateProductChange.Entity.Identifier"
        },
        "DetailsDocument": {
            "ProductTitle": "Sample product",
            "ShortDescription": "Brief description",
            "LongDescription": "Detailed description",
            "Highlights": [
                "Sample highlight"
            ],
            "SearchKeywords": [
                "Sample keyword"
            ],
            "Categories": [
                "Operating Systems"
            ],
            "LogoUrl": "https://s3.amazonaws.com/logos/sample.png",
            "VideoUrls": [
                "https://sample.amazonaws.com/awsmp-video-1"
            ],
            "AdditionalResources": []
        }
    },
    {
        "ChangeType": "AddRegions",
        "Entity": {
            "Type": "AmiProduct@1.0",
            "Identifier": "$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": {}
}
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a public or limited AMI product and public offer with hourly monthly pricing using an AWS SDK

The following code examples show how to create a public or limited AMI product and public offer with hourly monthly pricing. This example creates either a standard or custom EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a public or limited AMI product and public offer with hourly pricing using an AWS SDK

The following code examples show how to create a public or limited AMI product and public offer with hourly pricing. This example creates either a standard or custom EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

```
]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create an draft AMI product with a draft public offer using an AWS SDK

The following code examples show how to create an draft AMI product with a draft public offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Restrict a region where an AMI product is deployed using an AWS SDK

The following code examples show how to restrict a region where an AMI product is deployed.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "Regions": [
          "us-west-2"
        ]
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictRegions",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "Regions": [
          "us-west-2"
        ]
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Restrict product visibility using an AWS SDK

The following code examples show how to restrict product visibility.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateVisibility",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "TargetVisibility": "Restricted"
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateVisibility",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-11111111111111"
      },
      "DetailsDocument": {
        "TargetVisibility": "Restricted"
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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.usagem_demo(change_set, "Restrict existing AMI")
```

```
if __name__ == "__main__":  
    main()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Specify whether AMI assets are deployed in new regions using an AWS SDK

The following code examples show how to specify whether AMI assets are deployed in new regions built by AWS to support future regions.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{  
  "Catalog": "AWSMarketplace",  
  "ChangeSet": [  
    {  
      "ChangeType": "UpdateFutureRegionSupport",  
      "Entity": {  
        "Type": "AmiProduct@1.0",  
        "Identifier": "prod-11111111111111"  
      },  
      "DetailsDocument": {  
        "FutureRegionSupport": {  
          "SupportedRegions": [  
            "All"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```

    ]
  }
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateFutureRegionSupport",
      "Entity": {
        "Type": "AmiProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "FutureRegionSupport": {
          "SupportedRegions": [
            "All"
          ]
        }
      }
    }
  ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Channel partner offers for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Create a draft CPPO for any product type using an AWS SDK](#)

- [Create a resale authorization replacement private offer from an existing agreement with contract pricing using an AWS SDK](#)
- [List all CPPOs created by a channel partner using an AWS SDK](#)
- [List all shared resale authorizations available to a channel partner using an AWS SDK](#)
- [Publish a CPPO and append a buyer EULA using an AWS SDK](#)
- [Publish a CPPO using one-time resale authorization and update price markup using an AWS SDK](#)
- [Publish a draft CPPO and update price markup using an AWS SDK](#)
- [Update the expiration date of a CPPO using an AWS SDK](#)

Create a draft CPPO for any product type using an AWS SDK

The following code examples show how to create a draft CPPO for any product type so you can review them internally before publishing to buyers.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "11111111-1111-1111-1111-111111111111",

```

```

        "Name": "Test Offer",
        "Description": "Test product"
    }
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ResaleAuthorizationId": "11111111-1111-1111-1111-111111111111",
        "Name": "Test Offer name"
      }
    }
  ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a resale authorization replacement private offer from an existing agreement with contract pricing using an AWS SDK

The following code examples show how to create a resale authorization replacement private offer from an existing agreement with contract pricing.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateReplacementOfferUsingResaleAuthorization",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "ChangeName": "CreateReplacementOfferResaleAuth",
      "DetailsDocument": {
        "AgreementId": "agmt-11111111111111111111111111111111",
        "ResaleAuthorizationId": "resaleauthz-11111111111111111111111111111111"
      }
    },
    {
      "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": {}
}
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType" : "CreateReplacementOfferUsingResaleAuthorization",
            "Entity": {
                "Type": "Offer@1.0"
            },
            "ChangeName": "CreateReplacementOfferResaleAuth",
            "DetailsDocument": {
                "AgreementId": "agmt-11111111111111111111111111111111",

```

```

        "ResaleAuthorizationId": "resaleauthz-111111111111"
    },
    {
        "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List all CPPOs created by a channel partner using an AWS SDK

The following code examples show how to list all CPPOs created by a channel partner.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [ListEntities](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

# 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())

```

- For API details, see [ListEntities](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List all shared resale authorizations available to a channel partner using an AWS SDK

The following code examples show how to list all shared resale authorizations available to a channel partner.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}

}
```

- For API details, see [ListEntities](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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)
```

- For API details, see [ListEntities](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish a CPPO and append a buyer EULA using an AWS SDK

The following code examples show how to publish a CPPO and append a buyer EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish a CPPO using one-time resale authorization and update price markup using an AWS SDK

The following code examples show how to publish a CPPO using one-time resale authorization on AMI, SaaS, or Container products and update price markup.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish a draft CPPO and update price markup using an AWS SDK

The following code examples show how to publish a draft CPPO and update price markup.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "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": {}
}

```

```
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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 CPP0 for any product type adn update price markup"
    )

```

```
)  
  
    return response  
  
if __name__ == "__main__":  
    main()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update the expiration date of a CPPO using an AWS SDK

The following code examples show how to update the expiration date of a CPPO to give buyers more time to evaluate and accept the offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{  
  "Catalog": "AWSMarketplace",  
  "ChangeSet": [  
    {  
      "ChangeType": "UpdateAvailability",  
      "Entity": {  
        "Type": "Offer@1.0",
```

```

        "Identifier": "offer-111111111111"
    },
    "DetailsDocument": {
        "AvailabilityEndDate": "2025-07-31"
    }
}
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "UpdateAvailability",
            "Entity": {
                "Type": "Offer@1.0",
                "Identifier": "offer-111111111111"
            },
            "DetailsDocument": {
                "AvailabilityEndDate": "2025-07-31"
            }
        }
    ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Container products for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Create a draft container product with a draft public offer using an AWS SDK](#)
- [Create a limited container product with public offer, contract pricing using an AWS SDK](#)

Create a draft container product with a draft public offer using an AWS SDK

The following code examples show how to create a draft container product with a draft public offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

```
{
  "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"
        }
    }
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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"
        }
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a limited container product with public offer, contract pricing using an AWS SDK

The following code examples show how to create a limited container product with a public offer, contract pricing, and standard EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Entities for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Describe all entities in a single call using an AWS SDK](#)
- [List and describe all offers associated with a product using an AWS SDK](#)

Describe all entities in a single call using an AWS SDK

The following code examples show how to describe all entities in a single call.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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()));
    }
}
}

```

- For API details, see [BatchDescribeEntities](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
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()

```

- For API details, see [BatchDescribeEntities](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List and describe all offers associated with a product using an AWS SDK

The following code examples show how to list and describe all offers associated with a product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
    }
}

```

- For API details, see the following topics in *AWS SDK for Java 2.x API Reference*.
 - [DescribeEntity](#)
 - [ListEntities](#)

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
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}")
```

- For API details, see the following topics in *AWS SDK for Python (Boto3) API Reference*.
 - [DescribeEntity](#)
 - [ListEntities](#)

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Offers for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Create a custom dimension for a SaaS product and create a private offer using an AWS SDK](#)
- [Create a draft private offer for an AMI or SaaS product using an AWS SDK](#)
- [Create a private offer with contract and Pay-As-You-Go pricing for a SaaS product using an AWS SDK](#)
- [Create a private offer with contract pricing and a flexible payment schedule for a SaaS product using an AWS SDK](#)
- [Create a private offer with contract pricing for a Container product using an AWS SDK](#)
- [Create a private offer with contract pricing for an AMI product using an AWS SDK](#)
- [Create a private offer with hourly annual pricing and a flexible payment schedule for an AMI product using an AWS SDK](#)

- [Create a private offer with hourly annual pricing for an AMI product using an AWS SDK](#)
- [Create a private offer with hourly pricing for an AMI product using an AWS SDK](#)
- [Create a private offer with subscription pricing for a SaaS product using an AWS SDK](#)
- [Create a private offer with tiered contract pricing for a SaaS product using an AWS SDK](#)
- [Create a public free trial offer with subscription pricing for a SaaS product using an AWS SDK](#)
- [Create a replacement private offer from an existing agreement with contract pricing using an AWS SDK](#)
- [Describe a public offer using an AWS SDK](#)
- [Expire a private offer using an AWS SDK](#)
- [List all private offers using an AWS SDK](#)
- [List released public and private offers for a specific product ID using an AWS SDK](#)
- [Update an offer to apply a contract with Pay-As-You-Go pricing using an AWS SDK](#)
- [Update an offer to apply hourly annual pricing using an AWS SDK](#)
- [Update an offer to apply targeting to specific geographic regions using an AWS SDK](#)
- [Update name and description of a public offer using an AWS SDK](#)
- [Update the EULA of an offer using an AWS SDK](#)
- [Update the expiration date of a private offer to a future date using an AWS SDK](#)
- [Update the free trial duration of a public free trial offer for a SaaS product using an AWS SDK](#)
- [Update the refund policy of an offer using an AWS SDK](#)

Create a custom dimension for a SaaS product and create a private offer using an AWS SDK

The following code examples show how to create a custom dimension for a SaaS product and create a private offer.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

```
{
  "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"
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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-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"
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a draft private offer for an AMI or SaaS product using an AWS SDK

The following code examples show how to create a draft private offer for an AMI or SaaS product so you can review it internally before publishing to buyers.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "Test Private Offer"
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
```

```

        "Name": "Test Private Offer"
    }
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with contract and Pay-As-You-Go pricing for a SaaS product using an AWS SDK

The following code examples show how to create a private offer with contract and Pay-As-You-Go pricing for a SaaS product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with contract pricing and a flexible payment schedule for a SaaS product using an AWS SDK

The following code examples show how to create a private offer with contract pricing and a flexible payment schedule for a SaaS product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "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": {}
}
]

```

```
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with contract pricing for a Container product using an AWS SDK

The following code examples show how to create a private offer with contract pricing for a Container product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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": {}
        }
    ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with contract pricing for an AMI product using an AWS SDK

The following code examples show how to create a private offer with contract pricing for an AMI product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with hourly annual pricing and a flexible payment schedule for an AMI product using an AWS SDK

The following code examples show how to create a private offer with hourly annual pricing and a flexible payment schedule for an AMI product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with hourly annual pricing for an AMI product using an AWS SDK

The following code examples show how to create a private offer with hourly annual pricing for an AMI product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```

{
  "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": {}
    }
  ]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with hourly pricing for an AMI product using an AWS SDK

The following code examples show how to create a private offer with hourly pricing for an AMI product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with subscription pricing for a SaaS product using an AWS SDK

The following code examples show how to create a private offer with subscription pricing for a SaaS product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a private offer with tiered contract pricing for a SaaS product using an AWS SDK

The following code examples show how to create a private offer with tiered contract pricing for a SaaS product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a public free trial offer with subscription pricing for a SaaS product using an AWS SDK

The following code examples show how to create a public free trial offer with subscription pricing for a SaaS product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{  
  "Catalog": "AWSMarketplace",  
  "ChangeSet": [  
    {  
      "ChangeType": "CreateOffer",  
      "Entity": {  
        "Type": "Offer@1.0"  
      },  
      "ChangeName": "CreateOfferChange",  
      "DetailsDocument": {  
        "ProductId": "prod-11111111111111"  
      }  
    }  
  ]  
}
```

```

    }
  },
  {
    "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": {}
  }
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a replacement private offer from an existing agreement with contract pricing using an AWS SDK

The following code examples show how to create a replacement private offer from an existing agreement with contract pricing.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Describe a public offer using an AWS SDK

The following code examples show how to describe a public offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
    }
}
```


- For API details, see [DescribeEntity](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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-111111111111"

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()

```

- For API details, see [DescribeEntity](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Expire a private offer using an AWS SDK

The following code examples show how to set the expiration date of a private offer to a date in the past so that buyers no longer see the offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2023-01-01"
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateOffer",
      "Entity": {
        "Type": "Offer@1.0"
      },
      "DetailsDocument": {
        "ProductId": "prod-111111111111",
        "Name": "Test Private Offer"
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List all private offers using an AWS SDK

The following code examples show how to list all private offers.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```


- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List released public and private offers for a specific product ID using an AWS SDK

The following code example shows how to list released public and private offers for a specific product ID.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();

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;
}
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update an offer to apply a contract with Pay-As-You-Go pricing using an AWS SDK

The following code examples show how to update an offer to apply a contract with Pay-As-You-Go pricing.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```



```

    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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"
      }
    }
  ]
}
]
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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

CAP-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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update an offer to apply hourly annual pricing using an AWS SDK

The following code examples show how to update an offer to apply hourly annual pricing.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

```
{
  "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"
                }
            }
        ]
    }
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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"
          }
        }
      ]
    }
  ]
}
]

```

```

    }
  }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update an offer to apply targeting to specific geographic regions using an AWS SDK

The following code examples show how to update an offer to apply targeting to specific geographic regions.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "ES",
            "FR",
            "AU"
          ]
        }
      }
    }
  ]
}
```


- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateTargeting",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "PositiveTargeting": {
          "CountryCodes": [
            "US",
            "ES",
            "FR",
            "AU"
          ]
        }
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update name and description of a public offer using an AWS SDK

The following code examples show how to update name and description of a public offer.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

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

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "Name": "New offer name",
        "Description": "New offer description"
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update the EULA of an offer using an AWS SDK

The following code examples show how to update the EULA of an offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

    "ChangeSet": [
      {
        "ChangeType": "UpdateInformation",
        "Entity": {
          "Type": "Offer@1.0",
          "Identifier": "offer-111111111111"
        },
        "DetailsDocument": {
          "Name": "New offer name",
          "Description": "New offer description"
        }
      }
    ]
  }
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateLegalTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "LegalTerm",
            "Documents": [
              {

```

```
custom-eula.pdf"
    "Type": "CustomEula",
    "Url": "https://s3.amazonaws.com/sample-bucket/"
  }
}
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update the expiration date of a private offer to a future date using an AWS SDK

The following code examples show how to update the expiration date of a private offer to a date in the future to give buyers more time to evaluate and accept the offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2026-01-01"
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateAvailability",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "AvailabilityEndDate": "2026-01-01"
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update the free trial duration of a public free trial offer for a SaaS product using an AWS SDK

The following code examples show how to update the free trial duration of a public free trial offer for a SaaS product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "FreeTrialPricingTerm",
            "Duration": "P21D",
            "Grants": [
              {
                "DimensionKey": "WorkloadSmall"
              },
              {
                "DimensionKey": "WorkloadMedium"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdatePricingTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-111111111111"
      },
      "DetailsDocument": {
        "PricingModel": "Usage",
        "Terms": [
          {
            "Type": "FreeTrialPricingTerm",
            "Duration": "P21D",
            "Grants": [
              {
                "DimensionKey": "WorkloadSmall"
              },
              {
                "DimensionKey": "WorkloadMedium"
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update the refund policy of an offer using an AWS SDK

The following code examples show how to update the refund policy of an offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      },
      "DetailsDocument": {
        "Terms": [
          {
            "Type": "SupportTerm",
            "RefundPolicy": "Updated refund policy description"
          }
        ]
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateSupportTerms",
      "Entity": {
        "Type": "Offer@1.0",
        "Identifier": "offer-11111111111111"
      }
    }
  ]
}
```

```

        },
        "DetailsDocument": {
            "Terms": [
                {
                    "Type": "SupportTerm",
                    "RefundPolicy": "Updated refund policy description"
                }
            ]
        }
    ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Products for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Describe an AMI, SaaS, or Container product using an AWS SDK](#)
- [List all AMI, SaaS, or Container products and associated public offers using an AWS SDK](#)

Describe an AMI, SaaS, or Container product using an AWS SDK

The following code examples show how to describe an AMI, SaaS, or Container product and check if it contains all the information you want to know about the product.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
    }
}

```

- For API details, see [DescribeEntity](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
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()

```

- For API details, see [DescribeEntity](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List all AMI, SaaS, or Container products and associated public offers using an AWS SDK

The following code examples show how to list all AMI, SaaS, or Container products and associated public offers.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see the following topics in *AWS SDK for Java 2.x API Reference*.
 - [DescribeEntity](#)
 - [ListEntities](#)

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
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()
```

- For API details, see the following topics in *AWS SDK for Python (Boto3) API Reference*.
 - [DescribeEntity](#)
 - [ListEntities](#)

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Resale authorization for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Create draft resale authorization for any product type using an AWS SDK](#)
- [Describe a resale authorization using an AWS SDK](#)
- [Publish a one-time resale authorization with a private offer using an AWS SDK](#)
- [Publish multi-use resale authorization with an expiration date using an AWS SDK](#)
- [Publish multi-use resale authorization with an expiration date and a EULA to be sent to the buyer using an AWS SDK](#)
- [Publish multi-use resale authorization with an expiration date and add reseller contract documentation using an AWS SDK](#)
- [Publish multi-use resale authorization with expiration and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish multi-use resale authorization without an expiration date using an AWS SDK](#)
- [Publish multi-use resale authorization without an expiration date and add a custom EULA using an AWS SDK](#)
- [Publish multi-use resale authorization without an expiration date and add reseller contract documentation using an AWS SDK](#)
- [Publish multi-use resale authorization without expiration date and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish one-time resale authorization and add Flexible payment schedule using an AWS SDK](#)

- [Publish one-time resale authorization for any product type and add a EULA using an AWS SDK](#)
- [Publish one-time resale authorization and add a specific buyer account for the resale using an AWS SDK](#)
- [Publish one-time resale authorization for any product type and add reseller contract documentation using an AWS SDK](#)
- [Publish one-time resale authorization for and add whether it is a renewal using an AWS SDK](#)
- [Restrict resale authorization using an AWS SDK](#)
- [Update name and description of one-time or multi-use resale authorization using an AWS SDK](#)

Create draft resale authorization for any product type using an AWS SDK

The following code examples show how to create draft resale authorization for any product type so you can review them internally before publishing to a Channel Partner.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

```
{
  "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"
    }
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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"
      }
    }
  ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Describe a resale authorization using an AWS SDK

The following code examples show how to describe a resale authorization.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}
```

- For API details, see [DescribeEntity](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()
```

- For API details, see [DescribeEntity](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish a one-time resale authorization with a private offer using an AWS SDK

The following code examples show how to publish a one-time resale authorization with a private offer so a Channel Partner can use it to create a Channel Partner Private Offer (CPPO).

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{  
  "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
        }
    }
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "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
        }
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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 (CPPO)
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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization with an expiration date using an AWS SDK

The following code examples show how to publish multi-use resale authorization with an expiration date for an AMI product with hourly annual pricing so a Channel Partner can use it to create a CPPO.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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 (CPPO)
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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization with an expiration date and a EULA to be sent to the buyer using an AWS SDK

The following code examples show how to publish multi-use resale authorization with an expiration date for any product type and add a custom EULA to be sent to the buyer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "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"
                }
            ]
        }
    ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization with an expiration date and add reseller contract documentation using an AWS SDK

The following code examples show how to publish multi-use resale authorization with an expiration date for any product type and add reseller contract documentation between the ISV and Channel Partner.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

```
]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization with expiration and add a specific buyer account for the resale using an AWS SDK

The following code examples show how to publish multi-use resale authorization with an expiration date for any product type and add a specific buyer account for the resale.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization without an expiration date using an AWS SDK

The following code examples show how to publish multi-use resale authorization without an expiration date for an AMI product with hourly annual pricing so a CP can use that to create a CPPO.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

```

    }
  ]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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"
                    }
                ]
            }
        ]
    }
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization without an expiration date and add a custom EULA using an AWS SDK

The following code examples show how to publish multi-use resale authorization without an expiration date for any product type and add a custom EULA to be sent to the buyer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization without an expiration date and add reseller contract documentation using an AWS SDK

The following code examples show how to publish multi-use resale authorization without an expiration date for any product type and add reseller contract documentation between the ISV and Channel Partner.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "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"
        }
      ]
    }
  ]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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"
                        }
                    ]
                }
            ]
        }
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish multi-use resale authorization without expiration date and add a specific buyer account for the resale using an AWS SDK

The following code examples show how to publish multi-use resale authorization without an expiration date for any product type and add a specific buyer account for the resale.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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",

```


[illegible]

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
"""
Purpose
Publish multi-use resale authorization without expiry date for any product type
(AMI/SaaS/Container) and add specific buyer account for the resale
CAPI-83
"""

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 without expiry date and add
specific buyer account",
    )
```

```
if __name__ == "__main__":  
    main()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish one-time resale authorization and add Flexible payment schedule using an AWS SDK

The following code examples show how to publish one-time resale authorization for any product type and add Flexible payment schedule.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

```

    ]
  }
}
]
}
]
}
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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"
            }
          ]
        }
      ]
    }
  }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish one-time resale authorization for any product type and add a EULA using an AWS SDK

The following code examples show how to publish one-time resale authorization for any product type and add a custom EULA to be sent to the buyer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {

```


Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish one-time resale authorization and add a specific buyer account for the resale using an AWS SDK

The following code examples show how to publish one-time resale authorization for any product type and add a specific buyer account for the resale.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish one-time resale authorization for any product type and add reseller contract documentation using an AWS SDK

The following code examples show how to publish one-time resale authorization for any product type and add reseller contract documentation between the ISV and Channel Partner.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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-Contract-for-AWS-Marketplace-2022-07-14.pdf"
          }
        ]
      }
    ]
  }
}
}

```

```
]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish one-time resale authorization for and add whether it is a renewal using an AWS SDK

The following code examples show how to publish one-time resale authorization for any product type and add whether it is a renewal.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "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"
            }
        }
    }
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Restrict resale authorization using an AWS SDK

The following code examples show how to restrict resale authorization.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "RestrictResaleAuthorization",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-111111111111"
      },
      "DetailsDocument": {}
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
```

```

    "Catalog": "AWSMarketplace",
    "ChangeSet": [
        {
            "ChangeType": "RestrictResaleAuthorization",
            "Entity": {
                "Type": "ResaleAuthorization@1.0",
                "Identifier": "resaleauthz-111111111111"
            },
            "DetailsDocument": {}
        }
    ]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update name and description of one-time or multi-use resale authorization using an AWS SDK

The following code examples show how to update name and description of one-time or multi-use resale authorization before publishing for any product type.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-111111111111"
      },
      "DetailsDocument": {
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product"
      }
    }
  ]
}
```

```
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateInformation",
      "Entity": {
        "Type": "ResaleAuthorization@1.0",
        "Identifier": "resaleauthz-111111111111"
      },
      "DetailsDocument": {
        "Name": "TestResaleAuthorization",
        "Description": "Worldwide ResaleAuthorization for Test Product"
      }
    }
  ]
}
```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

SaaS products for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Create a draft SaaS product with a draft public offer using an AWS SDK](#)
- [Create a public or limited SaaS product and public offer with contract pricing using an AWS SDK](#)
- [Create a public or limited SaaS product and public offer with contract with Pay-As-You-Go pricing using an AWS SDK](#)

- [Create a public or limited SaaS product and public offer with subscription pricing using an AWS SDK](#)
- [Publish a SaaS product and associated public offer using an AWS SDK](#)
- [Publish a SaaS product and associated public offer from an existing draft using an AWS SDK](#)
- [Update dimensions on an AMI or SaaS product using an AWS SDK](#)

Create a draft SaaS product with a draft public offer using an AWS SDK

The following code examples show how to create a draft SaaS product with a draft public offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

```
{
  "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"
    }
}
]
}

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
    "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"
            }
        }
    ]
}

```

```

        }
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()

```


- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a public or limited SaaS product and public offer with contract pricing using an AWS SDK

The following code examples show how to create a public or limited SaaS product and public offer with contract pricing. This example creates either a standard or custom EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a public or limited SaaS product and public offer with contract with Pay-As-You-Go pricing using an AWS SDK

The following code examples show how to create a public or limited SaaS product and public offer with a contract with Pay-As-You-Go pricing. This example creates either a standard or custom EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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"
    }
  ]
}
```

```

        ],
        "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a public or limited SaaS product and public offer with subscription pricing using an AWS SDK

The following code examples show how to create a public or limited SaaS product and public offer with subscription pricing. This examples creates either a standard or custom EULA.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
    }
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish a SaaS product and associated public offer using an AWS SDK

The following code examples show how to publish a SaaS product and associated public offer. The product will be in a limited state by default.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities to start a changeset* from the **Utilities** section.

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

```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

{
  "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": {}
}
]
}

```

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```

# 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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Publish a SaaS product and associated public offer from an existing draft using an AWS SDK

The following code example shows how to publish a SaaS product and associated public offer from an existing draft. The product will be in a limited state by default.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateVisibility",
      "ChangeName": "CreateProductChange",
      "Entity": {
        "Type": "SaaSProduct@1.0",
        "Identifier": "prod-111111111111"
      },
      "DetailsDocument": {
        "TargetVisibility": "Public"
      }
    }
  ]
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update dimensions on an AMI or SaaS product using an AWS SDK

The following code examples show how to update dimensions on an AMI or SaaS product.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

To run this example, pass the following JSON changeset to `RunChangesets` in *Utilities* to *start a changeset* from the **Utilities** section.

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

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

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

Run this script to start the changeset. Helper functions are defined in *Utilities to start a changeset* from the **Utilities** section.

```
# 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

CAP-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()
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Utilities for AWS Marketplace Catalog API

The following code examples show how to use AWS Marketplace Catalog API with AWS SDKs.

Examples

- [Utilities to start a changeset using an AWS SDK](#)

Utilities to start a changeset using an AWS SDK

The following code examples show how to define utilities to start a changeset.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

Utility to load a changeset from a JSON file and start processing it.

```
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;
}
```

```
}  
}
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

Utility to start a changeset.

```
"""  
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)

```

Utility to load a changeset from a JSON file.

```

"""
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
```

- For API details, see [StartChangeSet](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Code examples for AWS Marketplace Agreement API using AWS SDKs

The following code examples show how to use AWS Marketplace Agreement API with an AWS software development kit (SDK).

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

AWS Marketplace Agreement API

- [Agreements for AWS Marketplace Agreement API](#)
 - [Accept an agreement cancellation request using an AWS SDK](#)
 - [Accept an agreement payment request using an AWS SDK](#)
 - [Amend a SaaS contract agreement renewal term using an AWS SDK](#)
 - [Amend an AMI ConfigurableUpfrontPricingTerm for usage pricing model using an AWS SDK](#)
 - [Create a new AMI free trial agreement using an AWS SDK](#)
 - [Create a new SaaS contract agreement with upfront payment using an AWS SDK](#)
 - [Get all agreement IDs using an AWS SDK](#)
 - [Get all agreements using an AWS SDK](#)
 - [Get customer ID from an agreement using an AWS SDK](#)
 - [Get details of an agreement cancellation request using an AWS SDK](#)
 - [Get details of an agreement payment request using an AWS SDK](#)
 - [Get financial details from an agreement using an AWS SDK](#)
 - [Get free trial details from an agreement using an AWS SDK](#)
 - [Get information about an agreement using an AWS SDK](#)
 - [Get product and offer details from an agreement using an AWS SDK](#)
 - [Get the EULA of an agreement using an AWS SDK](#)
 - [Get the auto renewal terms of an agreement using an AWS SDK](#)
 - [Get the dimensions purchased in an agreement using an AWS SDK](#)
 - [Get the instances of each dimension purchased in an agreement using an AWS SDK](#)
 - [Get the payment schedule of an agreement using an AWS SDK](#)
 - [Get the pricing per dimension in an agreement using an AWS SDK](#)
 - [Get the pricing type of an agreement using an AWS SDK](#)
 - [Get the product type of an agreement using an AWS SDK](#)
 - [Get the status of an agreement using an AWS SDK](#)
 - [Get the support terms of an agreement using an AWS SDK](#)

- [Get the terms of an agreement using an AWS SDK](#)
- [List agreement cancellation requests using an AWS SDK](#)
- [List agreement payment requests using an AWS SDK](#)
- [Reject an agreement cancellation request using an AWS SDK](#)
- [Reject an agreement payment request using an AWS SDK](#)
- [Replace a SaaS contract agreement with an agreement-based offer using an AWS SDK](#)
- [Replace a SaaS free trial with Contract with Consumption Pricing using an AWS SDK](#)
- [Replace a SaaS usage-based pricing term and cancel the previous agreement using an AWS SDK](#)
- [Replace an AMI usage-based pricing term but keep the existing ConfigurableUpfrontPricingTerm using an AWS SDK](#)
- [Search for agreements by account ID using an AWS SDK](#)
- [Search for agreements by agreement ID using an AWS SDK](#)
- [Search for agreements by end date using an AWS SDK](#)
- [Search for agreements by offer ID using an AWS SDK](#)
- [Search for agreements by product ID using an AWS SDK](#)
- [Search for agreements by status using an AWS SDK](#)
- [Search for agreements with one custom filter using an AWS SDK](#)
- [Search for agreements with two custom filters using an AWS SDK](#)
- [Update purchase orders after agreement acceptance using an AWS SDK](#)

Agreements for AWS Marketplace Agreement API

The following code examples show how to use AWS Marketplace Agreement API with AWS SDKs.

Examples

- [Accept an agreement cancellation request using an AWS SDK](#)
- [Accept an agreement payment request using an AWS SDK](#)
- [Amend a SaaS contract agreement renewal term using an AWS SDK](#)
- [Amend an AMI ConfigurableUpfrontPricingTerm for usage pricing model using an AWS SDK](#)
- [Create a new AMI free trial agreement using an AWS SDK](#)
- [Create a new SaaS contract agreement with upfront payment using an AWS SDK](#)

- [Get all agreement IDs using an AWS SDK](#)
- [Get all agreements using an AWS SDK](#)
- [Get customer ID from an agreement using an AWS SDK](#)
- [Get details of an agreement cancellation request using an AWS SDK](#)
- [Get details of an agreement payment request using an AWS SDK](#)
- [Get financial details from an agreement using an AWS SDK](#)
- [Get free trial details from an agreement using an AWS SDK](#)
- [Get information about an agreement using an AWS SDK](#)
- [Get product and offer details from an agreement using an AWS SDK](#)
- [Get the EULA of an agreement using an AWS SDK](#)
- [Get the auto renewal terms of an agreement using an AWS SDK](#)
- [Get the dimensions purchased in an agreement using an AWS SDK](#)
- [Get the instances of each dimension purchased in an agreement using an AWS SDK](#)
- [Get the payment schedule of an agreement using an AWS SDK](#)
- [Get the pricing per dimension in an agreement using an AWS SDK](#)
- [Get the pricing type of an agreement using an AWS SDK](#)
- [Get the product type of an agreement using an AWS SDK](#)
- [Get the status of an agreement using an AWS SDK](#)
- [Get the support terms of an agreement using an AWS SDK](#)
- [Get the terms of an agreement using an AWS SDK](#)
- [List agreement cancellation requests using an AWS SDK](#)
- [List agreement payment requests using an AWS SDK](#)
- [Reject an agreement cancellation request using an AWS SDK](#)
- [Reject an agreement payment request using an AWS SDK](#)
- [Replace a SaaS contract agreement with an agreement-based offer using an AWS SDK](#)
- [Replace a SaaS free trial with Contract with Consumption Pricing using an AWS SDK](#)
- [Replace a SaaS usage-based pricing term and cancel the previous agreement using an AWS SDK](#)
- [Replace an AMI usage-based pricing term but keep the existing ConfigurableUpfrontPricingTerm using an AWS SDK](#)

- [Search for agreements by account ID using an AWS SDK](#)
- [Search for agreements by agreement ID using an AWS SDK](#)
- [Search for agreements by end date using an AWS SDK](#)
- [Search for agreements by offer ID using an AWS SDK](#)
- [Search for agreements by product ID using an AWS SDK](#)
- [Search for agreements by status using an AWS SDK](#)
- [Search for agreements with one custom filter using an AWS SDK](#)
- [Search for agreements with two custom filters using an AWS SDK](#)
- [Update purchase orders after agreement acceptance using an AWS SDK](#)

Accept an agreement cancellation request using an AWS SDK

The following code examples show how to accept an agreement cancellation request initiated by the seller.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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.AcceptAgreementCancellationRe
import
    software.amazon.awssdk.services.marketplaceagreement.model.AcceptAgreementCancellationRe

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();

    AcceptAgreementCancellationRequestRequest request =
        AcceptAgreementCancellationRequestRequest.builder()
            .agreementId(AGREEMENT_ID)

.agreementCancellationRequestId(AGREEMENT_CANCELLATION_REQUEST_ID)
            .build();

    AcceptAgreementCancellationRequestResponse 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());
}
}

```

- For API details, see [AcceptAgreementCancellationRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
```

- For API details, see [AcceptAgreementCancellationRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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_REQUEST_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()
```

- For API details, see [AcceptAgreementCancellationRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Accept an agreement payment request using an AWS SDK

The following code examples show how to accept an agreement payment request initiated by the seller.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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.AcceptAgreementPaymentRequest

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());
    }
}

```

- For API details, see [AcceptAgreementPaymentRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
```

- For API details, see [AcceptAgreementPaymentRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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()
```

- For API details, see [AcceptAgreementPaymentRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Amend a SaaS contract agreement renewal term using an AWS SDK

The following code examples show how to amend a SaaS contract agreement to add or update its renewal term.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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());
    }
}

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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();

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
```

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()
```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Amend an AMI ConfigurableUpfrontPricingTerm for usage pricing model using an AWS SDK

The following code examples show how to amend an AMI agreement with ConfigurableUpfrontPricingTerm by updating dimension quantity for usage pricing model.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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());
    }
}

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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();

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

"""

```

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
                        },
                    ],
                }
            },
        }

    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()

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a new AMI free trial agreement using an AWS SDK

The following code examples show how to create a new AMI free trial agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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());
    }
}

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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();

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

"""
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()
```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Create a new SaaS contract agreement with upfront payment using an AWS SDK

The following code examples show how to create a new SaaS contract agreement with upfront payment.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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.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()

                    .build(),
                    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());
    }
}

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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();

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

"""
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()
```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get all agreement IDs using an AWS SDK

The following code examples show how to get all agreement IDs.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [SearchAgreements](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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)
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get all agreements using an AWS SDK

The following code examples show how to get all agreements.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [SearchAgreements](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

# 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)
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get customer ID from an agreement using an AWS SDK

The following code examples show how to get customer ID from an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
    }
}
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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']}")
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get details of an agreement cancellation request using an AWS SDK

The following code examples show how to get details of a specific agreement cancellation request.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();

        GetAgreementCancellationRequestRequest request =
            GetAgreementCancellationRequestRequest.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());
    }
}
```

- For API details, see [GetAgreementCancellationRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
```

- For API details, see [GetAgreementCancellationRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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()

```

- For API details, see [GetAgreementCancellationRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get details of an agreement payment request using an AWS SDK

The following code examples show how to get details of a specific agreement payment request.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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.GetAgreementPaymentRequestReq
import
    software.amazon.awssdk.services.marketplaceagreement.model.GetAgreementPaymentRequestRes

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());
    }
}
```

- For API details, see [GetAgreementPaymentRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
```

- For API details, see [GetAgreementPaymentRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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()

```

- For API details, see [GetAgreementPaymentRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get financial details from an agreement using an AWS SDK

The following code examples show how to get financial details from an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

# 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")

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get free trial details from an agreement using an AWS SDK

The following code examples show how to get free trial details from an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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}")

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get information about an agreement using an AWS SDK

The following code examples show how to get information about an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

# 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)
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get product and offer details from an agreement using an AWS SDK

The following code examples show how to get product and offer details from an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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)

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the EULA of an agreement using an AWS SDK

The following code examples show how to get the EULA of an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()
```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the auto renewal terms of an agreement using an AWS SDK

The following code examples show how to get the auto renewal terms of an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

# 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()

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the dimensions purchased in an agreement using an AWS SDK

The following code examples show how to get the dimensions purchased in an agreement.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
    }
}

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()
```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the instances of each dimension purchased in an agreement using an AWS SDK

The following code examples show how to get the instances of each dimension purchased in an agreement.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

# 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()
```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the payment schedule of an agreement using an AWS SDK

The following code examples show how to get the payment schedule of an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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;
}

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the pricing per dimension in an agreement using an AWS SDK

The following code examples show how to get the pricing per dimension in an agreement.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()
```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the pricing type of an agreement using an AWS SDK

The following code examples show how to get the pricing type of an agreement.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
                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;
}
}

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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-11111111111111"

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 temr 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()
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the product type of an agreement using an AWS SDK

The following code examples show how to get the product type of an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
    }
}
```

```
}  
}
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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\nagreement 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()

```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the status of an agreement using an AWS SDK

The following code examples show how to get the status of an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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}")
```

- For API details, see [DescribeAgreement](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the support terms of an agreement using an AWS SDK

The following code examples show how to get the support terms of an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
    }
}

```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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}")
```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Get the terms of an agreement using an AWS SDK

The following code example shows how to get the terms of an agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}
```

- For API details, see [GetAgreementTerms](#) in *AWS SDK for Java 2.x API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List agreement cancellation requests using an AWS SDK

The following code examples show how to list agreement cancellation requests for agreements I participate in as acceptor.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.


```

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);
}
}

```

- For API details, see [ListAgreementCancellationRequests](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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();
```

- For API details, see [ListAgreementCancellationRequests](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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()
```

- For API details, see [ListAgreementCancellationRequests](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

List agreement payment requests using an AWS SDK

The following code examples show how to list agreement payment requests for agreements I participate in as acceptor.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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());
            }
        } while (response.nextToken() != null);
    }
}

```

```

        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);
}
}

```

- For API details, see [ListAgreementPaymentRequests](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

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,
            })
        );
    } while (response.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();
```

- For API details, see [ListAgreementPaymentRequests](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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()

```

- For API details, see [ListAgreementPaymentRequests](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Reject an agreement cancellation request using an AWS SDK

The following code examples show how to reject an agreement cancellation request initiated by the seller.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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.RejectAgreementCancellationRequest;
import
    software.amazon.awssdk.services.marketplaceagreement.model.RejectAgreementCancellationResponse;

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();

        RejectAgreementCancellationRequest request =
            RejectAgreementCancellationRequest.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());
    }
}

```

- For API details, see [RejectAgreementCancellationRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
```

- For API details, see [RejectAgreementCancellationRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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_REQUEST_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()
```

- For API details, see [RejectAgreementCancellationRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Reject an agreement payment request using an AWS SDK

The following code examples show how to reject an agreement payment request initiated by the seller.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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.RejectAgreementPaymentRequest;
```

```

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());
    }
}

```

- For API details, see [RejectAgreementPaymentRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
```


- For API details, see [RejectAgreementPaymentRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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()
```

- For API details, see [RejectAgreementPaymentRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Replace a SaaS contract agreement with an agreement-based offer using an AWS SDK

The following code examples show how to replace a SaaS contract agreement with a new agreement-based offer.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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 ABO. New AgreementId: " +
acceptAgreementRequestResponse.agreementId());
    }
}

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

"""
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 ABO. New AgreementId: " +
accept_response["agreementId"])

if __name__ == "__main__":
    ReplaceSaaSContractWithAgreementBasedOffer.replace_existing_agreement()

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Replace a SaaS free trial with Contract with Consumption Pricing using an AWS SDK

The following code examples show how to replace a SaaS free trial with a Contract with Consumption Pricing (CCP).

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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());
    }
}

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
```

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()
```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Replace a SaaS usage-based pricing term and cancel the previous agreement using an AWS SDK

The following code examples show how to replace a SaaS usage-based pricing term and cancel the previous agreement.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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) {
        replaceSaaSUBptAndCancel();
    }

    /**
     * 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 replaceSaaSUBptAndCancel() {
        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);
    }

```

```
}
```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```

"""

```

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()
```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Replace an AMI usage-based pricing term but keep the existing ConfigurableUpfrontPricingTerm using an AWS SDK

The following code examples show how to replace an AMI usage-based pricing term while keeping the existing ConfigurableUpfrontPricingTerm.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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())
            .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());
    }
}

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();
```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
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()

```

- For API details, see [CreateAgreementRequest](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements by account ID using an AWS SDK

The following code example shows how to search for agreements by account ID.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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)
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements by agreement ID using an AWS SDK

The following code example shows how to search for agreements by agreement ID.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements by end date using an AWS SDK

The following code examples show how to search for agreements by end date.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Java 2.x API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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)
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements by offer ID using an AWS SDK

The following code example shows how to search for agreements by offer ID.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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()

```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements by product ID using an AWS SDK

The following code example shows how to search for agreements by product ID.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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-111111111111"

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()
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements by status using an AWS SDK

The following code example shows how to search for agreements by status.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
# 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)
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements with one custom filter using an AWS SDK

The following code example shows how to search for agreements with one custom filter.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}
```

- For API details, see [SearchAgreements](#) in *AWS SDK for Java 2.x API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Search for agreements with two custom filters using an AWS SDK

The following code example shows how to search for agreements with two custom filters.

Java

SDK for Java 2.x

 Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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;
}
}

```

- For API details, see [SearchAgreements](#) in *AWS SDK for Java 2.x API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Update purchase orders after agreement acceptance using an AWS SDK

The following code examples show how to update purchase orders after the agreement has been accepted.

Java

SDK for Java 2.x

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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));
}

```



```
}

```

- For API details, see [UpdatePurchaseOrders](#) in *AWS SDK for Java 2.x API Reference*.

JavaScript

SDK for JavaScript (v3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
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();

```

- For API details, see [UpdatePurchaseOrders](#) in *AWS SDK for JavaScript API Reference*.

Python

SDK for Python (Boto3)

Note

There's more on GitHub. Find the complete example and learn how to set up and run in the [AWS Marketplace API Reference Code Library](#) repository.

```
"""
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_

```

- For API details, see [UpdatePurchaseOrders](#) in *AWS SDK for Python (Boto3) API Reference*.

For a complete list of AWS SDK developer guides and code examples, see [Using this service with an AWS SDK](#). This topic also includes information about getting started and details about previous SDK versions.

Document history

The following table describes the documentation updates for this guide.

Change	Description	Date
AWS Marketplace Discovery API public launch	The AWS Marketplace Discovery API is now available . Added 9 API operations: GetListing, GetProduct, GetOffer, GetOfferTerms, GetOfferSet, ListPurchaseOptions, ListFulfillmentOptions, SearchFacets, and SearchListings. Standard AWS SDK support and IAM-based access.	April 8, 2026
Updates to ML products	Updated seller product machine learning information.	March 26, 2025
CustomerIdentifier	Added notice of deprecation of CustomerIdentifier in March 2026.	March 25, 2025
AWS Marketplace Catalog API topic updates	Updated CPPO prerequisites.	May 9, 2024
AWS Marketplace Catalog API topic updates	Updated instances of the Details attribute to DetailsDocument .	April 30, 2024
AWS Marketplace Catalog API topic updates	Added notes to Working with offers to clarify constraints.	April 17, 2024
AWS Marketplace Catalog API supports service-linked for resale authorization	Updated resale authorization prerequisites for service-linked role.	March 20, 2024

AWS Marketplace Catalog API supports organization units	Added content to enable private marketplace support at Organization Unit (OU) level.	February 16, 2024
AWS Marketplace Catalog API supports setting intent on requests	Sellers now can request changes for entities with specific intent using the AWS Marketplace Catalog API.	February 9, 2024
AWS Marketplace Catalog API supports wildcard filter validation	Added wildcard filter validation in ListEntities API.	February 5, 2024
AWS Marketplace Catalog API supports Amazon EKS add-ons	Added content and error messages related to publishing to Amazon EKS add-ons from AWS Marketplace container-based product.	January 29, 2024
AWS Marketplace Catalog API supports listing details about entities	Sellers can now list details about entities using the AWS Marketplace Catalog API.	December 19, 2023
The AWS Marketplace Deployment Service API reference is now generally available	This service provides an API interface that supports a secure method for passing deployment parameters (for example, API keys and external IDs) during the Quick Launch experience.	November 29, 2023

The AWS Marketplace Agreement Service API reference is now generally available	This service provides an API interface that helps AWS Marketplace sellers manage their product-related agreements, including listing, searching, and filtering agreements.	November 29, 2023
AWS Marketplace Catalog API supports the ability to create products, offers, Resale Authorizations, and CPPOs	Sellers can now use the AWS Marketplace Catalog API to create and update products , offers , Resale Authorizations , and channel partner private offers (CPPOs) .	November 29, 2023
AWS Marketplace Catalog API supports enhanced filtering and sorting capabilities	Sellers can now sort and filter products using the AWS Marketplace Catalog API.	November 29, 2023
AWS Marketplace Catalog API supports resource sharing	The AWS Marketplace Catalog API integrates with AWS Resource Access Manager (AWS RAM) to enable resource sharing. See Working with AWS RAM to share resources .	April 12, 2023
AWS Marketplace Discovery API topic update	The AWS Marketplace Discovery API now supports CloudTrail. See Logging AWS Marketplace Discovery API calls using AWS CloudTrail .	December 15, 2022
AWS Marketplace supports archiving private marketplace experiences	Buyers can now archive and reactivate private marketplace experiences in AWS Marketplace. See Working with a private marketplace .	December 12, 2022

AWS Marketplace Private marketplace granular permissions	Buyers now have more granular permissions to manage private marketplace experiences. See Working with a private marketplace .	September 8, 2022
AWS Marketplace Discovery API release notes	Added Release notes for the AWS Marketplace Discovery API.	May 20, 2022
AWS Marketplace Discovery API topic update	Documentation-only update to the AWS Marketplace Discovery API topic .	January 14, 2022
Support for Helm chart delivery options and QuickLaunch for container-based products	Added documentation for adding or updating Helm chart delivery options in container-based product versions, including enabling QuickLaunch for buyers. See Working with container-based products .	November 29, 2021
Support for managing seller products	Added the ability to manage AMI and container products programmatically. See Working with seller products .	March 26, 2021
Support for managing private marketplaces	Added the ability to create and maintain private marketplaces for AWS Organizations programmatically. See Working with a private marketplace .	December 3, 2020

[The AWS Marketplace
Discovery API is now available](#)

The Discovery API provides programmatic access to find products in the AWS Marketplace. For details, see [Discovery API](#).

September 30, 2020

[The AWS Marketplace Catalog
API is now generally available](#)

This service provides an API interface for approved providers to programmatically access the self-service publishing capabilities on the AWS Marketplace Management Portal.

November 12, 2019

Release notes for AWS Marketplace Discovery API

Release notes for AWS Marketplace Discovery API. Details about new features, improvements, fixes, and announcements.

Discovery API release notes for 2026

April 8, 2026

The AWS Marketplace Discovery API is now publicly available through the standard AWS SDK. The API provides programmatic access to the AWS Marketplace catalog, including searching and browsing listings, retrieving product details and fulfillment options, and accessing public and private offer pricing and terms.

Launch announcements

- 9 API operations available: GetListing, GetProduct, GetOffer, GetOfferTerms, GetOfferSet, ListPurchaseOptions, ListFulfillmentOptions, SearchFacets, and SearchListings.
- Standard AWS SDK support in all languages (Java, Python, JavaScript, .NET, Go, and others). No private SDK required.
- IAM-based access control with resource-level permissions. No manual onboarding required.
- Available in US East (N. Virginia), US West (Oregon), and Europe (Ireland).

Discovery API release notes for 2024

August 15, 2024

Discovery API customers can access the updated Discovery API documentation and SDK on the Amazon Simple Storage Service bucket provided by the Discovery API team. Customers can refer to the Change Log in the private documentation for more details.

Launch announcements

The Discovery API Private SDK is now available in JavaScript V3.

- For information about using the AWS SDK for JavaScript V3, see the [AWS SDK for JavaScript V3 Developer Guide](#).
- For information about migrating from V2 to V3, see [Migrate from version 2.x to 3.x of the AWS SDK for JavaScript](#), in the *AWS SDK for JavaScript V3 Developer Guide*.

Improvements

- AWS SDK paginators are now available for the `SearchListings` and `ListListingViewQueries` API operations in all supported languages that have native SDK support for paginators.

You can still use `NextToken` to manually paginate. You can use `NextToken` with the `SearchListings`, `ListListingViewQueries`, and `GetSearchFacets` API operations across all the SDKs that we vend.

- SDK updated for all supported languages with the latest AWS SDK artifacts.
- Private documentation updated to include an SDK usage section for JavaScript V3.

Discovery API release notes for 2022

May 20, 2022

Published on May 20, 2022

Discovery API customers can access the updated Discovery API documentation and SDK on the Amazon Simple Storage Service (Amazon S3) bucket that the Discovery API team shared with them previously. Refer to the Change Log in the private documentation for more details.

Discovery API announces the following launch, and improvements:

Launch announcements

- Discovery API launched in two additional AWS Regions:
 - US West (Oregon) – `us-west-2`
 - Europe (Ireland) – `eu-west-1`
- Discovery API Private SDK is now available in Java 2.x:
 - For more information about how to use the AWS SDK for Java 2.x, see the [AWS SDK for Java 2.x Developer Guide](#).
 - For more information about migration, see [migrating from version 1.x to 2.x of the AWS SDK for Java](#) in the *AWS SDK for Java 2.x Developer Guide*.
 - For more information about changes between versions 1.11.x and 2.x of the AWS SDK for Java 2.x, see [1.11 to 2.x Changelog](#) on the GitHub website.

Improvements

- Enhanced sorting functionality for the `SearchListings` API operation by introducing new options for:
 - `SortBy` – `AVERAGE_CUSTOMER_RATING`, `CREATION_TIME`, `LAST_MODIFIED_TIME`
 - `SortOrder` – `ASCENDING`
- SDK updated for all the existing languages with the latest AWS SDK artifacts.
- Documentation updated to include SDK usage section for all languages.